



ББК 65.290-23  
УДК 658.014.1  
П27

**Перерва А., Еранов С., Иванова В., Сергеев С.**

П27 Путь IT-менеджера. Управление проектной средой и IT-проектами. — СПб.: Питер, 2016. — 320 с.: ил.

ISBN 978-5-496-01948-4

Существует много замечательных методологий и стандартов проектного управления. В своей книге авторы приводят только реально работающие и проверенные на собственном опыте практики, которые вы сразу же можете использовать в своей компании.

Что такое проектная среда и паттерны формирования проектной среды и управления?

Как выбрать оптимальный набор практик управления и разработки требований к информационным системам?

Как сделать проект управляемым и предсказуемым?

Как выгодно продать себя в качестве мастера управления проектной средой?

Ответы на эти и другие вопросы помогут вам максимально эффективно построить процессы управления и производства.

В издании представлена информация со ссылками на теоретические материалы из разных областей управления разработкой программного обеспечения: анализа, архитектуры, управления проектами, лидерства и управления персоналом — то, что может понадобиться в работе над реальными производственными проектами.

Проанализированы разнообразные кейсы и ситуации, приведены примеры документов и шаблонов, необходимых для разработки ПО. Помимо этого структурирован огромный массив теоретической информации в зависимости от того, что прошло проверку на практике и может быть использовано в работе профессионалов в области разработки программного обеспечения.

Книга адресована профессионалам-управленцам в проектной или функциональной областях разработки ПО, руководителям рабочих групп, менеджерам проектов, начальникам отделов, директорам департаментов по разработке ПО — найдите новые практики, готовые к применению в реальной производственной жизни. Издание также будет полезно начинающим менеджерам и менеджерам со стажем — обобщайте и систематизируйте знания, используйте уникальный опыт авторов в собственной практике.

**12+** (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 65.290-23  
УДК 658.014.1

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

# Содержание

<b>Предисловие автора</b> .....	7
<b>Об авторах</b> .....	9
<b>Благодарности</b> .....	12
<b>Введение</b> .....	14
Предыстория .....	15
Для кого эта книга .....	15
Чего вы НЕ найдете в этой книге .....	16
Как читать эту книгу .....	17
<b>1. Общие понятия</b> .....	21
1.1. Что такое бизнес в этой книге .....	21
1.2. Что такое информатизация бизнеса и зачем это надо знать .....	22
1.3. Что такое проект, программа проектов и портфель проектов .....	23
1.4. Что такое жизненный цикл продукта и проекта .....	26
1.5. Что такое архитектура, какой она бывает и зачем мне это надо знать .....	32
<b>2. Профиль и квалификация менеджеров</b> .....	35
2.1. Надо ли менеджеру быть аналитиком? .....	35
2.2. Профиль менеджера .....	37
2.3. Таблица квалификационных навыков .....	42
<b>3. Что такое паттерн</b> .....	49
<b>4. Паттерны формирования проектной среды</b> .....	62
4.1. Немного о среде проекта .....	62
4.2. Сводная таблица паттернов формирования проектной среды .....	69
4.2.1. Общие паттерны .....	69
4.2.2. Сводная таблица паттернов разработки ПО .....	75
4.3. Паттерн «Базовые правила работы» .....	75
4.4. Паттерн «Выученные уроки» .....	81
4.5. Паттерн «Хорошие практики» .....	84

4.6. Паттерн «Своевременное и полное информирование» . . . . .	86
4.6.1. Паттерн «Статус проекта» . . . . .	86
4.6.2. Паттерн «Информационный бюллетень» . . . . .	87
4.7. Паттерн «Управление средой проекта» . . . . .	89
4.8. Паттерн «Открытые вопросы» . . . . .	96
4.9. Паттерн «Система авторизации работ» . . . . .	97
4.10. Паттерн «Эффективный переговорщик» . . . . .	98
<b>5. Паттерны разработки ПО . . . . .</b>	<b>102</b>
5.1. Паттерн «Итеративная разработка ПО» . . . . .	102
5.1.1. Нулевая итерация (предварительный анализ) . . . . .	113
5.1.2. Первая итерация . . . . .	116
5.1.3. Вторая итерация . . . . .	118
5.1.4. Третья итерация . . . . .	120
5.1.5. Четвертая итерация . . . . .	122
5.1.6. Сводная таблица итераций . . . . .	123
5.1.7. Дальнейшая разработка . . . . .	126
5.1.8. Резюме паттерна . . . . .	126
5.2. Паттерн «Релизное управление» . . . . .	127
5.3. Паттерн «Обеспечение качества и анализ рисков» . . . . .	132
5.4. Паттерн «Пакеты спецификаций требований» . . . . .	140
5.4.1. Состав пакетов спецификаций требований . . . . .	141
5.4.2. Пакет Extreme . . . . .	141
5.4.3. Пакет Simple . . . . .	141
5.4.4. Пакет Basic . . . . .	142
5.4.5. Пакет Advanced . . . . .	143
5.4.6. Пакет Gold . . . . .	145
5.4.7. Пакет VIP . . . . .	147
5.4.8. Расширения пакетов спецификаций требований . . . . .	149
5.5. Паттерн CMMI . . . . .	151
5.6. Паттерн IDEAL . . . . .	154
5.7. Паттерн «Система управления производством ПО (СУПр)» . . . . .	158
5.8. Резюме главы . . . . .	162
<b>6. Паттерны управления . . . . .</b>	<b>163</b>
6.1. Сводная таблица паттернов управления . . . . .	163
6.1.1. Паттерны управления проектом . . . . .	163
6.1.2. Паттерны управления функциональным отделом . . . . .	168
6.1.3. Паттерны управления подразделением разработки ПО . . . . .	170
6.2. Еще раз о проекте . . . . .	172
6.3. Паттерны управления проектом . . . . .	173
6.3.1. Паттерн «Экстремальная разработка» . . . . .	173
6.3.2. Паттерн «Чек-листинг» . . . . .	180
6.3.3. Паттерн «Смартинг» . . . . .	181

6.3.4. Паттерн «Расширенный устав проекта» . . . . .	182
6.3.5. Паттерн «СУП на РМВОКе» . . . . .	184
6.3.6. Паттерн «СУП на воде, или Критические принципы проектного управления» . . . . .	190
6.3.7. Паттерн «Зажарка в СУП, или Универсальные принципы управления проектами» . . . . .	197
6.3.8. Паттерн «Стабилизация ПО». . . . .	201
6.3.9. Паттерн «С ног на голову, или ПМП ПСР» . . . . .	207
6.3.10. Паттерн «План управления документами» . . . . .	211
6.3.11. Плохие паттерны . . . . .	217
6.3.12. Так что же такое управление проектом? . . . . .	218
<b>6.4. Паттерны управления функциональным отделом . . . . .</b>	<b>219</b>
6.4.1. Паттерн «Фундамент отдела» . . . . .	220
6.4.2. Паттерн «Удаленное управление» . . . . .	223
6.4.3. Паттерн «Маслоу по Фрейду» . . . . .	224
6.4.4. Паттерн «Лидерство» . . . . .	227
6.4.5. Паттерн «Саморазвитие и личностный бренд» . . . . .	233
6.4.6. Паттерн «Самоорганизующиеся команды» . . . . .	235
6.4.7. Плохие паттерны . . . . .	241
6.4.8. Так что же такое управление отделом? . . . . .	242
<b>6.5. Паттерны управления подразделением разработки ПО . . . . .</b>	<b>243</b>
6.5.1. Наиболее важные паттерны . . . . .	244
6.5.2. Паттерн «Лидерство на уровне департамента» . . . . .	250
6.5.3. Паттерн «Стратегия и планирование в СУПр» . . . . .	252
6.5.4. Паттерн «Управление портфелем проекта» . . . . .	258
6.5.5. Плохие паттерны . . . . .	264
6.5.6. Так что же такое управление подразделением разработки ПО . . . . .	264
<b>7. Что умеет мастер управления проектной средой . . . . .</b>	<b>265</b>
7.1. Паттерн «Выживание в проектной среде» . . . . .	266
7.2. Паттерн «Особенности выполнения государственных контрактов и взаимодействия с государственными заказчиками» . . . . .	267
7.2.1. Заключаем государственный контракт . . . . .	267
7.2.2. Инициация проекта . . . . .	269
7.2.3. Психология государственного служащего . . . . .	270
7.2.4. Формализация процесса выполнения контракта . . . . .	271
7.2.5. Официальные письма . . . . .	272
7.2.6. Идите со всем готовым . . . . .	272
<b>8. Кадры решают все. . . . .</b>	<b>274</b>
8.1. Паттерн «IT-рекрутинг» . . . . .	274
8.1.1. Что важно? . . . . .	275
8.1.2. Охота за головами . . . . .	276
8.2. Паттерн «IT-интернатура». . . . .	277

8.3. Паттерн «Профиль должности» . . . . .	280
8.3.1. Шаг 1. Определяем должностной уровень . . . . .	280
8.3.2. Шаг 2. Определяем перечень профессиональных компетенций . . . . .	282
8.3.3. Шаг 3. Определяем, что важно для вашей команды . . . . .	283
8.3.4. Шаг 4. Учитываем особенности компании . . . . .	284
8.3.5. Шаг 5. Определяем приоритеты . . . . .	285
8.4. Паттерн «Оценка кандидата» . . . . .	285
8.4.1. Оценка и отбор кандидатов по резюме . . . . .	285
8.4.2. Проведение собеседования . . . . .	287
8.4.3. Где и как проводить собеседование . . . . .	291
8.4.4. Об отношении к людям . . . . .	292
8.4.5. Не тратьте время зря . . . . .	292
8.4.6. Кандидат дал свое согласие . . . . .	292
<b>9. Как продать себя как мастера управления проектной средой . . . . .</b>	<b>293</b>
<b>10. Карта знаний серии «Путь IT-профессионала» . . . . .</b>	<b>297</b>
<b>Заключение . . . . .</b>	<b>299</b>
<b>Глоссарий . . . . .</b>	<b>300</b>
<b>Словарь терминов . . . . .</b>	<b>310</b>
<b>Дополнительные материалы . . . . .</b>	<b>314</b>
<b>Список использованной литературы . . . . .</b>	<b>316</b>
<b>Использованные интернет-ресурсы . . . . .</b>	<b>318</b>
<b>Послесловие . . . . .</b>	<b>319</b>

# Предисловие автора

---

---

## Андрей Перерва

В своей книге «Путь аналитика. Практическое руководство IT-специалиста»<sup>1</sup> мы с Верой Ивановой рассмотрели некую «дорожную карту» (road map) профессионального развития и постарались показать, зачем и что нужно знать и уметь, как применять эти знания и умения на каждом этапе становления профессионала в индустрии разработки программного обеспечения.

Мы поделились с вами своими знаниями, ноу-хау и опытом — показали путь, которым можно следовать в построении собственной карьеры, чтобы стать гуру в разработке программного обеспечения.

Судя по полученным отзывам (см. <http://bp4you.ru/index.php/77-news/87-otzyvy2>) и письмам, мы достигли цели — книга позволила многим из вас получить нужные приемы и навыки и дала толчок как к развитию, так и к систематизации уже имевшихся знаний и умений. Мы очень рады этому! Это крайне важно и приятно для нас — осознавать, что поставленная цель была достигнута. Спасибо за отзывы и письма!

Именно поэтому мы решили продолжить начатое дело и создать серию книг «Путь IT-профессионала», в которой углубленно и детально рассмотрим некоторые наиболее важные, на наш взгляд, навыки и умения профессионалов в области разработки программного обеспечения (ПО). Мы слышим вопросы, которые задают коллеги и читатели, видим потребность в определенных знаниях, в которых мы имеем компетенции и опыт, полученные в ходе ежедневной профессиональной деятельности, и хотим по возможности закрыть часть этих белых пятен или упростить жизнь тем, кто уже знает способы, как эти пятна закрывать самостоятельно.

Мы не испытываем иллюзий в том, что наш путь в информационные технологии (IT) — единственно правильный или самый оптимальный. Отнюдь. Но мы прошли этот путь, и для нас он был эффективен. А эффективность в бизнесе — это одна

---

<sup>1</sup> *Иванова В., Перерва А.* Путь аналитика. Практическое руководство IT-специалиста. 2-е изд. — СПб.: Питер, 2015. — 304 с.

из главных составляющих успеха. Именно это и придает нам уверенность для создания серии книг «Путь IT-профессионала». Эти книги ориентированы на старших/ведущих аналитиков, руководителей рабочих групп, менеджеров проектов, начальников отделов анализа/разработки ПО.

Мы видим как минимум еще две книги в серии: «Путь IT-менеджера. Управление проектной средой и IT-проектами» — ее вы держите в руках, и книгу с рабочим названием «Мастер разработки ПО. Жизненный цикл разработки и паттерны анализа». Как вы уже понимаете из названий, в этих книгах мы будем останавливаться на более серьезных вопросах и, в отличие от первой, постараемся не столько обозначить некий план развития, сколько дать конкретные «удочки для ловли рыбы» — инструменты, которые, мы верим, помогут вам вырасти в мастера.

Как и в первой книге, здесь вы не найдете голой теории, да и вообще практически не найдете теории и рассуждений о том, какие подходы лучше, а какие хуже. Выводы, как и весь материал, с которым вам предстоит познакомиться, «выстраданы» авторами книги в «боевых» проектах и отражают наш личный профессиональный взгляд изнутри и снаружи мира разработки ПО во вселенной под названием IT.

Мы видим, что наш опыт востребован коллегами, и верим в то, что этот опыт может быть интересен и полезен и вам. Мы делаем успешные проекты и качественные продукты. Мы умеем рассказать о том, что знаем и как применяем эти знания на практике.

Итак...

Верите ли вы, что есть работающие управленческие паттерны (шаблоны и рецепты) для проектов разработки программного обеспечения?

Верите ли вы, что проект разработки программного обеспечения может быть предсказуемым, выполняться и быть законченным с требуемым качеством в определенные сроки и с согласованным бюджетом?

Верите ли в то, что проектная среда может быть сконфигурирована и управляема вами?

Знаете ли вы, как и что надо сделать, чтобы этого достичь?

Если вы мастер управления проектной средой, владеющий паттернами управления, то ваш ответ будет: «Конечно, да! Знаю и умею!»

# Об авторах

---

---

## **Андрей Дмитриевич Перерва**

41 год, Master of Business Administration/Master of Business Information.

Опыт работы в индустрии разработки программного обеспечения — более 20 лет. Опыт работы на руководящих позициях — 12 лет, из них в топ-менеджменте — восемь лет. В настоящий момент специализируется на вопросах стратегического и тактического планирования, операционном управлении и управлении проектами компании по разработке ПО.

В своем карьерном развитии прошел все возможные позиции в разработке ПО, работал программистом, проектировщиком и разработчиком баз данных (БД), бизнес-аналитиком, системным аналитиком, архитектором, менеджером проекта, начальником отдела бизнес- и системного анализа. Принимал участие в разработке таких информационных систем, как система управления потоками работ (Workflow), Boeing Reference Engineering Data Automated Retrieval System, Customer Relationship Management System, системы управления предприятием (Enterprise Management System), корпоративные порталы и системы комплексной автоматизации бизнеса, и руководил такими разработками.

Имеет опыт организационного проектирования, создания функциональных отделов анализа с нуля, построения и развития команды, проведения обучающих тренингов для сотрудников и менеджеров компании.

Опыт собственного бизнес-стартапа от создания и продажи бизнес-плана инвестору с последующим открытием собственной компании до создания собственного уникального решения и семейства продуктов.

Имеет практический опыт создания системы менеджмента качества (QMS) компании, построения процесса разработки ПО (SDLC) в соответствии с методологиями и стандартами RUP, Iconix, Agile, PMBOK, CMMI, ISO 9001.

## **Сергей Валерьевич Еранов**

35 лет, опыт работы в разработке программного обеспечения — около 15 лет. За профессиональную карьеру Сергей побывал в ролях программиста, руководителя

группы, архитектора, менеджера проекта, руководителя функционального подразделения.

Имеет опыт внедрения изменений в процесс разработки программного обеспечения на основе принципов Agile (Scrum), а также опыт внедрения таких практик, как Story Mapping, модульное тестирование и непрерывная интеграция.

В ходе профессиональной карьеры участвовал в проектах по внутренней автоматизации, заказной разработке программного обеспечения и разработке тиражируемых программных продуктов.

В настоящее время работает в ГК «Проектная практика» руководителем IT-направления.

## **Вера Алексеевна Иванова**

Опыт работы в IT — 14 лет. Первые семь лет работала программистом в государственных учреждениях. Приобрела опыт разработки прикладного ПО с использованием систем управления базами данных (СУБД), системного администрирования, технической поддержки пользователей. Участвовала во внедрении и сопровождении автоматизированных систем финансового учета регионального уровня.

В 2003 году начала осваивать новую специальность — системный аналитик — и продолжила работу в компаниях, специализирующихся на разработке программного обеспечения. Участвовала в разработке веб-приложений, программных продуктов по защите информации. Выполняла разные роли в проектах по разработке ПО: системного аналитика, разработчика БД, архитектора, тест-дизайнера, менеджера проекта. Основная специализация — системный аналитик. Последние четыре года специализируется в области методологии системного анализа и разработки ПО.

## **Сергей Николаевич Сергеев**

36 лет. Опыт работы в сфере разработки программного обеспечения — 15 лет. Занимался реализацией различных проектов и разработкой программного обеспечения для органов государственной власти. Работал программистом, руководителем группы, сейчас топ-менеджер в компании «Кейсистемс».

Занимался разработкой тиражного программного обеспечения в роли основного проектировщика баз данных, ведущего разработчика продукта, разработчика основной бизнес-логики, графического интерфейса. По роду решаемых задач довольно быстро переориентировался на руководство полным циклом производства тиражного программного обеспечения.

Среди задач, которые решал на пути профессионального развития, — организация системы менеджмента качества производства программного обеспечения, создание выделенного подразделения разработчиков промышленного ядра,

внедрение систем автоматизации тестирования продуктов, обоснование необходимости и создание подразделения по управлению маркетингом и PR, создание партнерского канала продаж, организация IT-интернатуры и др.

Последние пять лет занимается вопросами поиска и развития новых направлений в сфере разработки ПО и системной интеграции, включая создание команд с нуля.

Занимается общественной деятельностью по развитию IT-отрасли: развитием системы высшего образования, популяризацией IT как сферы деятельности, работой со студентами и школьниками.

# Благодарности

---

---

## Андрей Перерва

Я хочу поблагодарить всех читателей первой книги «Путь аналитика» за отзывы, письма и вопросы. Спасибо вам. Именно осознание того, что книга вам понравилась, была интересной и принесла реальную пользу, привело к созданию этой книги. Надеюсь, она будет не менее полезной.

Хочу сказать спасибо компании «Кейсистемс», а особенно Алексею Матросову, за решение спонсировать издание этой книги. Также благодарю Ольгу Бурминову за решение практически всех оргвопросов по книге и ее продвижению.

Благодарю моих соавторов за их профессионализм, идейность, готовность практически бескорыстно делиться знаниями, за проект «Путь IT-менеджера», который успешно завершается сейчас изданием этой книги. С вами приятно и интересно работать. Спасибо вам.

Хочу также сказать спасибо MBA-клубу Государственного университета управления за возможность делиться своим опытом и знаниями на базе высшей школы бизнеса в формате курсов, презентаций и тренингов. Общение с коллегами и слушателями дало серьезный импульс к качественной огранке материала книги.

Спасибо моей лучшей половинке за понимание, участие, поддержку и помощь. Спасибо за вдохновение, Гаш!

В заключение благодарю всех друзей и коллег за участие, советы и конструктивную критику.

## Сергей Еранов

В первую очередь хочу поблагодарить моих соавторов Андрея Перерву и Веру Иванову, пригласивших меня работать над этой книгой. Эта работа оказалась в разы сложнее, чем я мог ожидать, но и в разы интереснее и полезнее.

Также хочу поблагодарить своих коллег, бывших и настоящих, тех у кого я учился и учусь. Особенно хочется сказать спасибо коллегам по компании «ИнфоТекС». Считаю годы, проведенные там, решающими в моем профессиональном

развитии. Не могу не назвать конкретных людей: Владимир Николаевич Кокарев, Светлана Старовойт и Андрей Кирюшенков, спасибо вам! Не кривя душой могу сказать, что мой вклад в эту книгу основывается на опыте, приобретенном за время совместной работы именно с вами.

Спасибо Асхату Уразбаеву и Никите Филиппову за тренинги и консультации по Agile.

И, конечно, спасибо моей жене Наташе за бесконечное терпение, проявленное за время моей работы над книгой.

## **Вера Иванова**

Выражаю большую благодарность моим единомышленникам, особенно Андрею Перерве и Сергею Еранову.

Большое спасибо всем моим учителям, преподавателям, наставникам и тренерам за их вклад в мое профессиональное и личностное развитие.

## **Сергей Сергеев**

Хочу выразить благодарность моим коллегам, чья помощь и советы помогли мне найти идеи и практический материал для данной книги. Андрею Перерве и Ольге Бурмисовой — за идею написать «что-нибудь практическое и полезное для книги» и еще раз Ольге — за нашу совместную практику отбора специалистов, а также коллегам из «Кейсистемс», Сергею Соколову и Максиму Сорокину, которые вычитывали мои черновики.

# Введение

---

---

Что такое паттерны формирования проектной среды и управления? Что такое проектная среда? Бывают ли эти паттерны эффективными? Зачем они нужны? Как ими пользоваться и в каких конкретно случаях они эффективны, а в каких — нет?

Как выбрать оптимальный набор практик управления и разработки требований к информационным системам в зависимости от требований к качеству продукта, среды проекта, рисков проекта?

Как заставить команду работать? Как вывести проект в сегмент управляемого и предсказуемого проекта из портфеля проектов компании?

Что делать, когда проект не клеится? Как справиться со сложной проектной средой? Что это вообще за зверь такой — проектная среда и как с ней бороться?

Ну и, наконец, как выгодно продать себя как мастера управления проектной средой? Как объективно и максимально информативно раскрыть свой потенциал при поиске работы и непосредственно на интервью?

Ответы на эти и многие другие вопросы читатель найдет в книге.

Существует много замечательных методологий и стандартов проектного управления и не меньше замечательных книг, детально описывающих практики PMBOK, СММІ, СОВНЕТ, RUP. Эта книга не претендует на то, чтобы считаться методологическим справочником. Авторы не думают, что в ней досконально освещено то, что уже описано в других источниках.

Стремясь развить подход, использованный в первой книге серии «Путь IT-профессионала», авторы приводят только реально работающие и только проверенные ими на собственном опыте в реальных проектах хорошие практики, ссылаясь на рекомендуемые или базовые методологические первоисточники. Все практики систематизированы по блокам с целью помочь быстро найти нужные рекомендации и сразу же начать их использовать.

Функции управленца не сводятся к механическому применению стандартов и методологий. Искусство менеджера заключается в том, чтобы из их богатейшего арсенала выбрать именно те, которые максимально подходят данному проекту, его среде и данной организации в зависимости от уровня ее зрелости и квалификации персонала. Именно на этом в книге делается серьезный акцент.

С этой точки зрения миссия и цель издания — дать толчок к переосмыслению имеющегося опыта, пониманию того, каких знаний, возможно, не хватает, чтобы выработать глубинные принципы в управлении разработкой ПО и превратить их впоследствии в паттерны с четким пониманием критериев того, когда какой паттерн имеет смысл применять.

Эта книга — результат более чем 15-летней работы авторов в области разработки программного обеспечения и информационных систем. Главное ее достоинство — практическая направленность. В ней анализируются разнообразные кейсы и ситуации, приводятся примеры документов и шаблонов, необходимых для разработки ПО. Помимо этого, книга структурирует огромный массив теоретической информации в зависимости от того, что прошло проверку на практике и может быть использовано в работе профессионалов в области разработки программного обеспечения.

## Предыстория

В нашей книге «Путь аналитика. Практическое руководство IT-специалиста» (сайт книги — [bp4you.ru](http://bp4you.ru); страничка в Facebook — <http://www.facebook.com/pages/путь-аналитика/582969531717404>) мы с Верой Ивановой рассмотрели некую «дорожную карту» (road map) профессионального развития IT-специалиста.

Судя по полученным отзывам (см. <http://bp4you.ru/index.php/77-news/87-otzyvy2>) и письмам, мы достигли цели — книга позволила многим из вас получить нужные приемы и навыки и дала толчок как к развитию, так и к систематизации уже имевшихся знаний и умений. Мы очень рады этому! Крайне важно и приятно для нас осознавать, что поставленная цель была достигнута. Спасибо за отзывы и письма!

Мы надеемся, что эта книга поможет не наступать на те же грабли, на которые наступали мы, не повторять наших ошибок, а главное, поможет вам эффективно построить собственную профессиональную карьеру и повысить эффективность процессов управления и разработки ПО в качестве руководителя группы, менеджера проекта, начальника отдела, руководителя департамента по разработке ПО.

Мы прошли этот путь и накопили большой багаж хороших практик и выученных уроков, которые в этой книге пытаемся систематизировать и вручить вам. Именно поэтому мы назвали ее «Путь IT-менеджера. Управление проектной средой и IT-проектами».

## Для кого эта книга

Эта книга для профессионалов-управленцев в проектной или функциональной областях разработки ПО, руководителей рабочих групп, менеджеров проектов, начальников отделов/директоров департаментов по разработке ПО.

Книга будет полезна как начинающим менеджерам, так и менеджерам со стажем. Первые смогут найти в ней новые для себя практики, которые почти готовы к применению в реальной производственной жизни. Вторые смогут обобщить и систематизировать знания, критически посмотреть с разных сторон на собственный опыт и опыт авторов книги, возможно, пересмотреть какие-то паттерны или решить использовать какие-то наработки авторов в собственной практике.

Особо следует отметить, что в книге сделана попытка восполнить пробел для начальников подразделений/функциональных отделов, связанных с разработкой ПО. Существует довольно много литературы, описывающей практики и принципы проектного управления, но довольно слабо освещены аналогичные аспекты для функциональных менеджеров. В данной книге этот массив информации структурирован в группе паттернов управления подразделением разработки ПО.

Уникальность этой книги в том, что вы найдете в ней собранную в одном месте практическую информацию со ссылками на теоретические материалы из разных областей управления разработкой программного обеспечения: анализа, архитектуры, управления проектами, лидерства и управления персоналом — то, что вам понадобится в работе над реальными производственными проектами.

Общие паттерны — это паттерны, которые гарантированно могут применяться независимо от специфики IT и разработки ПО. Это не означает, что паттерны, структурированные в разделе «Паттерны IT», нельзя использовать с той же целью, — можно, но их придется адаптировать.

Мы надеемся, что эта книга поможет вам скорректировать и дополнить ваш личный план развития и максимально эффективно построить процессы управления и производства на вверенных вам направлениях. На сайте <http://bp4you.ru> вы можете скачать дополнительные материалы, в том числе некоторые шаблоны спецификаций и документов.

## Чего вы НЕ найдете в этой книге

В этой книге вы не найдете голый теории. Мы будем приводить только необходимый теоретический минимум, чтобы проиллюстрировать практики и методы.

Управление, анализ и остальные затрагиваемые в книге дисциплины разработки ПО — на сегодняшний день довольно зрелые дисциплины, и вам не составит труда найти материалы по ним в Интернете и учебники — в продаже.

Мы не ставим целью дать читателю теоретическое образование. Книга ориентирована на активного читателя, который уже понял, что теоретическую базу он должен искать и впитывать самостоятельно. Вы не найдете в книге также уверенности авторов в том, что правильны только описываемый путь менеджера и только предлагаемые авторами методики и приемы. Напротив, к одной

и той же цели ведут несколько путей. Мы хотим показать вам тот путь, который прошли сами, в котором уверены, дать только те паттерны, которые приводили нас к успеху на этом пути.

## Как читать эту книгу

Если изобразить серию книг «Путь IT-профессионала» в виде UML-диаграммы (рис. В.1), то основные концептуальные понятия данной книги на ней выделены серым цветом. Как видите, в этой книге мы с вами рассмотрим понятия проекта, типа проекта, жизненного цикла (ЖЦ) проекта и их связи с рисками проекта, требованиями к качеству продукта и проектной средой, а также вопросы, тесно связанные с перечисленными понятиями и сущностями.

Когда задумка написать эту книгу вылилась в попытки построить ее структуру так же, как структуру книги «Путь аналитика. Практическое руководство IT-специалиста», мы столкнулись с тем, что практически невозможно сделать это, отталкиваясь от какой-то квалификационной шкалы менеджера. Эта задача была непростой и в первой книге, несмотря на то что уровни квалификации в ней соотнесены с этапами карьерного развития. Непросто потому, что крайне сложно однозначно соотнести знания, навыки и умения с каким-то конкретным квалификационным уровнем, при этом не перегружая требования к квалификации. Должен ли младший аналитик знать вот эту теорию? А должен ли он уметь делать вот это? Или он это должен уметь все-таки на следующем уровне своего развития, уже как старший аналитик? Такие вопросы мы множество раз задавали сами себе, когда писали первую книгу. В итоге нашли некоторые границы квалификаций, которые более или менее отражают взгляд авторов на этапы профессионального и карьерного развития аналитиков.

Впрочем, мы всегда утверждали: даже несмотря на то, что эти требования реальными требованиями не являются и часто избыточны, подходите к ним с умом и выбирайте только то, что вам реально нужно в работе. Многие наши читатели так и поступали, о чем нам впоследствии и писали.

В этой же книге все еще сложнее. Нельзя сказать, что менеджер продуктов, например, является следующей ступенькой карьеры после менеджера проектов. Или что начальник отдела обязан иметь опыт управления проектом или развитием продуктов. Честно попытавшись сверстать первые версии книги в этом ключе, мы оставили эти безуспешные попытки и придумали другой подход — упорядочить информацию с помощью паттернов. Что понимается под паттерном в этой книге, вы узнаете позже, здесь же мы отметим только, что это некоторый блок информации, знакомиться с которым и в дальнейшем применять на практике который можно довольно независимо от других. Через набор таких паттернов мы и решили поделиться с вами нашим опытом и знаниями.

Мы также довольно долго думали, как упорядочить паттерны, и решили сделать это относительно субъектов управления. Таким образом, все паттерны в книге

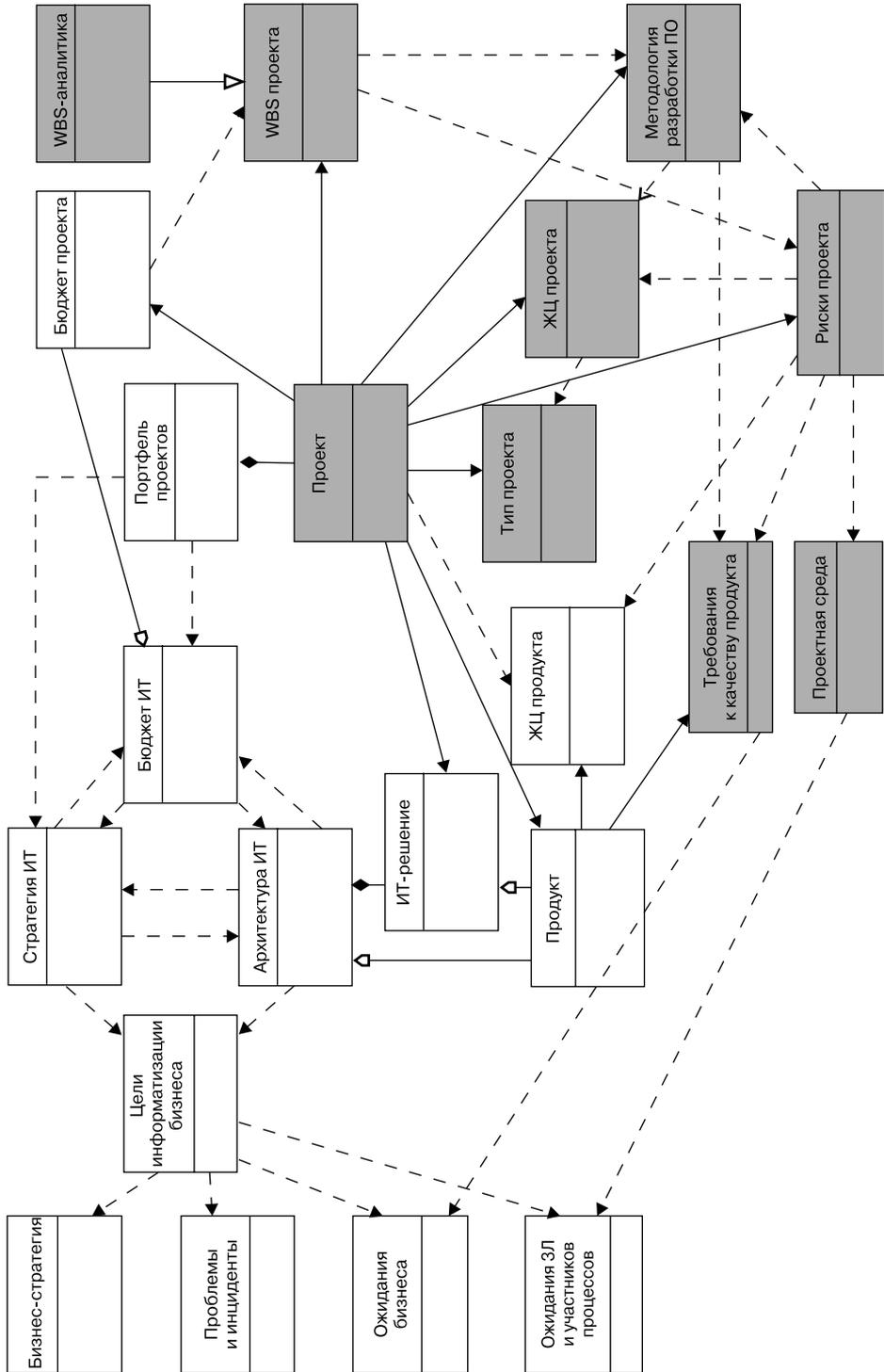


Рис. В.1. Основные концептуальные понятия книги

сгруппированы по какому-то единому целевому признаку и отсортированы внутри группы в «рекомендованном» порядке их применения. Благодаря этому, мы надеемся, начинающие управленцы смогут увидеть место каждого паттерна в, скажем так, возможной цепочке их применения, а также понять, каких знаний и опыта им не хватает для того, чтобы применять паттерны в «правильном», то есть рекомендуемом авторами, порядке.

Можете начать чтение с любого интересующего вас раздела любой главы в зависимости от своих целей и позиции в компании. Особенно важные места выделены особыми рисунками, значение которых приведено далее. А для того, чтобы проще было ориентироваться в материалах книги и паттернах в привязке к квалификации, в разделе «Сводная таблица паттернов» мы приводим таблицу всех паттернов, о которых говорится в книге, в зависимости от квалификации/профиля менеджера. Но эта зависимость, как вы уже понимаете, не является строгой, единственно правильной и тем более не может рассматриваться как какие-то квалификационные требования.

Главы «Общие понятия», «Что умеет мастер управления проектной средой» и разделы «Сводная таблица паттернов», «Плохие паттерны» каждой главы могут оказаться полезными и после прочтения книги и выступать как своеобразный навигатор по паттернам. Рекомендуем вам также активно работать с дополнительными материалами на сайте книги [br4you.ru](http://br4you.ru).

Глава «Как продать себя как мастера управления проектной средой» будет полезна, чтобы сформировать и успешно реализовать собственную стратегию поиска работы, и также содержит некоторые полезные паттерны.

*Примеры в книге помечены значком*



*Теоретическая информация помечена значком*



*Наиболее важные, с нашей точки зрения, методологические моменты, на которые мы рекомендуем вам обратить особое внимание, помечены значком*



Несмотря на то что в этой книге обобщен практический опыт, накопленный множеством людей, профессионально занимающихся бизнес- и системным анализом в проектах по производству ПО, дальнейшее повествование ведется от первого лица единственного числа для простоты изложения и восприятия.

Искусство управления крайне многогранно, я ни в коей мере не ставлю цель дать читателю «серебряную пулю», которой, как все мы знаем, не существует. Поэтому есть масса нюансов, которые я не смогу учесть, описывая те или иные паттерны и разбираться с которыми в плане применимости паттернов вам придется самостоятельно.

Так, например, в книге приводится некоторая классификация менеджеров, но надо отметить, что перечень нужных навыков и знаний будет сильно зависеть от того, в какой организации работает менеджер — проектной или функциональной, занимается организация разработкой коробочных продуктов или решений на заказ. А может, это вообще подразделение внутренней автоматизации. Существует также зависимость от того, крупная это компания или нет, какая в ней корпоративная культура, насколько менеджер проекта самостоятелен, имеет ли полномочия для принятия решения и пространство для маневра. Поэтому в книге вы, возможно, встретите какие-то категоричные, на ваш взгляд, суждения, например, о наличии у менеджеров определенных навыков, хотя мы все прекрасно понимаем, что границы между разными квалификациями менеджеров очень условны.

На определенную условность этих границ влияет и то, что очень сложно донести одну и ту же суть до людей разных степени подготовки и опыта: кто-то занят разработкой заказного ПО и практически ничего не слышал о продуктовом управлении; кто-то разрабатывает коробочные продукты и меньше времени уделяет анализу и управлению изменениями в ожиданиях конкретного заказчика; кто-то занимается внутренней автоматизацией бизнеса, и его задача сделать все «вчера» со средним качеством.

Именно поэтому я позиционирую книгу как паттерны. Паттерны, которые нуждаются в переосмыслении и адаптации в зависимости от конкретной ситуации.

Для простоты восприятия рекомендую вам забыть на время о своей работе и читать книгу так, будто вы менеджер некоторой абстрактной компании, в которой нет какого-то устоявшегося уровня зрелости, где численность и степень формализации процессов опускаются за вторичностью по отношению к сути паттернов, которая занимается и заказной, и коробочной разработкой, то есть в какой-то степени будто вы менеджер, который или уже работал, или, вполне возможно, завтра будет работать в иных условиях, среде и типе организации. Я же в паттернах буду описывать свое видение параметров среды (масштаб компании, уровень формализации процессов, уровень квалификационных стандартов), в которой данный паттерн будет работать, чтобы вам было проще адаптировать информацию под свой случай.

# 1. Общие понятия

---

---

В этой главе я остановлюсь на некоторых основополагающих понятиях, глубокое понимание которых является необходимым для мастера управления проектной средой. Эти понятия не будут являться новостью для вас, однако мы прекрасно знаем, что, прежде чем обсуждать какие-то более сложные информационные материи, нужно договориться о единых терминах и связях сущностей предметной области, чтобы вести дальнейший диалог на одной и той же понятийной платформе.

## 1.1. Что такое бизнес в этой книге

Как это ни парадоксально, но разные люди, говоря «бизнес», имеют в виду совершенно разные вещи. В самом широком и очень упрощенном смысле бизнес можно назвать проектом, а все вокруг этого бизнеса — законы, нишу рынка, конкурентов, инфраструктуру, финансы, кадры, процессы, продукты и их стратегии развития, отношения с потребителями и многое другое — проектной средой бизнеса.

В данной книге я рассматриваю бизнес с точки зрения управленца, находящегося внутри этого бизнеса, то есть являющегося частью общей проектной среды бизнеса. Это значит, что основное для нас — не цели и задачи бизнеса, не репутация и капитализация, не стратегия бизнеса, а выполнение поставленных бизнес-целей в части вверенного вам направления в сложившихся условиях с использованием имеющихся ресурсов, будь то рабочая группа, отдел, проект, департамент.



---

Упрощенно говоря, когда я в роли технического директора или директора разработки ПО говорю «бизнес», в 90 % случаев я имею в виду своих коллег из коммерческого блока, включая генерального директора, совместно с которыми мы определили цели на ближайшие 1–3 года и которые, находясь на границах проектной среды бизнеса, постоянно генерируют для моих целей изменения и корректировки, на которые я обязан оперативно и эффективно реагировать.

---

## 1.2. Что такое информатизация бизнеса и зачем это надо знать

Подробнее об этом вы можете узнать из книги [1]. Здесь я отмечу только некоторые основные моменты и дополню актуальными для этой книги выводами.

Прежде чем мы рассмотрим, в чем заключается информатизация бизнеса и ее связь с другими бизнес-понятиями, добавлю еще пару слов о бизнесе. На любой бизнес можно посмотреть как на проект, а если это проект, то у него есть своя проектная среда.

Итак, проектная среда бизнеса — по сути, то же самое, что и проектная среда. Естественно, что у топ-менеджеров свои инструменты и методологии, свои паттерны управления этой средой, но в любом случае они тоже менеджеры. И их задача — управлять. Управление неэффективно, если нет стратегии и глобального видения того, куда и зачем идет компания. Поэтому и разрабатывается бизнес-стратегия.

Бизнес-стратегия часто разрабатывается по нескольким направлениям: стратегии роста (продуктовая, повышения эффективности, капитализации, партнерства, построения сети), инновационные стратегии, конкурентные стратегии («гонка за лидером», дифференциации, лидерства в издержках, «война стандартов», выживания, поглощения). Все они преследуют свои цели. Цели бизнеса и цели бизнес-стратегии компании должны быть согласованы. Кроме того, бизнес-стратегия зависит от текущего и перспективного состояния рынка (на самом деле от гораздо большего количества параметров, это только часть проектной среды бизнеса).

После определения бизнес-стратегии соответствующих дивизионов компании, выполняющих свои функции (маркетинг, производство и т. д.), устанавливают способ достижения целей конкурентной стратегии. Такие стратегии называются функциональными и чаще всего создаются по направлениям «Маркетинг», «Финансы», «Производство/операционная стратегия», «Стратегия НИОКР», ИТ.



---

Информатизация бизнеса — комплекс мероприятий, направленных на то, чтобы модель бизнеса компании наполнялась с помощью ИТ реальным информационным содержанием и получала долгосрочные конкурентные преимущества.

---

Архитектура ИТ — это согласованная система представлений об организации информационной среды компании. Данная система должна соответствовать модели бизнеса, определять целевое состояние ИТ, быть основой долгосрочного планирования, обеспечиваться процессами и инфраструктурой ИТ. Совокупность модели бизнеса и архитектуры ИТ представляет собой корпоративную архитектуру. Чтобы разрабатывать эффективные приложения, управленец должен иметь представление обо всех аспектах проектной среды, то есть не только владеть информацией об ожиданиях от разрабатываемой системы, но

и понимать, какое место она занимает в ИТ-архитектуре собственной компании и компании заказчика, каковы тренды изменений этой архитектуры и бизнес-выгоды, которые принесет внедрение новой системы как собственной компании, так и компании заказчика.

Именно поэтому вы, как управленец, должны стремиться к тому, чтобы:

- ◆ иметь одинаковое видение с бизнесом;
- ◆ ясно понимать, как устроена проектная среда бизнеса вашей организации;
- ◆ иметь представление о том, как построена корпоративная архитектура вашей компании;
- ◆ формировать проектную среду вашего проекта, в том числе:
  - ◆ иметь представление о проектной среде бизнеса ваших клиентов или заказчиков и понимать, как она устроена;
  - ◆ иметь представление о корпоративной архитектуре ваших клиентов или заказчиков и понимать, как она построена.



---

Очень часто менеджер проекта считает, что это все нужно знать только топ-менеджерам, а его задача — просто выполнять поставленные задачи. Однако это не так.

Конечно, он может ограничиться только зоной видимости той дороги, что лежит непосредственно перед ним. Но даже в этом случае он обязан смотреть, куда эта дорога ведет и куда он, собственно, везет этот воз: действительно ли заказчик хочет попасть именно туда, нет ли более короткой дороги, что произойдет, когда он туда прибудет, что его организация будет делать дальше?

Говоря же об одинаковом видении с бизнесом, в общем случае я утверждаю, что ваш непосредственный руководитель и есть ваш бизнес-провайдер и если на какие-то вопросы нет ответов и у него, вы ему поможете тем, что выявите такие белые пятна.

---

## 1.3. Что такое проект, программа проектов и портфель проектов



---

**Проект** [2] (от лат. *projectus* — «брошенный вперед», «выступающий», «выдающийся вперед») — согласно новому стандарту ISO 21500, уникальный набор процессов, состоящих из скоординированных и управляемых задач с начальной и конечной датами, предпринятых для достижения цели. Достижение цели проекта требует получения результатов, соответствующих определенным заранее требованиям, в том числе ограничения на получение результатов, таких как время, деньги и ресурсы.

---

Проект имеет ряд характеристик, определив которые, можно точно сказать, относится ли анализируемый вид деятельности к проектам.

- ◆ Временность — любой проект имеет четкие временные рамки (это не относится к его результатам). В случае если таких рамок нет, деятельность называется операцией и может длиться сколь угодно долго.
- ◆ Уникальные продукты, услуги, результаты — проект должен порождать уникальные результаты, достижения, продукты, в противном случае такое предприятие становится серийным производством.
- ◆ Последовательная разработка — любой проект развивается во времени, проходя через определенные ранее этапы или шаги, но при этом составление спецификаций проекта строго ограничивается содержанием, установленным на начальном этапе.

Несмотря на то что результат выполнения проекта должен быть уникальным, он обладает рядом общих с процессным производством характеристик:

- ◆ выполняется людьми;
- ◆ ограничен доступностью ресурсов;
- ◆ планируется, исполняется и управляется.

Проекты могут быть объединены в программу проектов для достижения единого результата или в портфель проектов для более эффективного управления. Портфель проектов может состоять из программ.



Основная работа менеджера проекта (МП) должна заключаться в управлении изменениями проекта. Однако без управления изменениями на уровне бизнеса и без понимания того, как взаимосвязаны изменения текущей ситуации бизнеса с изменением ситуации в каждом конкретном проекте, управление изменениями в проекте может быть стрельбой из пушки по воробьям.

Для того чтобы этого не произошло, в организации существует управление портфелем проектов.



**Портфель проектов** [3] (англ. *project portfolio*) — это набор проектов, программ проектов и других работ, объединенных для достижения более эффективного управления и обеспечения выполнения стратегических целей организации.

Процессы управления портфелем проектов делятся на две группы: группа процессов выравнивания (*Aligning Process Group*), или формирования портфеля, и группа процессов мониторинга и контроля (*Monitoring Process Group*).



За более детальными определениями рекомендую обратиться к [2] и [3]. В этой же главе резюмируем главное.

- ◆ Цель портфельного управления — сделать правильную работу.
- ◆ Цель проектного управления — сделать работу правильно.

Что же требуется для того, чтобы определить, какую правильную работу нужно сделать? Для формирования портфеля проектов организации обычно выполняют следующие этапы и шаги [4].

### 1. Идентификация и описание.

- ◆ Определить область формирования портфеля проектов:
  - функциональная область — проекты по функциям компании, обычно по тем областям, которые наиболее важны в текущий момент, например по маркетингу, производству, ИТ и т. д.;
  - продуктовая область — проекты различных функциональных направлений, направленные на развитие определенных продуктов;
  - рыночная область — проекты различных функциональных направлений, объединенные в портфель, способствующий лучшему управлению определенным типом продукта/услуги;
  - географическая область — сводный портфель проектов. Формируют по каждому подразделению/дивизиону. Каждый из таких портфелей может состоять из портфелей трех перечисленных ранее типов.
- ◆ Определить принципы портфельного управления в каждой области.
- ◆ Сформировать паспорта текущих проектов, которые выполняются или запланированы.
- ◆ Сформировать паспорта идеальных проектов, которые следовало бы выполнить в отсутствии текущих проектов (о том, что такое идеальный проект, сказано позже).

### 2. Оценка.

- ◆ Сформировать и оценить первичный портфель проектов.
- ◆ Выполнить предварительный выбор проектов кандидатов в состав целевого портфеля.

### 3. Выбор.

- ◆ Сопоставить каждый идеальный проект с наибольшим количеством близких по смыслу текущих проектов
- ◆ Принять решение по трансформации текущих проектов и окончательному включению проектов в целевой портфель.

### 4. Оптимизация и балансировка.

- ◆ Актуализировать оценки проектов.
- ◆ Оптимизировать и сбалансировать портфель. При необходимости повторить мероприятия п. 3.
- ◆ Сравнить параметры целевого и первичного портфелей. При необходимости повторить мероприятия п. 3.

### 5. Авторизация. Авторизовать проекты.



Идеальный проект — оптимальный проект, который появился бы из цепочки «Бизнес-цели — Архитектура IT — Стратегия IT — Стратегические инициативы — Проект IT» в случае информатизации бизнеса или из цепочки «Бизнес-цели — Линейка продуктов — Стратегия развития и продвижения продуктов — Стратегические инициативы — Проект развития/создания продукта» — в случае компании, производящей ПО. О том, что такое паспорт проекта, и о его роли в проектном и портфельном управлении поговорим позже.

На рис. 1.1 показана взаимосвязь проектного и портфельного управления в контексте стратегического и операционного управления компанией.



**Программа проектов** [5] — это ряд связанных друг с другом проектов, управление которыми координируется для достижения преимуществ и степени управляемости, недоступных при управлении ими по отдельности.

Управление портфелем, программой и проектом представляет собой иерархическую структуру. На верхней ступени пирамиды стоит управление портфелем, которое содержит в себе все программы и проекты, обладающие приоритетами в зависимости от их бизнес-целей. Далее располагается управление программами, которое содержит множество взаимосвязанных проектов, поскольку они поддерживают определенные бизнес-цели. На нижней ступени находится проект.

В заключение хочу увязать между собой предыдущий и настоящий разделы.

Проект, программа проектов и портфель проектов не существуют сами по себе — это основные инструменты достижения задачи информатизации бизнеса. Сама специфика бизнеса обрекает проект на пассивное ожидание постоянных новшеств в результате изменений целей, задач и приоритетов бизнеса. Для того чтобы этого не случилось, и требуется управление портфелем проектов. Если в вашей организации по каким-то причинам этого не происходит, значит, вы должны как минимум суметь построить с бизнесом двухсторонний диалог и постоянно поддерживать его.

## 1.4. Что такое жизненный цикл продукта и проекта

Далее кратко рассмотрим связь понятий «портфель проектов» и «проект» с понятием «жизненный цикл продукта».

В книге [1] жизненному циклу продукта посвящен раздел «SDLC продукта: важность управления требованиями и основные методики инженерии требований к продуктам». В этой главе остановимся только на наиболее важных моментах ЖЦ продукта (рис. 1.2).

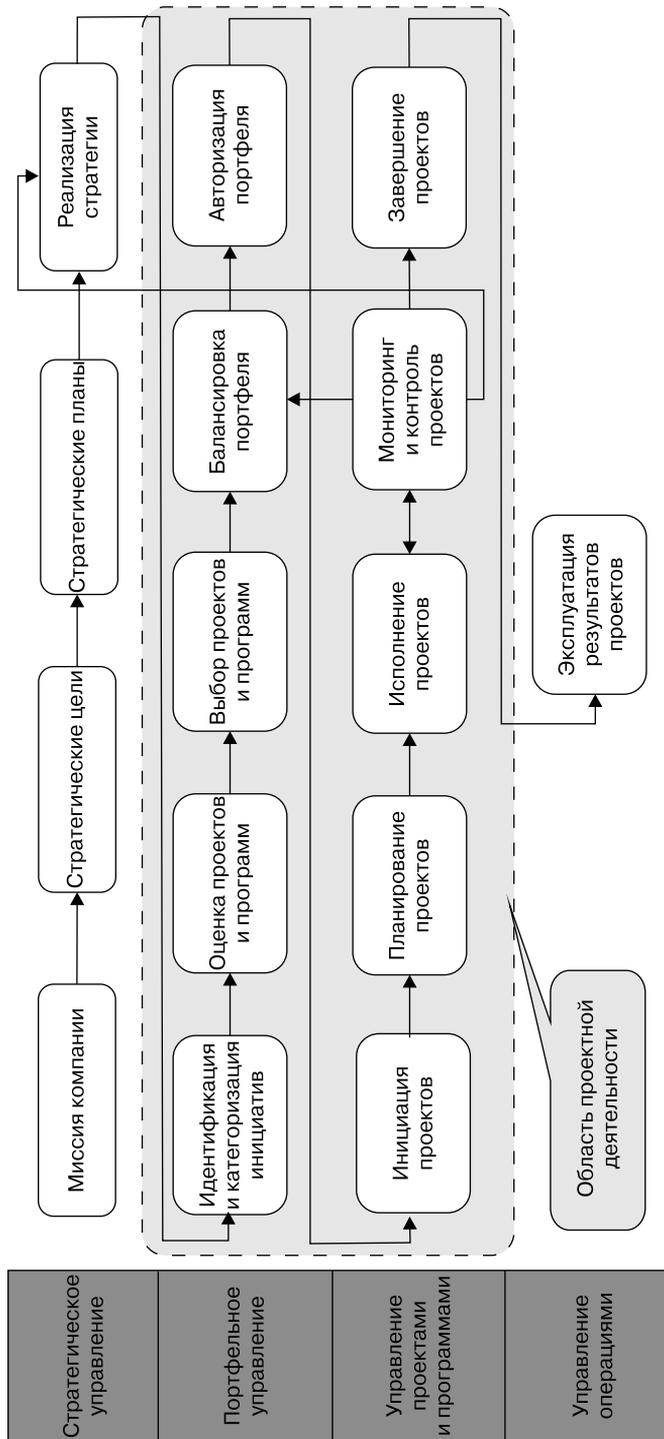
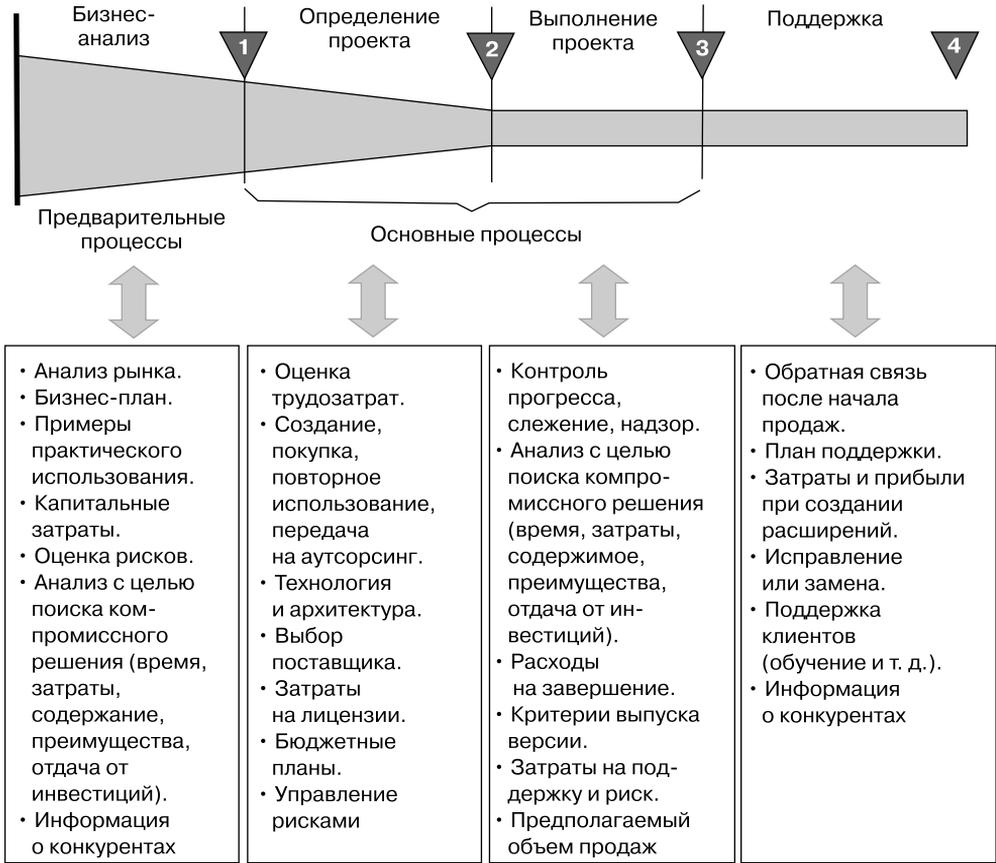


Рис. 1.1. Стратегическое, портфельное и проектное управление



**Рис. 1.2.** Упрощенный ЖЦ продукта с четырьмя точками принятия решения и предварительными (слева) и основными (справа) процессами [6]



В идеале для того, чтобы создать успешный продукт, необходимо определить потребности рынка и на их основе сформулировать цели и назначение продукта, которые затем реализуются в соответствии с принципами управления проектом.

Однако в реальной жизни основные задачи современного менеджера продуктов таковы.

- ◆ Предугадать будущие ожидания покупателя с помощью маркетинговых исследований.
- ◆ Если пока ожиданий нет, то к моменту выхода нового продукта сформировать у покупателя правильные ожидания.

За 1,5–2 года, которые проходят от начала разработки нового software product до момента его выпуска на рынок, продукт может частично устареть. Залог

успеха — выпустить продукт первыми любой ценой (качество, как правило, приносят в жертву срокам).

Управление продуктом — это контроль продукта с момента его замысла до выпуска с целью добиться максимальной пользы для компании. Требования — своего рода строительные блоки, объединяющие различные этапы ЖЦ продукта. Под жизненным циклом продукта мы понимаем совокупность всех операций, необходимых для того, чтобы описать, разработать, реализовать, скомплектовать, использовать, обслуживать продукт и прекратить выпуск его самого и его вариантов.

Цепочка «*Бизнес-цели – Линейка продуктов – Стратегия развития и продвижения продуктов – Стратегические инициативы – Проект развития/создания продукта*», описанная ранее как идеальный проект, и этапы «Бизнес-анализ» и «Определение проекта» аналогичны и являются частью процессов управления портфелем проектов. Если вспомнить, что на этапе «Идентификация и описание» первый шаг — это определение областей портфеля проектов и одним из признаков формирования портфеля проектов может выступать как раз продуктовая область, то в портфель включаются все проекты всех имеющихся в компании функциональных направлений, предназначенных для развития определенных продуктов. То есть это не только проекты разработки ПО, но и маркетинговые проекты, и проекты внедрения, и проекты сопровождения. Управляет таким портфелем проектов управляющий совет компании или непосредственно генеральный директор.



Понятие «ЖЦ проекта» в соответствии с одной из наиболее популярных методологий на сегодняшний день — PMBOK [7] в самом простом виде можно изобразить так, как на рис. 1.3.

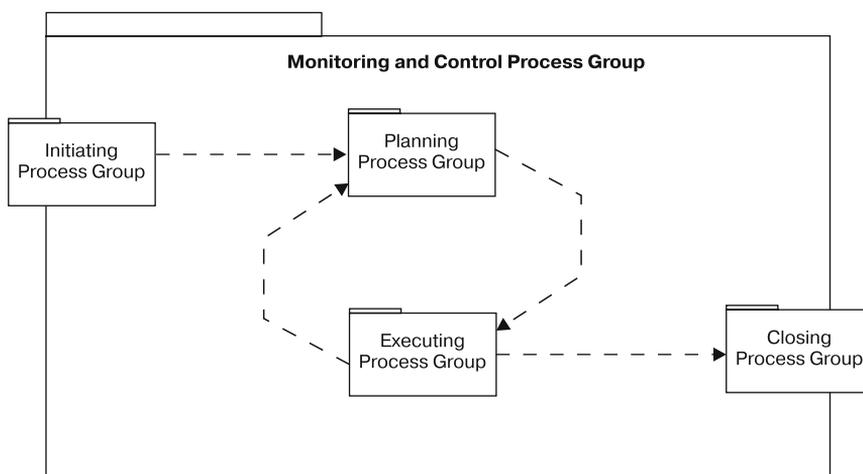


Рис. 1.3. Группы процессов PMBOK

Здесь представлены пять групп процессов.

- ◆ **Initiating Process Group (Группа процессов инициации).** Определяет и авторизует проект или фазу проекта.
- ◆ **Planning Process Group (Группа процессов планирования).** Определяет и уточняет цели и планирует действия, необходимые для достижения целей и содержания, ради которых был предпринят проект.
- ◆ **Executing Process Group (Группа процессов исполнения).** Объединяет человеческие и другие ресурсы для выполнения плана управления процессами данного проекта.
- ◆ **Monitoring and Control Process Group (Группа процессов мониторинга и контроля).** Регулярно оценивает прогресс проекта и осуществляет мониторинг, чтобы обнаружить отклонения от плана управления проектом и в случае необходимости провести корректирующие действия для достижения целей проекта.
- ◆ **Closing Process Group (Группа завершающих процессов).** Формализует приемку продукта, услуги или результата и подводит проект или фазу проекта к правильному завершению.

PMBOK выделяет 44 процесса управления проектами, разделяя их на пять групп процессов управления проектом и девять областей знаний по управлению проектами.

Интересно то, что многие активности в ЖЦ проекта пересекаются, но не покрывают все активности, определяемые методологиями разработки ПО (например, RUP). Также следует отметить, что некоторые активности методологий проектного управления будут менее эффективно работать или не будут работать вовсе при использовании определенных методологий разработки ПО. Например, план управления проектом в классическом его понимании принесет мало пользы в проектах разработки ПО, выполняемых по методологии Scrum.

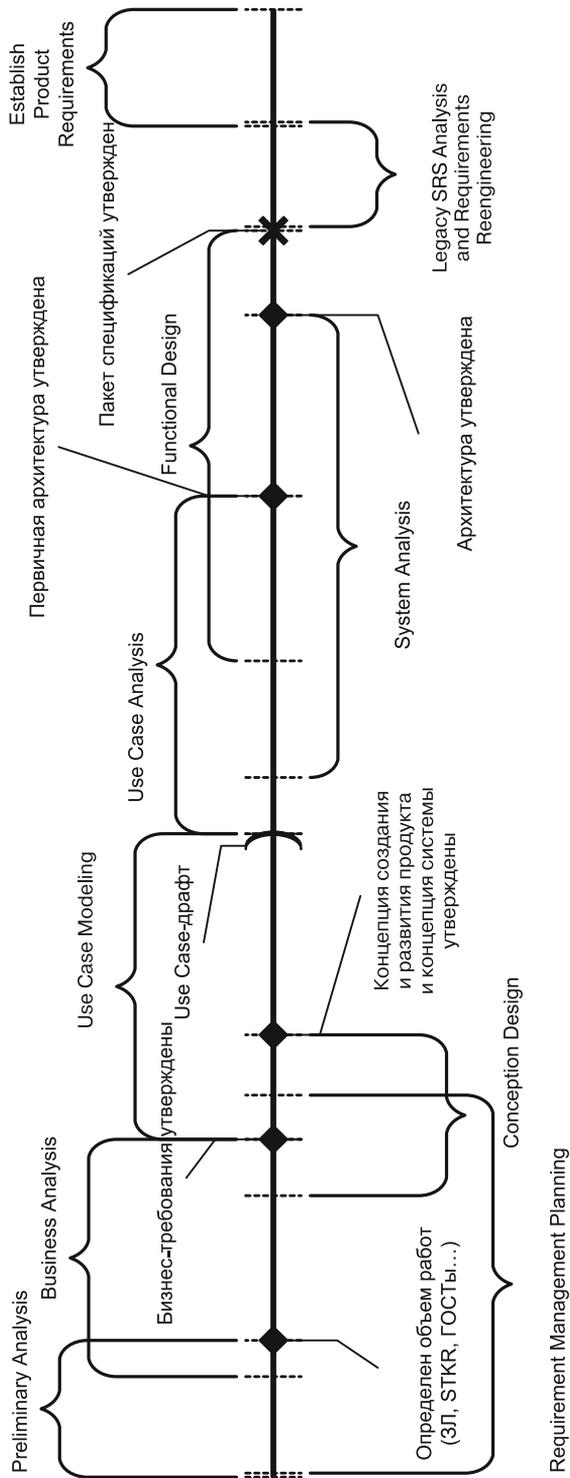
Детально различные ЖЦ проектов разработки ПО с точки зрения методологии разработки ПО мы рассматривали в книге [1]. В качестве примера приведу только водопадный ЖЦ разработки, который можно проиллюстрировать рис. 1.4.

С подробной информацией о данном и о других жизненных циклах можно ознакомиться на сайте <http://bp4you.ru>. Материалы по ЖЦ проектов также доступны на сайте [http://bit.ly/pmway\\_sdlc](http://bit.ly/pmway_sdlc).

Давайте вспомним и дополним то, о чем мы говорили раньше, предположив, что мы работаем в компании — производителе ПО, в которой разрабатывается несколько коммерческих продуктов.



Управление портфелем, программой и проектом ориентировано на достижение целей, определенных в бизнес-стратегии, и представляет собой иерархическую структуру. Ступени структуры пересматриваются на регулярной основе и влияют друг на друга.



**Рис. 1.4.** Иллюстрация последовательности этапов разработки требований

Requirement Management Planning

На верхней ступени иерархии стоит управление портфелем, которое содержит в себе все программы и проекты, обладающие приоритетами в зависимости от их бизнес-целей и стратегий развития продуктов.

Далее располагается управление программами, которое содержит множество взаимосвязанных проектов, поскольку они поддерживают определенные бизнес-цели и реализацию согласованных по времени и функционалу релизов разных продуктов.

И наконец, на нижней ступени находится проект, который реализует выпуск промежуточного или конечного релиза конкретного продукта.

Но, кроме управленческого и методологического аспектов, в цепочке наших рассуждений о понятиях есть и такой важнейший технологический аспект, как архитектура. О ней поговорим в следующем разделе.

## 1.5. Что такое архитектура, какой она бывает и зачем мне это надо знать



Существует много различных определений термина «архитектура». Все они имеют свои плюсы и минусы. Мне импонирует определение, данное в стандарте ANSI/IEEE Std 1471–2000: «Архитектура — это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы».

Архитектура предприятия, или корпоративная архитектура (Enterprise Architecture), — это высокоуровневая архитектура всего предприятия, удовлетворяющая бизнес-потребности с помощью ИТ-способностей. Корпоративная архитектура фокусируется на определении потоков и бизнес-процессов, действий, функций, информации, данных и технологий предприятия, а также на вызовах, стоящих перед ИТ, которые необходимы для того, чтобы эффективно применить технологию в ответ на изменение бизнес-потребностей.

Enterprise Architecture объединяет Business Architecture, Information Architecture, Solution Architecture и Technology Architecture [8] (рис. 1.5).

**Business Architecture.** Описывает все бизнес-процессы, бизнес-факторы, бизнес-сущности и бизнес-правила с точки зрения бизнеса. Не зависит от применяемых при разработке технологий.

**Information Architecture.** Определяет структуры данных и описывает все потоки данных, которые используются для поддержки Business Architecture. Такие операции, как идентификация, систематизация, категоризация, хранение данных, относятся к Information Architecture. Может представляться в виде Data Model.

**Solution (System/Application) Architecture.** Архитектура программного обеспечения, которое реализует функции Business Architecture.

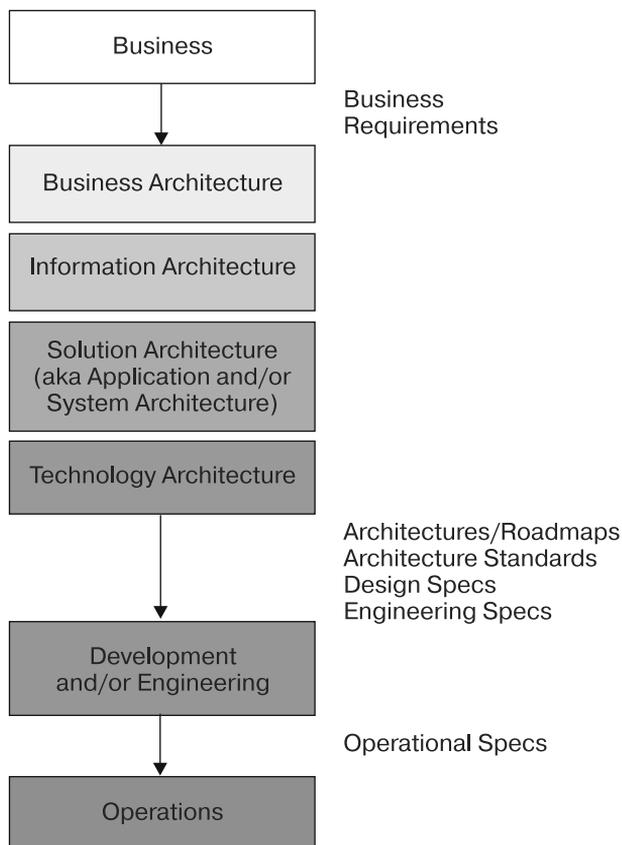


Рис. 1.5. Архитектура предприятия/корпоративная архитектура

**Technology Architecture.** Описывает архитектуру IT-окружения, которое используется для поддержки Information Architecture и Solution (System/ Application) Architecture.

Различные представления архитектуры (Business Architecture, Information Architecture, Solution Architecture и Technology Architecture) тесно взаимосвязаны, и их интеграция — необходимое условие успешного построения Enterprise Architecture.

С точки зрения разработки ПО наиболее важное понятие — Solution (System/ Application) Architecture, составными частями которой являются System Architecture, Data Architecture и Software Architecture.

**System Architecture.** Представление системы, которое показывает реализацию функциональных возможностей системы аппаратными средствами и компонентами программного обеспечения, устанавливает связь архитектуры программного обеспечения и архитектуры аппаратных средств, а также регламентирует человеческое взаимодействие с этими компонентами. Существуют и другие

определения System Architecture, например: «Ряд взаимосвязанных шаблонов (паттернов), которые структурируют модули и данные и обеспечивают требуемое поведение системы (см. определение Data Architecture)». System Architecture является составной частью Solution Architecture.

**Software Architecture.** Составная часть System Architecture. Описывает организацию системы с точки зрения программных компонентов, из которых она состоит, и связи между компонентами.

**Data Architecture.** Составная часть System Architecture. Описывает структуры данных и логические связи между данными (Hierarchical Model, Network Model, Relational Model, Object Model и т. д.).

Мастер управления проектной средой должен понимать смысл приведенных понятий, их предназначение и взаимосвязь. С точки зрения развития продукта (для СЮ, СТО, руководителя разработкой ПО, руководителя проектного офиса) важно понимание Enterprise Architecture, разработки продукта — Software Architecture, System Architecture, Data Architecture. С точки зрения успешного выполнения проекта менеджеру проекта как минимум нужно знать и понимать System Architecture, Data Architecture, а также иметь представление о Solution (System/ Application) Architecture.

Теперь мы готовы перейти непосредственно к паттернам и хорошим практикам, но прежде сохраним преемственность серии «Путь IT-профессионала» и представим профиль менеджера по аналогии с профилем аналитика в первой книге.

## 2. Профиль и квалификация менеджеров

---

---

В настоящий момент существует как минимум несколько взглядов на квалификацию менеджеров. Работы в этом направлении ведутся довольно давно, и еще в 2004 году вышла статья [http://www.iteam.ru/publications/project/section\\_39/article\\_2209/](http://www.iteam.ru/publications/project/section_39/article_2209/), в которой описываются различные принципы квалификации менеджеров.



---

Национальная ассоциация управления проектами СОВНЕТ проводит международную сертификацию специалистов по управлению проектами — установление соответствия профессиональных знаний, опыта, навыков и поведения кандидата международным и национальным требованиям к компетентности специалиста по управлению проектами. Четырехуровневая модель сертификации IPMA (Four Level Certification (4-L-C)) разработана как процесс подтверждения компетентности специалистов в сфере управления проектами (<http://www.sovnet.ru/specialists/ipma-certification/>).

---

В данной главе я приведу таблицу, в которой отражен собственный взгляд на одну из возможных квалификационных градаций менеджеров, а материал данной книги представлен в виде навыков, необходимых менеджерам разной квалификации. Соответственно, эта глава может использоваться как своеобразный навигатор по книге.

Но перед этим давайте поговорим об одном моменте, который я считаю важным для получения ответа на вопрос, использованный в качестве заголовка следующего раздела. А именно...

### 2.1. Надо ли менеджеру быть аналитиком?

Довольно часто спонсор современного проекта разработки ПО ожидает, что менеджер проекта будет выступать также как аналитик проекта. А как вы

думаете, надо ли менеджеру быть аналитиком? Возможно ли это в принципе? Есть ли в этом практический смысл и принесет ли это пользу проекту?

Первоочередные и главные компетенции менеджера — быть хорошим управленцем, администратором и лидером, но он должен обладать и определенными аналитическими навыками: уметь мыслить аналитически, обрабатывать и структурировать информацию с разных точек зрения, ясно и последовательно развивать идеи и соображения, выстраивая причинно-следственную связь в логике своих рассуждений, детализируя и умея отделять главное от второстепенного, картинку крупным планом — от неактуальных в конкретный момент мелочей.

Если же в арсенале менеджера, кроме аналитических навыков, имеются инструменты и методы анализа, это повышает его способность к всестороннему пониманию информационных потоков, а значит, и к лучшему управлению ими как вовне проекта, так и внутри него. Сказанное, конечно же, не означает того, что менеджер в проекте *должен* выполнять роль аналитика или контролировать работу аналитиков глубже, чем работу других проектных ролей. Но это значит, что он как минимум *умеет* анализировать информацию и владеет инструментами анализа, а в случае разработки ПО хорошо, если он владеет еще и инструментами моделирования.



Степень аналитической составляющей в деятельности менеджера во многом зависит от масштаба проекта и поставленных задач, сложности управленческой модели и его занятости именно как управленца. Чем выше позиция управленца в бизнесе, тем более концептуально он должен уметь мыслить, уметь принимать стратегические решения, основываясь на опыте, знаниях, интуиции при неполных и не всегда достоверных данных. Для концептуального уровня характерна менее детальная проработка исходных данных и ограничений, чем для аналитического, и умение принимать решения на этом уровне — это искусство, которое приходит с опытом. Но если управленец умеет принимать решения, анализируя детальную информацию и риски, на аналитическом уровне, то такому профессионалу будет проще принимать решения и на концептуальном уровне.

Верно сказанное и для менеджеров старшего звена (senior management). Например, сложно предположить, что менеджер программы проектов по созданию системы управления ресурсами предприятия, включающей в себя 5–7 больших проектов, с общей численностью команды порядка 50–70 человек будет выполнять еще и функции аналитика. Даже если он в совершенстве умеет это делать, в данном случае это не его роль, а каждый член команды должен заниматься своим делом. Но если это проект по разработке довольно простого программного продукта с общей численностью проектной команды 7–15 человек, то функция ведущего аналитика вполне может быть объединена с функцией руководителя проекта. А если это проект по внедрению существующего решения на объекте с минимальной его адаптацией и доработкой, то зачастую функцию аналитика таких проектов выполняют менеджеры.

В любом случае управленцы всех уровней должны уметь управлять информацией — ожиданиями заинтересованных лиц и бизнес-требованиями, уметь анализировать информационное поле проекта и проектную среду. Без быстрого и всестороннего анализа менеджер будет управлять этой средой менее эффективно. Именно поэтому я отвечаю на вопрос: «Надо ли менеджеру быть аналитиком?» — так: «Менеджер должен иметь сильные аналитические навыки и быть аналитиком в широком смысле этого понятия».

В то же время, когда лично я ищу менеджера проекта или начальника отдела для решения задач своего бизнеса или задач моей компании, я всегда обращаю внимание не только на его управленческие, но и на аналитические компетенции. В конце концов, с человеком аналитического склада ума, данного ему от природы или же приобретенного с опытом, проще и эффективнее решать сложные вопросы и задачи. Он структурирует, умеет строить причинно-следственные связи, отделять второстепенное от главного, управлять ожиданиями и изменениями, ясно и четко излагать свои мысли.

А теперь давайте рассмотрим навыки менеджеров в квалификационной прогрессии «с высоты птичьего полета», то есть не вдаваясь в детали и глубину.

## 2.2. Профиль менеджера

В книге «Путь аналитика» под профилем аналитика я понимал составляющие, приведенные на рис. 2.1.

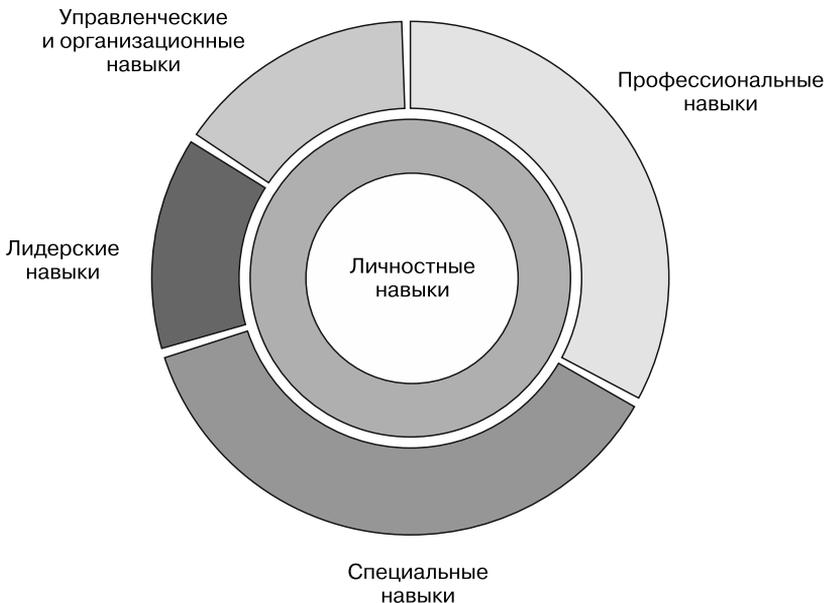


Рис. 2.1. Профиль аналитика

Здесь я позволю себе процитировать свою же книгу [1] для объяснения сути нотации этого профиля. Итак.

*«В основе профиля аналитика лежат личностные навыки. Без развития нужных личностных навыков вы не состоите в этой профессии. Будьте готовы к тому, что вам придется постоянно чему-то учиться. И по мере узнавания нового вы будете понимать, что знаете слишком мало – этакая здравая энтропия профессии. Выбрав путь аналитика, вы выбираете определенный стиль мышления, общения, развития и самой жизни в целом. Личностные навыки – это красный свет светофора на пути аналитика. Без наличия и развития некоторых навыков лучше вообще уйти из этой профессии.*

*Первый внешний сегмент в профиле аналитика – профессиональные навыки. Желтый свет этого сегмента – это деньги, которые вы тратите на книги, это ваши инвестиции в самого себя, это желтый свет светофора. Получив нужные профессиональные знания и навыки, вы уже сможете продавать себя на рынке.*

*Но выгодно продавать себя вы сможете, только получив специальные навыки, эти навыки приобретаются в ходе практических работ, выполненных реальных проектов, часто путем проб и ошибок. Это ваш зеленый свет на пути аналитика к дальнейшему развитию и это цвет денег, которые вы сможете зарабатывать, уже не просто продавая себя на рынке, но выгодно продавая себя».*

Все сказанное, на мой взгляд, характерно и для профиля менеджера, который я представляю таким (рис. 2.2).



**Рис. 2.2.** Профиль менеджера

Наиболее важной составляющей в профиле менеджера являются лидерские навыки. Вы можете не иметь богатого и сложного опыта, не знать многих управленческих приемов, можете вообще в первый раз получить в свои руки проект.

Но если в вас сильно лидерское начало, если вы стремитесь расти постоянно и знаете как минимум паттерн «Основные принципы проектного управления», то сможете управлять проектом.

За лидером идут. В том числе и более опытные и старшие товарищи. Если вы при этом достаточно умны, чтобы учиться у окружения, то у вас гораздо больше шансов на то, что ваш проект окажется успешнее, чем у менеджера с хорошим управленческим образованием, сертификатом по управлению и опытом выполнения нескольких проектов, но с менее выраженными лидерскими качествами.

Именно поэтому составляющая «лидерские навыки» самая большая в профиле менеджера. Однако, как и многие другие утверждения в этой книге, все зависит от ситуации, и в профиле администратора проекта, например, данная составляющая будет минимальной, потому что задачи координирования и администрирования проектом в меньшей степени требуют лидерских качеств, чем управление проектом.

Книга, которую вы держите в руках, в основном про «зеленый свет». Я хочу поделиться с вами проверенными специальными навыками, которые я структурировал и объединил в группы паттернов. Я не ставлю перед собой ни задачу научить вас управлению, ни задачу научить вас лидерству. Напротив, я ставлю задачу через готовые к использованию паттерны подвигнуть вас к самостоятельному получению нужных теоретических знаний и переосмыслению паттернов на базе этих знаний и вашего собственного уникального опыта. То есть в идеале у вас после прочтения книги и определенной самостоятельной работы сформируется собственный уникальный набор паттернов управления на все случаи жизни.



---

Я рекомендую для развития своих управленческих навыков обратиться в первую очередь к РМВОК как основе (смотрите также паттерн «Простой РМВОК» в этой книге), а для развития навыков в части анализа — к книге «Путь аналитика. Практическое руководство IT-специалиста». Для развития лидерских навыков стоит обратиться к [http://www.cecsi.ru/coach/leadership\\_attributes.html](http://www.cecsi.ru/coach/leadership_attributes.html). Там вы найдете массу ссылок на заслуживающие самого пристального внимания ресурсы и книги.

---

Давайте теперь посмотрим, как изменяется профиль менеджера в зависимости от выделенных в книге квалификаций (рис. 2.3–2.7).

Несмотря на то что на диаграммах сегменты профессиональных и специальных навыков закрашены полностью, всегда остаются белые пятна, так как технологии и методологии не стоят на месте и постоянно развиваются. Поэтому любой профессионал должен непрерывно работать над всеми составляющими своего профиля, чтобы не вернуться на предыдущий уровень. Также обратите внимание на то, что профиль от квалификации к квалификации становится объемнее и весомее. То есть происходит качественное и экстенсивное развитие.

Какие же навыки, знания и паттерны нужны менеджеру на каждом этапе? Об этом в следующем разделе.

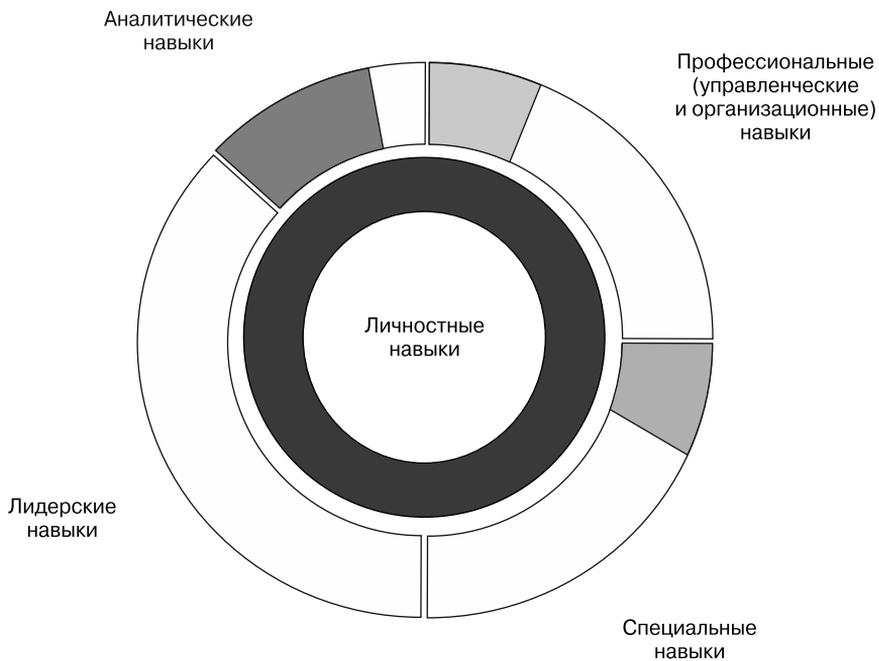


Рис. 2.3. Администратор проекта

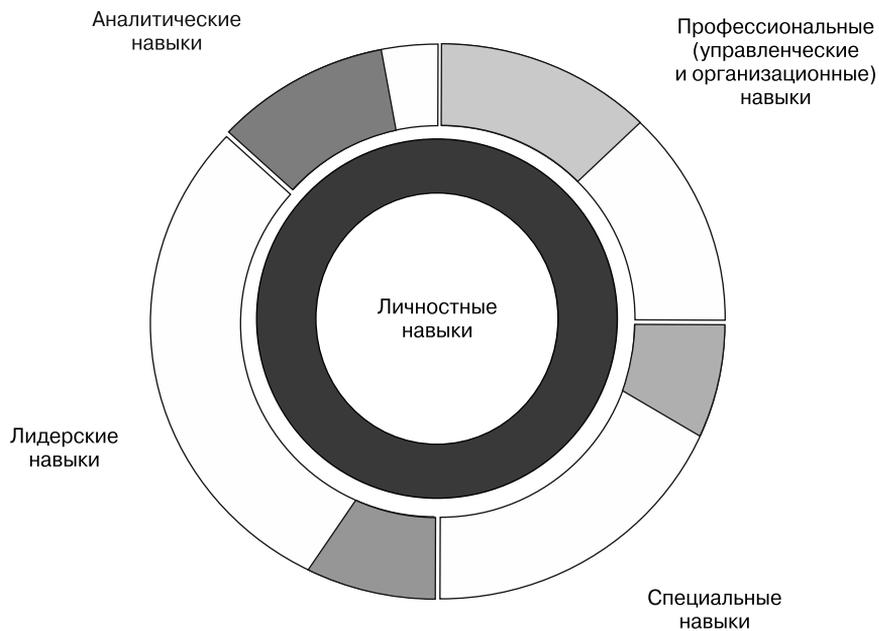


Рис. 2.4. Младший менеджер проекта/релиз-менеджер

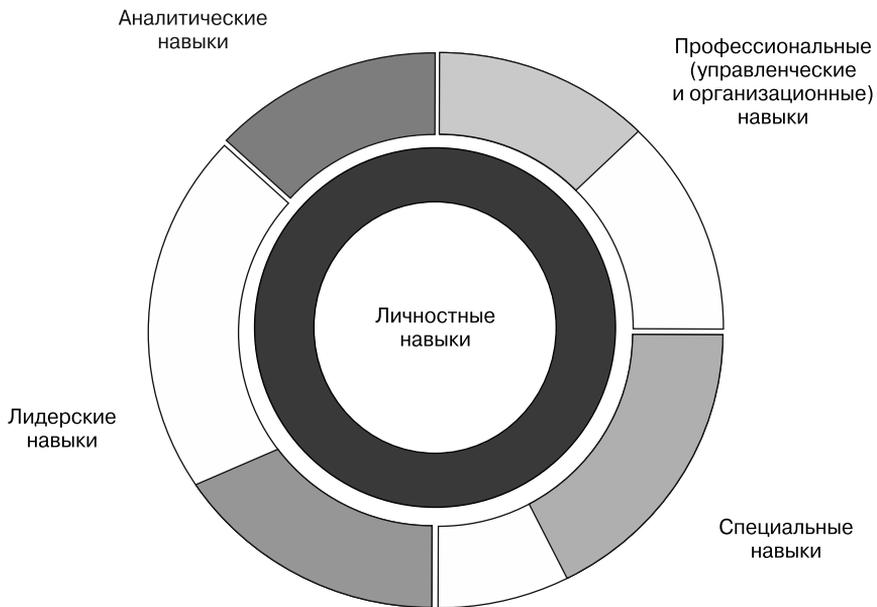


Рис. 2.5. Менеджер проекта

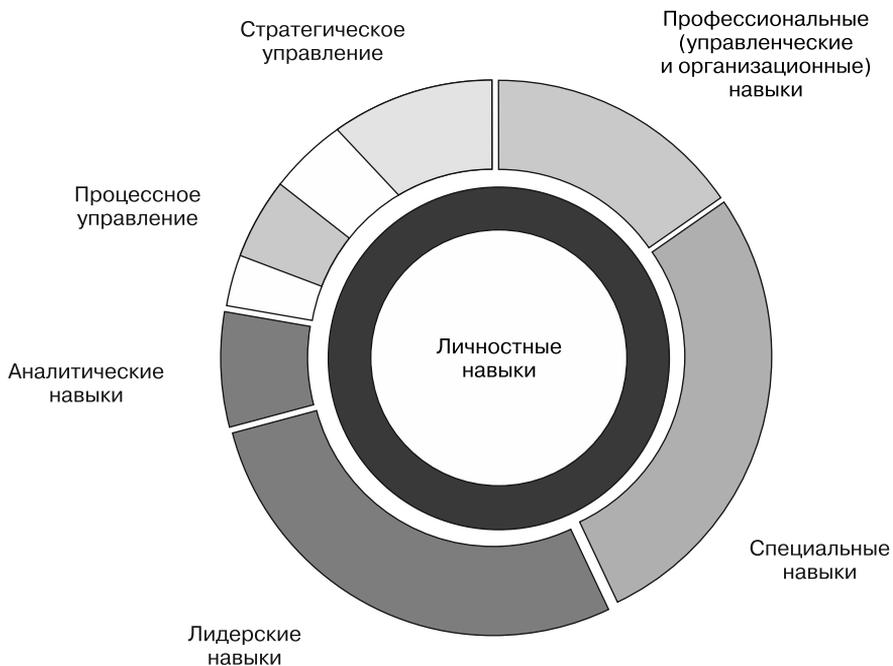


Рис. 2.6. Менеджер программы проектов/менеджер продукта



Рис. 2.7. Руководитель разработки ПО

## 2.3. Таблица квалификационных навыков

В табл. 2.1 я привожу классификацию менеджеров, но надо отметить, что перечень нужных навыков и знаний будет очень сильно зависеть от того, в какой организации работает менеджер, проектной или функциональной, занимается организация разработкой коробочных продуктов или создает решения на заказ. А может, это вообще подразделение внутренней автоматизации? В крупных компаниях более жесткая иерархия, сильнее выражена специализация. Там есть эксперты, которых можно привлечь для решения специфических задач. И МП в большой компании — управленец, который работает согласно процедурам и приказам, он может позволить себе не обладать экспертными знаниями. В средних и маленьких компаниях МП должен быть экспертом, чтобы стать успешным. Требования к нему как к управленцу часто менее строги.



Кроме того, многое зависит от корпоративной культуры. Регламентирована ли работа проектного офиса, есть ли он в компании в принципе, есть ли у менеджера, кроме ответственности, полномочия и инструменты, достаточные для управления проектом? Есть ли у него рычаги влияния и насколько они формальны, или ему требуется в каждом конкретном случае решать проблемы и тушить пожары преимущественно за счет своей харизмы и лидерских качеств?

Границы между разными квалификациями менеджеров сильно зависят от того, какие в принципе типы управления выделены в компании и как именно функция управления распределяется по управленческому сегменту. Так, например, если в компании есть менеджеры продукта и менеджеры программы проектов, то, скорее всего, менеджер продукта будет более жестко, чем указано в таблице, ориентирован в сторону бизнес- и клиентской направленности, а также архитектуры продукта, а менеджер программы проектов — в сторону административно-управленческую. В этом случае, строго говоря, нельзя сказать что, прежде чем стать менеджером продукта, вы должны стать менеджером программы проектов, так как функция управления продуктом разведена между двумя типами менеджеров, зоны ответственности которых имеют довольно четкие границы.

Часто бывает и так, что под менеджером продукта в компаниях подразумевают бизнес-аналитика, способного управлять бизнес-требованиями и ожиданиями заказчика.

Ну и последние две квалификации в таблице — руководитель функционального подразделения и руководитель подразделения разработки ПО — несколько выбиваются из иерархии развития менеджеров. Но я посчитал возможным объединить в таблице эти роли в компании, чтобы показать примерное развитие конкретного специалиста, проходящего по всем карьерным ступеням.

Обратите внимание на то, что курсивом в таблице я выделяю те навыки и знания, которые считаю *опциональными, необязательными, если указанные методологии/приемы не применяются в компании, но рекомендуются к изучению.*

**Таблица 2.1.** Квалификационная таблица менеджеров

Квалификация	Теория и профессиональные навыки	Специальные, лидерские и управленческие навыки
1	2	3
Администратор проекта	<p>Книги Leffingwell D., Widrig D. Managing Software Requirements я называю библией аналитика. Но несмотря на то, что акцент в книге сделан на разработке требований, вы найдете здесь и ряд конкретных полезных для управления проектом примеров и приемов проведения интервью, мозговых штурмов и т. д.</p> <p>Кроме того, по моему глубокому убеждению, менеджер обязан глубоко знать предмет этой книги, так как любой проект работает с требованиями и управление требованиями (обратите внимание — управление, не выявление требований) — основное, что должен делать менеджер совместно с профессиональными аналитиками.</p> <p>Иметь общее представление о различных методологиях разработки ПО.</p>	<p>Выявлять заинтересованных лиц (ЗЛ).</p> <p>Управлять ожиданиями ЗЛ.</p> <p>Проводить:</p> <ul style="list-style-type: none"> <li>• собрания;</li> <li>• интервьюирование;</li> <li>• анкетирование;</li> <li>• мозговые штурмы.</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• описывать концепцию системы;</li> <li>• определять границы системы;</li> <li>• работать с управленческими артефактами.</li> </ul> <p>Помогать МП в планировании работ.</p>

Продолжение ⇨

Таблица 2.1 (продолжение)

1	2	3
	<p>Знать основы RUP.</p> <p>Иметь представление о РМВОК</p>	<p>Контролировать выполнение работ.</p> <p>Понимать основные принципы тестирования.</p> <p>Знать английский язык на уровне, достаточном для чтения технической литературы</p>
<p><b>Младший менеджер проекта/релиз-менеджер</b></p>	<p>Все теоретические знания администратора проекта.</p> <p>Разбираться в таких методологиях разработки, как водопадная модель, итерационная модель, RUP, Iconix, Agile. Понимать принципы работы Scrum Team.</p> <p>Знать основы РМВОК, касающиеся управления границами проекта (Scope Management), качеством (Quality Control and Quality Assurance), изменениями (Change Management and Change Control Board), планом-расписанием проекта (Time Management), ресурсами проекта (Human Resource Management).</p> <p><i>Знать, как применять UML, и уметь делать это, а именно читать Use Case-модель, уметь читать и в идеале строить Domain Object Model.</i></p> <p>Понимать основные принципы объектно-ориентированного проектирования и моделирования.</p> <p>Рекомендую регулярно посещать сайты <a href="http://www.interface.ru/">http://www.interface.ru/</a>, <a href="http://www.pmprofy.ru/pmstart.asp">http://www.pmprofy.ru/pmstart.asp</a></p>	<p>Иметь все навыки администратора проекта.</p> <p>Помогать МП определять границы релиза.</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• определять цели релиза и требования к качеству;</li> <li>• строить и мотивировать команду;</li> <li>• проводить презентации.</li> </ul> <p>Отслеживать достижение целей и качества релиза через WBS проекта.</p> <p>Знать, как строить WBS, и уметь делать это.</p> <p>Знать, как оценивать трудоемкость и стоимость выполнения работ, и уметь делать это.</p> <p>Знать, что такое план-расписание проекта, и уметь его разрабатывать, мониторить и управлять изменениями.</p> <p>Понимать, какие модели существуют и каково их место в разработке ПО</p>
<p><b>Менеджер проекта</b></p>	<p>Все теоретические знания младшего менеджера проекта/релиз-менеджера.</p> <p><i>Знать и уметь применять ГОСТы 34-й и 19-й серий.</i></p> <p><i>Знать и уметь читать нотацию IDEF0, диаграммы eEPC (extended Event Process Chain).</i></p> <p>Иметь представление о СММІ.</p> <p>Расширять свой кругозор в методологиях. Познакомиться с гибкими методологиями, например Iconix.</p> <p>Книги: «Знать РМВОК»; A Guide to the Project Management Body of Knowledge. ANSI/PMI; PMP® Exam Prep; Mulcahy R. PMP; Д. Де Карло «Экстремальное управление проектами»; Де Марко Т., Листер Т. «Вальсируя с Медведями: управление рисками в проектах и по разработке программного обеспечения»; Де Марко Т. «Deadline. Роман об управлении проектами»;</p>	<p>Иметь все навыки младшего менеджера проекта/релиз-менеджера.</p> <p>Уметь:</p> <ul style="list-style-type: none"> <li>• выявлять цели и требования к качеству проекта и управлять ими;</li> <li>• анализировать среду проекта, управлять ею, влиять на ее формирование;</li> <li>• совместно с архитекторами и аналитиками определять и адаптировать методологии разработки ПО в конкретном проекте;</li> <li>• применять на практике РМВОК;</li> <li>• сверстать бюджет проекта и управлять им;</li> </ul>

1	2	3
	<p>Rosenberg D. Agile Development with Iconix Process.</p> <p>Рекомендую регулярно посещать сайты <a href="http://www.happy-pm.com">www.happy-pm.com</a>, <a href="http://www.iteam.ru/">http://www.iteam.ru/</a>, <a href="http://www.pmpofy.ru/pmexpert.asp">http://www.pmpofy.ru/pmexpert.asp</a>, <a href="http://www.cecsi.ru/">http://www.cecsi.ru/</a>.</p> <p>Рекомендую зарегистрироваться в сети профессиональных контактов (например, <a href="http://moikrug.ru">moikrug.ru</a>, linked), участвовать в конференциях, форумах.</p> <p>Рекомендую пройти несколько курсов подготовки к получению сертификата PMI или COVNET</p>	<ul style="list-style-type: none"> <li>• выявлять риски и управлять ими;</li> <li>• профессионально проводить презентации;</li> <li>• проводить уроки по всем проектным практикам;</li> <li>• предотвращать и разрешать конфликты в проектах;</li> <li>• проводить аттестацию сотрудников.</li> </ul> <p>Иметь:</p> <ul style="list-style-type: none"> <li>• детальное представление о жизненном цикле проекта и продукта.</li> <li>• собственное представление о полезности использования практик PMBOK.</li> </ul> <p>Знать, что такое план управления проектом (ПУП), и уметь его разрабатывать.</p> <p>Знать, что такое план управления документами (ПУД), и уметь его создавать.</p> <p>Быть лидером</p>
<p><b>Менеджер программы проектов</b></p>	<p>Все теоретические знания менеджера проекта.</p> <p>Книги: Boehm V. W. A Spiral Model of Software Development and Enhancement; Capability Maturity Model for Software. SEI.</p> <p>Рекомендую регулярно посещать сайты <a href="http://www.cecsi.ru/">http://www.cecsi.ru/</a>, <a href="http://profyclub.ru/section/65/">http://profyclub.ru/section/65/</a>.</p> <p>Рекомендую в сети профессиональных контактов (например, <a href="http://moikrug.ru">moikrug.ru</a>, linked) заниматься просветительской деятельностью, участвовать в конференциях, форумах.</p> <p>Рекомендую получить сертификат PMI или COVNET</p>	<p>Иметь все навыки менеджера проекта.</p> <p>Формировать концепцию программы проектов, уметь взаимодействовать с бизнес-дивизионами и конечными потребителями.</p> <p>Определять цели и требования к качеству релизов продукта в рамках программы проектов.</p> <p>Влиять на формирование проектной среды в проектах программы/развития продуктов и управлять ею в рамках программы/развития продуктов.</p> <p>Бюджетировать программу проектов.</p> <p>Строить финансовые модели.</p> <p>Управлять работой/участвовать в работе Change Control Board (CCB) в рамках проектов по программе проектов.</p> <p>Быть активным участником офиса управления проектами организации.</p>

Таблица 2.1 (продолжение)

1	2	3
		Иметь представление о модели зрелости процессов компании СММІ 1.2 и активно участвовать в работе групп по совершенствованию процессов
<p><b>Руководитель функционального подразделения</b></p>	<p>Большинство теоретических знаний менеджера.</p> <p>Книги: Кей Б., Джордан-Эванс Ш. «Любите их, или вы их потеряете. Как удержать ценных сотрудников»; Фиорина К. «Трудный выбор. Уроки бескомпромиссного лидерства в сложных ситуациях от экс-главы Hewlett-Packard».</p> <p>Рекомендую в сети профессиональных контактов (например, moikrug.ru, linked) заниматься просветительской деятельностью.</p> <p><i>Руководителю функционального подразделения в большей степени следует детально развивать профессиональные знания в области своей специализации, изучая специальную теоретическую литературу</i></p>	<p>Иметь большинство навыков менеджера проектов.</p> <p>Разрабатывать стратегию развития подразделения.</p> <p>Определять KGI и KPI.</p> <p>Выстраивать эффективное взаимодействие с другими подразделениями.</p> <p>Разрешать конфликты на всех уровнях.</p> <p>Проводить уроки по результатам выполненных работ/проектов.</p> <p>Профессионально развивать подчиненных.</p> <p>Курировать создание базы знаний подразделения.</p> <p>Управлять формализацией процессов и созданием системы менеджмента качества отдела.</p> <p>Разрабатывать процедуры, регламенты, должностные и рабочие инструкции.</p> <p><i>Руководитель функционального подразделения в меньшей степени ориентирован «вовне», на взаимодействие с заказчиком и клиентами, а в большей степени – на развитие своей команды. Он владелец качественного ресурсного пула. Его задачи – выделить требования к качеству ресурсов, прогнозировать потребность в количестве ресурсов разного качества, определить, как должны профессионально развиваться подчиненные, и управлять этим процессом. Разрешать производственные и процессные конфликты</i></p>
<p><b>Руководитель разработки ПО</b></p>	<p>Все теоретические знания руководителя функционального подразделения.</p> <p>Книги: Минцберг Г. «Структура в кулаке: создание эффективной организации»; Адизис И. «Идеальный руководитель».</p> <p>Рекомендую регулярно посещать сайты <a href="http://www.iteam.ru/">http://www.iteam.ru/</a>, <a href="http://msdn.microsoft.com/en-us/architecture/default.aspx">http://msdn.microsoft.com/en-us/architecture/default.aspx</a>,</p>	<p>Иметь все навыки руководителя функционального подразделения.</p> <p>Формировать портфель проектов и управлять им.</p> <p>Влиять на проектную среду во всех проектах подразделения и управлять ею в рамках подразделения.</p>

1	2	3
	<a href="http://www.cecsi.ru/">http://www.cecsi.ru/</a> , <a href="http://managementzone.ru">http://managementzone.ru</a> . Рекомендую в сети профессиональных контактов (например, moikrug.ru, linked) заниматься просветительской деятельностью. Рекомендую получить степень MBA или мини-MBA	Знать модель зрелости процессов компании СММ1 1.2, уметь анализировать эти области на предмет требуемых улучшений и несоответствий модели и оптимизировать процессы. Управлять работой всех процессных групп. Утверждать разработанную методологию разработки ПО и методологию управления проектами и продуктами для компании. Проводить тренинги и семинары. Уметь строить и развивать команду менеджеров

Дополнительно стоит упомянуть менеджера продукта — сотрудника, отвечающего за формирование стратегии развития продукта. Этот «вид» менеджеров послужил поводом для спора между авторами книги — включать или не включать их в таблицу. В результате было принято соломоново решение — не включать в таблицу, но хотя бы кратко описать отдельно профиль менеджера продукта, так как результаты его работы являются вводными для перечисленных разновидностей менеджеров. И если умолчать о данной роли, то получается, что видение создаваемых продуктов и планов их развития появляется у всех остальных управленцев по волшебству.

Для успешного выполнения своих обязанностей менеджер продукта должен обладать следующими навыками.

- ◆ Предугадывать будущие ожидания покупателя с помощью маркетинговых исследований.
- ◆ Если ожиданий нет, то формировать у покупателя «правильные» (нужные для успешного продвижения) ожидания к моменту выхода нового продукта.
- ◆ Формировать концепцию развития продукта, уметь взаимодействовать с бизнес-дивизионами и конечными потребителями.
- ◆ Понимать принципы построения архитектуры IT-уровня предприятия.
- ◆ Определять стратегию развития продукта.
- ◆ Уметь находить баланс между функциональным и архитектурным развитием продукта.
- ◆ Определять цели и требования к качеству релизов.
- ◆ Управлять качеством и рисками на уровне всего продукта.
- ◆ Управлять проектами внедрения продуктов.
- ◆ Помогать строить процессы поддержки и сопровождения продукта.

- ◆ Уметь управлять изменениями на уровне продукта.
- ◆ Иметь навыки управления портфелем проектов.

Крайне полезны для менеджера продукта также навыки и знания менеджера проекта.



---

Для обеспечения теоретической подготовки менеджера проекта я рекомендую получить все теоретические знания менеджера проекта, а также изучить книги: Данилин А., Слюсаренко А. «Архитектура и стратегия. “Инь” и “Янь” информационных технологий предприятия»; Ebert С. Understanding the Product Life Cycle: Four Key Requirements Engineering Techniques.

Рекомендую регулярно посещать сайт <http://msdn.microsoft.com/ru-ru/default.aspx>.

Рекомендую в сети профессиональных контактов (например, <http://moikrug.ru>, <http://linkedin.com>) заниматься просветительской деятельностью.

---

*Важно понимать, что менеджер продукта — не одна из ступенек развития классического управленца «администратор проекта → менеджер проекта → менеджер программы», а самостоятельное направление развития. Фактически менеджер продукта в чем-то является CEO, только не на уровне компании, а на уровне своей продуктовой линейки.*

В этом разделе мы рассмотрели возможную градацию профессиональных и специальных навыков менеджеров, было указано, какими паттернами, описываемыми в этой книге, менеджер должен уметь пользоваться на каждом этапе своего пути.



---

Еще раз напомним вам, что табл. 2.1 ни в коем случае не стоит принимать «в работу» буквально. С ней надо работать так же, как с квалификационной таблицей/профилем аналитика из первой книги, — это скорее ориентиры для поэтапного роста в профессии. И если сложится так, что, работая в роли администратора проекта, вы должны будете «выявить ожидания бизнеса к моменту выхода следующей версии продукта», то навык «формировать концепцию развития продукта, уметь взаимодействовать с бизнес-дивизионами и конечными потребителями» у вас появится гораздо раньше.

---

# 3. Что такое паттерн

---

---

В этой книге я называю паттерном воспроизводимый поведенческо-организационный шаблон, направленный на достижение определенного результата.



В каждом паттерне я постараюсь показать не только организационный аспект (как набор шагов, выполняемых в определенной последовательности, работающих с определенными артефактами и их состояниями, которые можно воспроизвести для достижения определенного результата), но и поведенческий аспект (как донести идею паттерна до команды, какую поведенческую линию выбрать и как построить коммуникации в команде).

Все паттерны разбиты на три группы: паттерны управления проектом, паттерны управления функциональным отделом, паттерны управления департаментом производства ПО. Однако это не значит, что паттерны разных групп не могут применяться перекрестно. Вы можете сформировать собственный список паттернов, расширить его своими паттернами, отказаться от каких-то паттернов полностью или значительно переделать их.

Паттерны особенно хороши для организаций с конфигурацией «профессиональная бюрократия» (см. раздел «Конфигурации организации» книги «Путь IT-профессионала»), к которой относится разработка ПО, так как работают в сфере стандартизации знаний и квалификации — одних из наиболее эффективных методов деятельности в организациях подобной конфигурации.

Для того чтобы вы максимально точно и однозначно поняли, что такое паттерн, приведу пример паттерна «Глажка мужской рубашки».



Сначала прогладьте воротник (воротник всегда гладьте изнутри), затем рукава рубашки, потом плечи и верхнюю часть, расположив рубашку на полукруглой части гладильной доски (лучше, если вы побрызгаете рубашку водой из пульверизатора). Затем, начиная с одной стороны рубашки и постепенно перемещая к себе непроглаженные участки на горизонтальной части гладильной доски, прогладьте рубашку целиком. Если спешите, тщательно проглаживайте только верхнюю часть спины и края рубашки с пуговицами и петлями.

Как вы понимаете, этот паттерн может быть далеко не единственным, есть и другие способы глажки рубашки, но лично я пользуюсь им вот уже не хочу считать сколько лет и убежден, что этот паттерн эффективен.

Если к этому паттерну добавить поведенческий аспект (что не просто даже при моей довольно богатой фантазии), ну, скажем: «Начиная гладить рубашку, скажите себе, что это простая задача и тут дел всего-то на две минуты, а перейдя к проглаживанию основной части рубашки, позвольте себе порадоваться, наблюдая, как исчезают складки и помятости, и получите удовольствие от работы», получится именно то, о чем мы будем говорить в этой книге.

Паттерн не является ни жесткой методологией, ни описанием ЖЦ, ни четким алгоритмом. Паттерн — это воспроизводимый поведенческо-организационный шаблон, направленный на достижение определенного результата.

В начале каждой главы с группой паттернов я буду приводить вот такую табличку (табл. 3.1), чтобы вам было проще ориентироваться.

**Таблица 3.1.** Описание паттерна — формат таблицы

Паттерн	Параметры среды	Кому нужен? Зачем нужен?	Организационный аспект	Поведенческий аспект
«Погладь рубашку»	Организация любого масштаба. Уровень формализации процессов — любой. Уровень квалификационных стандартов — любой	Любому, кто гладит мужские рубашки. Чтобы узнать еще один способ глажки	Порядок глажки. Применение гладильной доски в процессе глажки. Хорошие практики и рекомендации	Самонастрой. Включенность в действительность

**Сводная таблица паттернов.** Как было сказано ранее, паттерны сгруппированы по единому целевому признаку и отсортированы внутри группы в рекомендованном порядке их применения. Таким образом, начинающие управленцы смогут увидеть место каждого паттерна в, скажем так, возможной цепочке их применения, а также понять, каких знаний и опыта им не хватает для того, чтобы применять паттерны в «правильном» (то есть рекомендуемом авторами) порядке.

Но я сразу хочу оговориться и напомнить вам, что все эти цепочки паттернов — просто удобство и некая системность изложения. Каждый из паттернов вы можете применять сам по себе, не используя другие паттерны.

Чтобы помочь ориентироваться в материалах книги и паттернах в привязке к квалификации, я привожу далее сводную перекрестную таблицу всех паттернов, которые рассмотрены в этой книге, в зависимости от квалификации/профиля менеджера (табл. 3.2). Можете использовать этот раздел для быстрого поиска нужного паттерна.

Таблица 3.2. Сводная таблица паттернов с привязкой к ролям менеджеров

Паттерн	Кому нужен? Зачем нужен?	Администратор проекта	Релиз-менеджер	МП	Руководитель отдела	Менеджер программы проектов	Менеджер продукта	Руководитель разра-ботки ПО	Страница
1	2	3	4	5	6	7	8	9	10
«Базовые правила работы»	Команде. Чтобы установить единые прозрачные и четкие правила работы, помочь новому участнику команды максимально быстро влиться в среду проекта/компании (в качестве основной части пакета документов «Портфель нового сотрудника»), закрепить не-которые хорошие практики, формализованные в виде про-цедур и процессов								
«Выученные уроки»	Команде. Чтобы неформально описать нештатные ситуации, ошибки, на которых полезно поучиться всем сотрудникам компании, чтобы не наступать на одни и те же грабли. Чтобы сформировать список рекомендаций — по сути, квалификационных стандартов компании								
«Хорошие практики»	Управленцу и команде. Для формирования единых для всей организации полезных знаний и навыков,								

Продолжение ⇨



1	2	3	4	5	6	7	8	9	10
«План управления документами»	Команде. Для понимания скоупа (гранитиц проекта) и сроков объектов поставки проекта, раннего выявления и снятия рисков по качеству и срокам объектов поставки, использования в качестве основы для плана управления коммуникацией в проекте								
«Открытые вопросы»	Управленцу, архитектору. Чтобы быстро взглянуть в суть проекта, продукта и среды проекта, выявить и снять проблемы и неопределенности проекта, эффективно управлять рисками проекта								
«Система авторизации работ»	Управленцу, команде. Чтобы выполнять правильную работу в правильные сроки. Для контроля трудозатрат и точности оценок, контроля сроков и выявления рисков								
«Эффективный переносный разговорчик»	Любому участнику проектной команды. Чтобы договориться в формате «выигрыш-выигрыш» (win-win) со второй стороной								
«Итеративная разработка ПО»	Управленцу, команде. Чтобы минимизировать риски выявления новых требований на поздних стадиях разработки, минимизировать архитектурные риски, минимизировать								



1	2	3	4	5	6	7	8	9	10
	артфактов, а также получить готовый состав работ в зависимости от выбранного пакета спецификаций								
СМММ	Управленцу по развитию продуктов, управленцу по разработке ПО, команде. Чтобы определить рекомендации, конкретные практики и виды деятельности по процессу производства в целом и разработке ПО в частности								
IDEAL	Управленцу по развитию продуктов, управленцу по разработке ПО. Чтобы определить универсальный подход к стратегическому развитию продуктов								
«Система управления производством»	Управленцу по развитию продуктов, управленцу по разработке ПО, команде. Чтобы определить рекомендации, конкретные практики и виды деятельности по процессу производства в целом и разработке ПО в частности								
«Портрет проекта»	Управленцу. Чтобы проанализировать среду проекта, почувствовать уровень зрелости организации и на основании знаний различных методологий и реальности нарисовать свой портрет проекта								



1	2	3	4	5	6	7	8	9	10
«Управление неопределенностью проекта»	Управленцу, команде. Идея паттерна состоит в том, что по мере получения информации о среде проекта и самом проекте и в ходе управления ими вы постоянно снимаете неопределенности и смотрите, какие еще неопределенности нужно снять								
«СУП на РМВОК»	Управленцу. Чтобы построить архитектуру (процедуры, их взаимодействия, действия и проектные артефакты)								
«СУП на воде»	Управленцу, команде. Чтобы разработать универсальные и наиболее критические принципы управления проектами и проектными коммуникациями, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев								
«Зажарка в СУП, или Универсальные принципы управления проектом»	Управленцу, команде. Чтобы разработать универсальные принципы управления проектами, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев								

Продолжение ⇨

Таблица 3.2 (продолжение)

1	2	3	4	5	6	7	8	9	10
«Стабилизация ПО»	<p>Управленцу, команде. В ситуации, когда вы не можете позволить себе полноценное управление проектом, можно применить паттерн «Экстремальная разработка» совместно с данным паттерном для более или менее эффективного управления проектом «на колени».</p> <p>Для своевременного обнаружения и мониторинга картины стабилизации ПО с помощью этого паттерна, что позволит своевременно перенаправить ресурсы разработчика на решение проблем, которые были наиболее критическими с точки зрения начала внедрения продукта.</p> <p>Этот же паттерн используется для регрессионного тестирования. Тестируются критические баги или которые устранялись крайне долго и чаще, чем все прочие</p>								
«С ног на голову, или ПИП ПСР (программа и методика проведения приемосдаточных работ)»	<p>Управленцу со стороны заказчика.</p> <p>Может применяться как фактор, «вытягивающий» представление о системе в случае, когда проект в критическом пике, а подрядчик не настроен на конструктив и не хочет ничего вам рассказывать</p>								

1	2	3	4	5	6	7	8	9	10
	и создавать дополнительные спецификации или диаграммы. Паттерн может применяться не только в ситуации «с ног на голову», но и при нормальной, прямой и последовательной разработке								
«Фундамент отдела»	Руководителю функционального отдела. Относится к паттернам формирования среды. От того, насколько хорошо применен и применен ли вообще этот паттерн в отделе, зависит общая корпоративная культура и то, насколько в принципе жизнеспособны любые производственные процессы организации								
«Удаленное управление»	Управленцу, руководителю функционального отдела. Для управления распределенной или удаленной командой								
«Маслоу по Фрейдцу»	Управленцу, руководителю функционального отдела. Для того чтобы сформировать у сотрудников нужную модель поведения, точно довести до них существующие правила, научить их работать в коллективе								
«Лидерство»	Руководителю функционального отдела. Сборник некоторых хороших практик с описанием								





# 4. Паттерны формирования проектной среды

---

---

## 4.1. Немного о среде проекта

Проект всегда существует в какой-то среде. Развивая блестящую метафору Дуга Де Карло [10] о том, что проект в широком смысле слова — сад, немного пофантазируем. В саду есть земля, глинистая или чернозем, богатая или бедная минералами, само растение (проект), рядом растущие растения (другие проекты), садовники, жуки и растения-вредители, яды и удобрения. Все, что вокруг нашего «растения-проекта», и есть среда проекта. Это организационная структура, процессы и инструменты как на стороне исполнителя, так и на стороне заказчика, архитектура IT (сегмент рынка, целевая аудитория), в которую должен вписаться новый продукт, уровень зрелости этих организаций, механизмы и принципы принятия решений и коммуникаций, бизнес- и личные (да-да, личные) цели заинтересованных лиц, квалификация всех этих лиц и в первую очередь проектной команды.

Прежде чем мы перейдем к дальнейшему изложению, я хочу поделиться с вами довольно старой историей, которая гуляет по Интернету. К сожалению, привожу текст без ссылки на первоисточник, потому что просто не знаю, кто автор этого замечательного рассказа.



---

Петров пришел во вторник на совещание.

Ему там вынули мозг, разложили по блюдецкам и стали есть, причмокивая и вообще выражая всяческое одобрение. Начальник Петрова, Недозайцев, предусмотрительно раздал присутствующим десертные ложечки. И началось.

— Коллеги, — говорит Морковьева, — перед нашей организацией встала масштабная задача. Нам поступил на реализацию проект, в рамках которого нам требуется изобразить несколько красных линий. Вы готовы взвалить на себя эту задачу?

— Конечно, — говорит Недозайцев. Он директор и всегда готов взвалить на себя проблему, которую придется нести кому-то из коллектива. Впрочем, он тут же уточняет:

— Мы же это можем?

Начальник отдела рисования Сидоряхин торопливо кивает:

— Да, разумеется. Вот у нас как раз сидит Петров, он наш лучший специалист в области рисования красных линий. Мы его специально пригласили на совещание, чтобы он высказал свое компетентное мнение.

— Очень приятно, — говорит Морковьева. — Ну, меня вы все знаете. А это — Леночка, она специалист по дизайну в нашей организации.

Леночка покрывается краской и смущенно улыбается. Она недавно окончила экономический и к дизайну имеет такое же отношение, как утконос к проектированию дирижаблей.

— Так вот, — говорит Морковьева. — Нам нужно нарисовать семь красных линий. Все они должны быть строго перпендикулярны, и, кроме того, некоторые нужно нарисовать зеленым цветом, а еще некоторые — прозрачным. Как вы считаете, это реально?

— Нет, — говорит Петров.

— Давайте не будем торопиться с ответом, Петров, — говорит Сидоряхин. — Задача поставлена, и ее нужно решить. Вы же профессионал, Петров. Не давайте нам повода считать, что вы не профессионал.

— Видите ли, — объясняет Петров, — термин «красная линия» подразумевает, что цвет линии — красный. Нарисовать красную линию зеленым цветом не то чтобы невозможно, но очень близко к невозможному...

— Петров, ну что значит «невозможно»? — спрашивает Сидоряхин.

— Я просто обрисовываю ситуацию. Возможно, есть люди, страдающие дальтонизмом, для которых действительно не будет иметь значения цвет линии, но я не уверен, что целевая аудитория вашего проекта состоит исключительно из таких людей.

— То есть, в принципе, это возможно, мы правильно вас понимаем, Петров? — спрашивает Морковьева.

Петров осознает, что переборщил с образностью.

— Скажем проще, — говорит он. — Линию как таковую можно нарисовать совершенно любым цветом. Но чтобы получилась красная линия, следует использовать только красный цвет.

— Петров, вы нас не путайте, пожалуйста. Только что вы говорили, что это возможно.

Петров молча проклиная свою болтливость.

— Нет, вы неправильно меня поняли. Я хотел лишь сказать, что в некоторых крайне редких ситуациях цвет линии не будет иметь значения, но даже и тогда линия все равно не будет красной. Понимаете, она красной не будет! Она будет зеленой. А вам нужна красная.

Наступает непродолжительное молчание, в котором отчетливо слышится тихое напряженное гудение синапсов.

— А что, если, — осененный идеей, произносит Недозайцев, — нарисовать их синим цветом?

— Все равно не получится, — качает головой Петров. — Если нарисовать синим, получатся синие линии.

Опять молчание. На этот раз его прерывает сам Петров.

— И я еще не понял... Что вы имели в виду, когда говорили о линиях прозрачного цвета?

Морковьева смотрит на него снисходительно, как добрая учительница на отстающего ученика.

— Ну как вам объяснить?.. Петров, вы разве не знаете, что такое «прозрачный»?

— Знаю.

— И что такое «красная линия», надеюсь, вам тоже не надо объяснять?

— Нет, не надо.

— Ну вот. Вы нарисуйте нам красные линии прозрачным цветом.

Петров на секунду замирает, обдумывая ситуацию.

— И как должен выглядеть результат? Будьте добры, опишите, пожалуйста. Как вы себе это представляете?

— Ну-у-у, Петро-о-ов! — говорит Сидоряхин. — Ну давайте не будем... У нас что, детский сад? Кто здесь специалист по красным линиям, Морковьева или вы?

— Я просто пытаюсь прояснить для себя детали задания...

— Ну а что тут непонятного-то?.. — встречается в разговор Недозайцев. — Вы же знаете, что такое красная линия?

— Да, но...

— И что такое «прозрачный», вам тоже ясно?

— Разумеется, но...

— Так что вам объяснять-то? Петров, ну давайте не будем опускаться до непродуктивных споров. Задача поставлена, задача ясная и четкая. Если у вас есть конкретные вопросы, так задавайте.

— Вы же профессионал, — добавляет Сидоряхин.

— Ладно, — сдается Петров. — Бог с ним, с цветом. Но у вас там еще что-то с перпендикулярностью?..

— Да, — с готовностью подтверждает Морковьева. — Семь линий, все строго перпендикулярны.

— Перпендикулярны чему? — уточняет Петров.

Морковьева начинает просматривать свои бумаги.

— Э-э-э, — говорит она наконец. — Ну как бы... Всему. Между собой. Ну или как там... Я не знаю. Я думала, это вы знаете, какие бывают перпендикулярные линии, — наконец находится она.

— Да конечно, знает, — взмахивает руками Сидоряхин. — Профессионалы мы тут или не профессионалы?..

— Перпендикулярны могут быть две линии, — терпеливо объясняет Петров. — Все семь одновременно не могут быть перпендикулярными по отношению друг к другу. Это геометрия, 6-й класс.

Морковьева встряхивает головой, отгоняя замаячивший призрак давно забытого школьного образования. Недозайцев хлопает ладонью по столу:

— Петров, давайте без вот этого: «6-й класс, 6-й класс». Давайте будем взаимно вежливы. Не будем делать намеков и скатываться до оскорблений. Давайте поддерживать конструктивный диалог. Здесь же не идиоты собрались.

— Я тоже так считаю, — говорит Сидоряхин.

Петров придвигает к себе листок бумаги.

— Хорошо, — говорит он. — Давайте, я вам нарисую. Вот линия. Так?

Морковьева утвердительно кивает головой.

— Рисуем другую... — говорит Петров. — Она перпендикулярна первой?

— Ну-у...

— Да, она перпендикулярна.

— Ну вот видите! — радостно восклицает Морковьева.

— Подождите, это еще не все. Теперь рисуем третью... Она перпендикулярна первой линии?..

Вдумчивое молчание. Не дождавшись ответа, Петров отвечает сам:

— Да, первой линии она перпендикулярна. Но со второй линией она не пересекается. Со второй линией они параллельны.

Наступает тишина. Потом Морковьева встает со своего места и, обогнув стол, заходит Петрову с тыла, заглядывая ему через плечо.

— Ну... — неуверенно произносит она. — Наверное, да.

— Вот в этом и дело, — говорит Петров, стремясь закрепить достигнутый успех. — Пока линий две, они могут быть перпендикулярны. Как только их становится больше...

— А можно мне ручку? — просит Морковьева.

Петров отдает ручку. Морковьева осторожно проводит несколько неуверенных линий.

— А если так?..

Петров вздыхает.

— Это называется треугольник. Нет, это не перпендикулярные линии. К тому же их три, а не семь.

Морковьева поджимает губы.

— А почему они синие? — вдруг спрашивает Недозайцев.

— Да, кстати, — поддерживает Сидоряхин. — Сам хотел спросить.

Петров несколько раз моргает, разглядывая рисунок.

— У меня ручка синяя, — наконец говорит он. — Я же просто, чтобы продемонстрировать...

— Ну так, может, в этом и дело? — нетерпеливо перебивает его Недозайцев тоном человека, который только что разобрался в сложной концепции и спешит поделиться ею с окружающими, пока мысль не потеряна. — У вас линии синие. Вы нарисуйте красные, и давайте посмотрим, что получится.

— Получится то же самое, — уверенно говорит Петров.

— Ну как то же самое? — говорит Недозайцев. — Как вы можете быть уверены, если вы даже не попробовали? Вы нарисуйте красные, и посмотрим.

— У меня нет красной ручки с собой, — признается Петров. — Но я могу совершенно...

— А что же вы не подготовились, — укоризненно говорит Сидоряхин. — Знали же, что будет собрание...

— Я абсолютно точно могу вам сказать, — в отчаянии говорит Петров, — что красным цветом получится точно то же самое.

— Вы же сами нам в прошлый раз говорили, — парирует Сидоряхин, — что рисовать красные линии нужно красным цветом. Вот, я записал себе даже. А сами рисуете их синей ручкой. Это что, красные линии по-вашему?

— Кстати, да, — замечает Недозайцев. — Я же еще спрашивал вас про синий цвет. Что вы мне ответили?

Петрова внезапно спасает Леночка, с интересом изучающая его рисунок со своего места.

— Мне кажется, я понимаю, — говорит она. — Вы же сейчас не о цвете говорите, да? Это у вас про вот эту, как вы ее называете? Перпер-чего-то-там?

— Перпендикулярность линий, да, — благодарно отзывается Петров. — Она с цветом линий никак не связана.

— Все, вы меня запутали окончательно, — говорит Недозайцев, переводя взгляд с одного участника собрания на другого. — Так у нас с чем проблемы? С цветом или с перпендикулярностью?

Морковьева издает растерянные звуки и качает головой. Она тоже запуталась.

— И с тем и с другим, — тихо говорит Петров.

— Я ничего не могу понять, — говорит Недозайцев, разглядывая свои сцепленные в замок пальцы. — Вот есть задача. Нужно всего-то семь красных линий. Я понимаю, их было бы двадцать!.. Но тут-то всего семь. Задача простая. Наши заказчики хотят семь перпендикулярных линий. Верно?

Морковьева кивает.

— И Сидоряхин вот тоже не видит проблемы, — говорит Недозайцев. — Я прав, Сидоряхин?.. Ну вот. Так что нам мешает выполнить задачу?

— Геометрия, — со вздохом говорит Петров.

— Ну вы просто не обращайтесь на нее внимания, вот и все! — произносит Морковьева.

Петров молчит, собираясь с мыслями. В его мозгу рождаются одна за другой красочные метафоры, которые позволили бы донести до окружающих сюрреализм происходящего, но, как назло, все они, облекаясь в слова, начинаются неизменно жаргонным словом, совершенно неуместным в рамках деловой беседы.

Устав ждать ответа, Недозайцев произносит:

— Петров, вы ответьте просто — вы можете сделать или вы не можете? Я понимаю, что вы узкий специалист и не видите общей картины. Но это же несложно — нарисовать какие-то семь линий? Обсуждаем уже два часа какую-то ерунду, никак не можем прийти к решению.

— Да, — говорит Сидоряхин, — вы вот только критикуете и говорите: «Невозможно! Невозможно!» Вы предложите нам свое решение проблемы! А то критиковать и дурак может, простите за выражение. Вы же профессионал!

Петров устало изрекает:

— Хорошо. Давайте я нарисую вам две гарантированно перпендикулярные красные линии, а остальные — прозрачным цветом. Они будут прозрачны, и их не будет видно, но я их нарисую. Вас это устроит?

— Нас это устроит? — обращившись Морковьева к Леночке. — Да, нас устроит.

— Только еще хотя бы пару — зеленым цветом, — добавляет Леночка. — И еще у меня такой вопрос, можно?

— Да, — мертвым голосом разрешает Петров.

— Можно одну линию изобразить в виде котенка?

Петров молчит несколько секунд, а потом переспрашивает:

— Что?

— Ну в виде котенка. Котеночка. Нашим пользователям нравятся зверюшки. Было бы очень здорово...

— Нет, — говорит Петров.

— А почему?

— Нет, я, конечно, могу нарисовать вам кота. Я не художник, но могу попытаться. Только это будет уже не линия. Это будет кот. Линия и кот — разные вещи.

— Котенок, — уточняет Морковьева. — Не кот, а котенок, такой маленький, симпатичный. Коты, они...

— Да все равно, — качает головой Петров.

— Совсем никак, да?.. — разочарованно спрашивает Леночка.

— Петров, вы хоть дослушали бы до конца, — раздраженно говорит Недозайцев. — Не дослушали, а уже говорите «нет».

— Я понял мысль, — не поднимая взгляда от стола, говорит Петров. — Нарисовать линию в виде котенка невозможно.

— Ну и не надо тогда, — разрешает Леночка. — А птичку тоже не получится? Петров молча поднимает на нее взгляд, и Леночка все понимает.

— Ну и не надо тогда, — снова повторяет она.

Недозайцев хлопает ладонью по столу.

— Так на чем мы остановились? Что мы делаем?

— Семь красных линий, — говорит Морковьева. — Две красным цветом, и две зеленым, и остальные прозрачным. Да? Я же правильно поняла?

— Да, — подтверждает Сидоряхин прежде, чем Петров успевает открыть рот.

Недозайцев удовлетворенно кивает.

— Вот и отлично... Ну тогда все, коллеги?.. Расходимся?.. Еще вопросы есть?..

— Ой, — вспоминает Леночка. — У нас еще есть красный воздушный шарик! Скажите, вы можете его надуть?

— Да, кстати, — говорит Морковьева. — Давайте это тоже сразу обсудим, чтобы два раза не собираться.

— Петров, — поворачивается Недозайцев к Петрову. — Мы это можем?

— А какое отношение ко мне имеет шарик? — удивленно спрашивает Петров.

— Он красный, — поясняет Леночка.

Петров тупо молчит, подрагивая кончиками пальцев.

— Петров, — нервно переспрашивает Недозайцев. — Так вы это можете или не можете? Простой же вопрос.

— Ну, — осторожно говорит Петров, — в принципе, я, конечно, могу, но...

— Хорошо, — кивает Недозайцев. — Съездите к ним, надуйте. Командировочные, если потребуется, выпишем.

— Завтра можно? — спрашивает Морковьева.

— Конечно, — отвечает Недозайцев. — Я думаю, проблем не будет... Ну теперь у нас все?.. Отлично. Продуктивно поработали... Всем спасибо и до свидания!

Петров несколько раз моргает, чтобы вернуться в объективную реальность, потом встает и медленно бредет к выходу. У самого выхода Леночка догоняет его.

— А можно еще вас попросить? — краснея, говорит Леночка. — Вы когда шарик будете надуть... Вы можете надуть его в форме котенка?..

Петров вздыхает.

— Я все могу, — говорит он. — Я могу абсолютно все. Я профессионал.

(с)

Вот такая замечательная история. Это хороший пример, иллюстрирующий в юмористической форме влияние проектной среды на проект, корпоративную культуру и не только. Уверен, что, кем бы вы ни работали, вы найдете свои аналогии и свой подтекст.

Вырастить растение в таком саду, на такой почве можно только в случае, если это растение неприхотливо, — есть ведь растения, растущие даже на скалах (хотя и скалах есть немного земли и воды). Или это растение по факту никому не интересно — не интересно ни что в итоге вырастет, ни сколько оно просуществует.

Мне искренне жаль Петрова как профессионала, но еще больше мне жаль акционеров и/или учредителей этой компании. Увеличить капитализацию компании при таких организационных активах невозможно, если заниматься легальным бизнесом. Что-то подсказывает мне, что, скорее всего, это или государственная структура, или компания, работающая на щедро финансируемых контрактах или грантах.

Я хочу обратить ваше внимание на то, что в подобной проектной среде крайне сложно применять паттерны, описанные в этой книге, но тем не менее это возможно. Паттерны по формированию проектной среды смогут облегчить жизнь персонально Петрову, и в идеале в зарослях этого сада сюрреализма он сможет организовать небольшой газончик объективной реальности для своего проекта. Однако для последовательных и системных преобразований организации нужна явная поддержка и воля со стороны высшего руководства и владельцев компании.

Искренне желаю, чтобы ваша проектная среда была более дружелюбна к вам и вашим проектам.

## 4.2. Сводная таблица паттернов формирования проектной среды

Далее приводится сводная таблица всех паттернов формирования среды, которые есть в данной главе (табл. 4.1). Вы можете использовать этот раздел для быстрого поиска нужного паттерна, чтобы начать его использовать прямо сейчас.

### 4.2.1. Общие паттерны

Общие паттерны — это паттерны, которые гарантированно могут применяться независимо от специфики IT и разработки ПО. Это не означает, что паттерны, структурированные в разделе «Паттерны IT», не могут применяться с той же целью, — могут, но их придется адаптировать.

Таблица 4. 1. Паттерны формирования проектной среды

Паттерн	Кому нужен? Зачем нужен?	Организационный аспект	Поведенческий аспект	Профиль менеджера	Страница
1 «Базовые правила работы»	2 Команде. Чтобы установить единые прозрачные и четкие правила работы, помочь новому участнику команды максимально быстро влиться в среду проекта/компании (в качестве основной части пакета документов «Портфель нового сотрудника»), закрепить некоторые хорошие практики, формализованные в виде процедур и процессов	3 Создать неформальный документ «Базовые правила», если не существует регламента работы, в котором эти же моменты отражены формально. Осветить в документе роли, ответственность, коммуникации, виды деятельности и результатов работ, установленные в компании процессы и процедуры, техническую инфраструктуру, обучение и профессиональный рост. Постараться разъяснить, кто, что, когда/по наступлении какого события, в какой форме, кому представляет по основным моментам деятельности	4 Разъяснение целей создания документа. Презентации/разъяснения в части процедур и коммуникационных правил. Документ должен быть понятен и безусловно принят всеми членами команды. Открытость паттернов «Выученные уроки» и «Хорошие практики» по отношению к данному паттерну	5 Менеджер проекта, менеджер программ проектов, руководитель разработки ПО	6
«Выученные уроки»	Команде. Чтобы неформально описать нештатные ситуации, ошибки, на которых полезно поучиться всем сотрудникам компании, чтобы не наступать на одни и те же грабли. Чтобы сформировать список рекомендаций – по сути, квалификационных стандартов компании	Создать реестр выученных уроков (ВУ). Описываемые ситуации должны быть максимально обезличены и абстрагированы. На периодической/событийной основе анализировать реестр, обсуждать его с сотрудниками. Рекомендовать отдельные ВУ как кандидаты в хорошие практики (ХП). Информировать команду о новых ВУ	Команда должна понимать, что инструмент ВУ — это не карательный, а обучающий механизм. Любой участник команды может внести запись в реестр ВУ в любое время. ВУ носит рекомендательный характер — применяется на усмотрение управленца. Любой участник команды может поставить под сомнение любой имеющийся ВУ	Администратор проекта, релиз-менеджер, менеджер проекта, менеджер программы проектов	

1	2	3	4	5	6
«Своевременное и полное информирование: статус проекта»	<p>Управленду и команде. Для формирования единых для всей организации полезных знаний и навыков, которые заработаны потом и кровью на ранее выполненных проектах/ работах, неформальной и гибкой организации процессов</p>	<p>На базе имеющихся ВУ провести анализ рекомендованных ВУ как кандидатов в ХП. Рекомендация ХП, как правило, более обобщенная, чем рекомендация ВУ, для уровня всей организации. Дать обратную связь по новым ХП инициатору. Информировать команду о новых ХП. Реестр ХП должен пересматриваться на регулярной основе с целью дальнейшей формализации в виде процессов, процедур или рабочих инструкций</p>	<p>Команда должна понимать, что ХП – это процессы, которые не высосаны из пальца, а реально существуют в компании. Команда должна серьезно относиться к ХП, так как «инструкции написаны кровью». ХП носит строго рекомендательный характер, то есть может не применяться только при согласовании с компетентным органом (например, с офисом управления проектом). Любой участник команды может поставить под сомнение любую имеющуюся ХП</p>	<p>Менеджер проектов, граммамы проектов, руководитель разработки ПО</p>	
	<p>Заинтересованным лицам, команде. Для выработки единой оценки происходящего в проекте на базе своевременно поступающей информации, раннего выявления рисков и проблем</p>	<p>Еженедельно готовить и рассылать в виде информационного письма актуальную информацию: обновленную базовую версию плана расписания, скоуп работ на неделю и возникающие в связи с этим задачи и поручения, проблемы и риски, статус артефактов по ПУД (см. паттерн ПУД)</p>	<p>Заинтересованные лица проекта и команда проекта должны понимать данное информационное письмо не как формальный отчет, а как общее видение ситуации в проекте. Любое заинтересованное лицо и любой участник команды может задать уточняющие вопросы/ дать комментарии/ внести поправки в информацию о статусе проекта в любой его части.</p>	<p>Администратор проекта, релиз-менеджер, менеджер проекта</p>	

Таблица 4.1 (продолжение)

1	2	3	4	5	6
«Своевременное и полное информирование: информационный бюллетень»	Управленцу, топ-менеджменту, команде. Для того чтобы выстроить конструктивный диалог в проектах, затрагивающих интересы многих сотрудников (например, совершенствование процессов), увлечь общими целями и идеями. Это командообразующий паттерн, он позволяет повысить вовлеченность сотрудников в процесс/проект	Подготовить и разослать всем сотрудникам компании (копию — топ-менеджерам компании) информационное письмо, выдержанное в нейтрально-доброжелательном ободрающем тоне. Критичные моменты описывать максимально беспристрастно и обезличенно, положительные, напротив, всегда подкреплять примером работы конкретных сотрудников с указанием фамилии и вклада, который они внесли в общее дело	План-расписание и ПУД должны править обе стороны проекта — исполнитель и заказчик  Сотрудники должны видеть, что одна и та же информация доводится до их сведения и до сведения руководства. Любое заинтересованное лицо и любой участник команды может внести поправки в разосланную информацию. Важно, чтобы все понимали: ответить на это письмо так, чтобы видели все, можно только в случае, если это действительно важная правка/дополнение, в противном случае ответ с уточнениями/вопросами посылается только автору письма. Важно также, чтобы все понимали, что возможна только одна итерация уточнений	Менеджер проекта, менеджер программы проектов, руководитель разработки ПО	
«Управление средой проекта»	Управленцу, спонсору проекта, заинтересованным лицам проекта. Для достижения успеха проекта путем	Управление средой сводится к управлению ожиданиями и отношениями участников проекта. Проект — это сад, экосистема, состоящая из цветов, почвы и погодных	Четыре ускорителя — принципы, которые дают свободу мотивации и инновациям. Необходимо обеспечение взаимного доверия для	Менеджер проекта, менеджер программы проектов	

1	2	3	4	5	6
«План управления документами»	Команде. Для понимания скоупа и сроков объектов поставки проекта, раннего выявления и снятия рисков по качеству и срокам объектов поставки, использования в качестве основы для плана управления коммуникацией в проекте	условий. Цветы являются содержанием, а почва и погодные условия представляют собой среду. Без благоприятной среды цветы погибнут. Должностная инструкция менеджера экстремального проекта. Четыре вопроса бизнеса. Механизм достижения согласованных позиций	достижения успеха путем внедрения 10 общих ценностей	Администратор проекта, релиз-менеджер, менеджер проекта, менеджер программы проектов	
«Открытые вопросы»	Управленцу, архитектору. Чтобы быстро вникнуть в суть проекта, продукта и среды проекта, выявить и снять проблемы	Выявляются все нужные для проекта объекты поставки. Объект поставки должен иметь конкретный «заказчик», которому он нужен. Объекты поставки привязываются к вехам проекта (фазам, этапам). Назначаются ответственные, согласующие и утверждающие объект поставки лица	Вся команда должна понимать цель и необходимость создания данного объекта поставки именно в это время. ПУД является и стратегией, и тактикой, и оперативным планом по выпуску объектов поставки. ПУД является «скелетом» коммуникаций в проекте — даже если роли и ответственности четко не прописаны, ПУД косвенным образом определяет зоны ответственности	Менеджер проекта, менеджер программ проектов	
		Паттерн «Открытые вопросы» дополняет паттерн ПУД и реестр рисков. При создании ПУД по каждому объекту поставки формируется перечень открытых вопросов и фиксируется в документе «Открытые вопросы».	За каждый открытый вопрос назначается один ответственный, который отвечает за получение качественной информации, достаточной для присвоения вопросу статуса «пояснен».		

Таблица 4.1 (продолжение)

1	2	3	4	5	6
«Система авторизации работ (WAS)»	и неопределенности проекта, эффективно управлять рисками проекта	По мере работы с открытыми вопросами анализируются возможные риски и их последствия. Всеми открытыми вопросами управляют с помощью статусов «открыт», «пояснен», «закрыт»	Этот же ответственный должен выявлять риски, связанные с открытым вопросом. Документ доступен, и с ним работает вся проектная команда, включая представителей заказчика, — все знают о существующих проблемах и открытых вопросах и могут добавлять выявленные или открытые вопросы	Администратор проекта, релиз-менеджер, менеджер проекта, менеджер группы, руководитель программы проектов, руководитель разработки ПО	
	Управленцу, команде. Чтобы выполнять правильную работу в правильные сроки. Для контроля трудозатрат и точности оценок, контроля сроков и выявления рисков	После формирования ПУД и состава выполняемых работ (WBS) проекта, для того чтобы получить объекты поставки, работы из WBS авторизуются уполномоченным лицом для принятия решения об их старте. Только после этого работы начинают выполняться. Работы имеют статусы и другие управленческие атрибуты. WAS может брать на себя функции workflow. Одна задача — один ответственный	Персональная ответственность и понимание контекста задачи и ее связи с целями проекта/ рабочей группы/ подразделения. Каждый участник команды понимает, что WAS — это инструмент эффективного планирования собственных работ. Каждый участник максимально информативно заполняет атрибуты задач. Если в WAS не предусмотрено на система автоматических уведомлений (например, реализованная в Excel), то ответственный сотрудник рассылает уведомления вручную		

## 4.2.2. Сводная таблица паттернов разработки ПО

Эти паттерны освещают более сложные вопросы, которые выходят за рамки общих паттернов формирования проектной среды, но тем не менее являются именно паттернами этой серии, так как позволяют сформировать единое поле выполнения проекта в команде (табл. 4.2).

## 4.3. Паттерн «Базовые правила работы»

Одна из первоочередных задач при формировании благоприятной среды для выполнения проекта — установить единые и прозрачные зоны ответственности участников команды и заинтересованных лиц, зоны принятия решений в соответствии с проектными ролями, а также принципы и правила коммуникации.



---

Если вы вспомните ситуацию с бедным Петровым и семью перпендикулярными красными линиями, нарисованными прозрачным цветом, то этот паттерн (при условии, что он работает во всей компании) сведет вероятность проведения подобного неэффективного совещания практически к нулю.

---

Далее приведена возможная структура документа «Базовые правила работы».

1. Роли и ответственность.
2. Коммуникации.
  - 2.1. Форма общения.
  - 2.2. Правила разрешения конфликтов и эскалации.
  - 2.3. Правила проведения дискуссий и принятия решений.
  - 2.4. Правила переписки.
  - 2.5. Правила проведения собраний.
  - 2.6. Правила подготовки и проведения презентаций.
3. Виды деятельности и результатов работ.
  - 3.1. Работы, выполняемые в рамках операционной деятельности.
  - 3.2. Работы, выполняемые в рамках проектной деятельности.
  - 3.3. ...
  - 3.4. Документальные отчеты.
  - 3.5. Спецификации, модели, диаграммы, планы.

Таблица 4.2. Паттерны разработки ПО

Паттерн	Кому нужен? Зачем нужен?	Организационный аспект	Поведенческий аспект	Профиль менеджера	Страница
1 «Итеративная разработка ПО»	2 Управленцу, команде. Чтобы минимизировать риски выявления новых требований на поздних стадиях разработки, минимизировать архитектурные риски, минимизировать риски по «плавающим» требованиям. Для вовлечения заказчика в процесс разработки продуктов на ранних стадиях проекта	3 Итерации строятся по мере узнавания предметной области. Каждая итерация завершается созданием прототипа. Заказчик исследует прототип на предмет анализа собственных ожиданий от системы, правильности общего направления разработки продукта, корректности интерфейсных взаимодействий и удобства использования будущей системы. Выявляется основной поток всех критичных блоков информации (use cases). Следует детализировать описание основных потоков первых 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, и сформировать список пожеланий к use case с точки зрения каждой роли. Нужно детализировать описание основного и альтернативных потоков выполнения всех критических use cases, детализировать описание основного пути для следующих 20 % из списка use cases, создать описание альтернативных потоков выполнения 40 % use cases. Создать первую версию пакета спецификаций требований (MSRS). Заказчик должен уточнить и утвердить полный пакет спецификаций требований.	4 Не теряем темпа. Не углубляемся в детали преждевременно. Двигаемся по приоритетам с шагом охвата предметной области 20 %. Всегда сначала описываем основные потоки, потом постепенно рассматриваем альтернативы. Как можно раньше и как можно чаще представляем заказчику прототип. Работа командная. Все члены команды участвуют в совещаниях и обсуждениях функциональности и прототипа	5 Менеджер проекта, менеджер программы проектов	6

1	2	3	4	5	6
«Релизное управление»	<p>Управленцу по развитию продуктов, управленцу по разработке ПО, команде.</p> <p>Для выработки единого видения и понимания управления поставками версий ПО заказчику, взвешенной оценки того, так ли это срочно, если так дорого</p>	<p>Сформировать три кита развития продукта (стратегию развития продукта, функциональную концепцию, архитектурную концепцию).</p> <p>Обозначить границы релизов. Оценить фичи.</p> <p>Обозначить новые границы релизов. Постоянно фокусироваться на трех китах</p>	<p>Все понимают, что такое точность порядка величин и три кита развития продукта.</p> <p>Основное достижение и необходимое условие для работы этого паттерна — командная работа. С самого начала до самого конца задействованы все роли ЖЦ производства компании</p>	<p>Менеджер программы проектов, менеджер продукта, руководитель ПО</p>	
«Обеспечение качества <sup>1</sup> и анализ рисков»	<p>Управленцу по развитию продуктов, управленцу по разработке ПО, команде.</p> <p>Чтобы определить требования к качеству продукта и всесторонне проанализировать риски их достижения, работы, которые целесообразно выполнить для производства продукта определенного качества. Для того чтобы вовлечь заказчика в процесс разработки продукта на ранних стадиях проекта</p>	<p>Выдвинуть требования к качеству в соответствии с ГОСТ 28195–89 «Оценка качества программных средств».</p> <p>В зависимости от выдвинутых требований к качеству и характеристик разрабатываемой системы определить методологию разработки ПО.</p> <p>Определить WBS. Он направлен на достижение качества или снятие рисков.</p> <p>Анализ WBS на наличие «голд платины» — работы по созданию никому не нужных «дорогих» функций</p>	<p>Менеджер продукта должен выдвигать требования к качеству продукта, менеджер проекта совместно с аналитиком проецирует список рисков, и формирует список рисков, мешающих достичь требуемого качества продукта.</p> <p>Менеджер продукта должен ранжировать риски как критические, приоритетные и некритические, то есть фактически какие-то риски принять, а какие-то отметить как те, что крайне необходимо устранить.</p> <p>Менеджер проекта должен управлять рисками — снижать вероятность их возникновения путем включения соответствующих работ в WBS проекта</p>	<p>Менеджер проекта, менеджер программы проектов, менеджер продукта</p>	

Продолжение ⇨

<sup>1</sup> Термин «качество» употребляется здесь в том смысле, в котором он используется в ISO: качество — совокупность свойств и характеристик продукции или услуги, которые придают им способность удовлетворять обусловленные или предполагаемые потребности.

Таблица 4.2 (продолжение)

1	2	3	4	5	6
«Пакеты спецификаций требований»	<p>Управленцу по развитию продуктов, управленцу по разработке ПО, команде.</p> <p>Чтобы определить методологию разработки ПО посредством выбора того или иного пакета спецификаций требований и управленческих артефактов, а также получить готовый состав работ в зависимости от выбираемого пакета спецификаций</p>	<p>При старте проекта определить максимально подходящий для целей и масштаба проекта пакет спецификаций требований и управленческих артефактов.</p> <p>Багфикс или hotfix — Extreme.</p> <p>Проект доработки нескольких несложных фич — Simple.</p> <p>Проект по выпуску нового релиза существующей системы — Basic.</p> <p>Проект по созданию новой системы — Advanced или Gold (масштабная система).</p> <p>Проект по созданию системы, в котором критичны бизнес- и пользовательские требования, — VIP</p>	<p>Проектная команда должна понимать суть и назначение каждого артефакта из выбранного пакета спецификаций.</p> <p>Если в ходе выполнения проекта становится понятно, что в пакете спецификаций чего-то не хватает, пакет расширяется.</p> <p>Если понятно, что что-то лишнее, пакет сужается.</p> <p>Нет заказчика — нет артефакта. Все артефакты должны быть востребованы кем-то в проектной команде. Если нет такого заказчика, артефакт или раздел артефакта не создается.</p> <p>Нет «голд-платине»!</p> <p>Гибкое управление информацией. Фиксируем только то, что нужно для производства продукта и выполнения проекта прямо сейчас, или то, что потребуется для развития в будущем</p>	<p>Менеджер проекта, менеджер программы проектов, менеджер продукта</p>	
СММИ	<p>Управленцу по развитию продуктов, управленцу по разработке ПО, команде.</p> <p>Чтобы определить рекомендации,</p>	<p>В качестве основы использовать модель СММИ с ее описанием общих и специфических целей по успешным областям, а также рабочим практикам и подпрактикам по достижению целей.</p>	<p>Проектная команда должна одинаково трактовать все рекомендации СММИ.</p> <p>В случае перевода практик СММИ на русский язык провести несколько сессий</p>	<p>Менеджер программы проектов, менеджер продукта, руководитель разработки ПО</p>	

1	2	3	4	5	6
IDEAL	<p>Управленцу по развитию продуктов, управленцу по разработке ПО.</p> <p>Чтобы определить универсальный подход к стратегическому развитию продуктов</p>	<p>Создать сводную таблицу для проведения анализа.</p> <p>Провести анализ с адаптированными рекомендациями СММ по направлениям «Целевая Су» и «СУПр».</p> <p>Выявить цели измерений и провести анализ требуемых метрик.</p> <p>Определить первоочередные шаги по созданию нужных инструментов измерений процессов и автоматизации процессов</p>	<p>Обсуждений, направленных на то, чтобы суть каждой цели была понята.</p> <p>По факту приоритизации рекомендаций желательно привлечь консультанта в лице компании – оценщика на соответствие зрелости компании по модели СММ.</p> <p>Приоритизация обязательно должна быть согласована с топ-менеджментом.</p> <p>При обсуждении целей измерений право голоса имеют все лидеры по своим направлениям (архитектор, ведущий аналитик, старший менеджер по качеству).</p> <p>Следует вовлекать в обсуждение метрик высшее руководство и бизнес-дивизион (менеджеров продуктов)</p>	<p>Менеджер программы проектов, менеджер продукта, руководитель разработки ПО</p>	
	<p>Управленцу по развитию продуктов, управленцу по разработке ПО.</p> <p>Чтобы определить универсальный подход к стратегическому развитию продуктов</p>	<p>За основу берем описание модели IDEAL.</p> <p>Определяем единый подход на каждом этапе (фазе) модели, при необходимости дополняем деятельностью из других моделей и методологий.</p> <p>Нужно последовательно выполнить фазы:</p> <ul style="list-style-type: none"> <li>• Initiating (инициация);</li> <li>• Diagnosing (диагностика);</li> <li>• Establishing (разработка, планирование);</li> </ul>	<p>Команда, использующая этот паттерн, должна посмотреть на текущий производственный процесс шире, чем просто разработать конкретную версию продукта.</p> <p>Команда должна быть открытой, постоянно следить за причинами и стимулами для изменений, анализировать окружающую обстановку (контекст)</p>		

Таблица 4.2 (продолжение)

1	2	3	4	5	6
«Система управления производством (СУПр)»	Управленку по развитию продуктов, управленку по разработке ПО, команде. Чтобы определить рекомендации, конкретные практики и виды деятельности по процессу производства в целом и разработке ПО в частности	<ul style="list-style-type: none"> <li>• Acting (внедрение);</li> <li>• Learning (оценка).</li> </ul> <p>В начале каждой фазы есть смысл пересматривать и прототипировать, что конкретно команда будет делать для реализации каждой деятельности модели, в каком виде будет зафиксирован результат и с кем и как он будет согласован (см. паттерн ПУД).</p> <p>Паттерн «СУПр» по сути своей есть применение практик разработки ПО для построения управленческих процессов и, как следствие, для формирования и управления средой проекта. Моделируем концептуальную модель предметной области СУПр. Выявляем варианты использования СУПр.</p> <p>Описываем потоки взаимодействия СУ и СУПр, получая в качестве классов СУ, артефакты СУ, а в качестве методов – хорошие практики и рекомендации</p>	(см. паттерн «Среда проекта»)	Менеджер программы проектов, менеджер проекта, руководитель разработки ПО	
«Эффективный переговорщик»	Любому участнику проектной команды. Чтобы договориться в формате «выигрыш-выигрыш» (win-win) со второй стороной	<p>Понимание структуры переговоров.</p> <ul style="list-style-type: none"> <li>• Начало.</li> <li>• Предложение.</li> <li>• Торг.</li> <li>• Резюме договоренностей.</li> </ul> <p>Умение заранее подготовиться к переговорам.</p> <p>Знание ключевых рекомендаций и «магических формул» переговорщика на каждом этапе.</p> <p>Выполненное «домашнее задание» и помощник с проектной стратегией проведения переговоров и того, кто за что отвечает</p>	Менеджер должен быть доброжелателен, конкретен, вежлив, эмоционален, харизматичен	Любой уровень менеджера	

4. Процессы и процедуры.
  - 4.1. Оценка трудоемкости и стоимости.
  - 4.2. Работа с планом управления проектом.
  - 4.3. Работа с реестром рисков.
  - 4.4. ...
  - 4.5. Управление документами/артефактами.
  - 4.6. Проведение Peer Review (однорангового ревью).
  - 4.7. Правила согласования документов/артефактов, а также их утверждения.
5. Рабочее окружение.
  - 5.1. Развертывание рабочей среды.
  - 5.2. Реестр и правила работы с сетевыми общедоступными ресурсами и материалами.
  - 5.3. ...
  - 5.4. Создание архивных копий.
6. Обучение и повышение квалификации.

В книге «Путь аналитика» я уже останавливался на базовых правилах и приводил пример, где этот документ организован немного иначе. Основное отличие заключается в том, что вначале декларируют некоторые принципы работы и осуществления деятельности, а затем уже переходят к регламентирующей части.

В этих подходах нет противоречия. Насколько полными и насколько формальными будут базовые правила, зависит от организации и уровня зрелости компании.

В общем случае я рекомендую объединять информацию из обеих книг.

## 4.4. Паттерн «Выученные уроки»

Паттерн «Выученные уроки» неразрывно связан с паттерном «Хорошие практики», который мы рассмотрим далее, и может применяться практически ко всем паттернам в текущей книге и к любой области деятельности вашей организации.

Суть данного паттерна состоит в том, что в ходе выполнения любой работы возникают нештатные ситуации, ошибки, на которых полезно поучиться всем сотрудникам компании, чтобы не наступать все время на одни и те же грабли. Для этого в процессе выполнения работы и особенно по окончании выполнения каких-то работ создается табличка выученных уроков (ВУ) (табл. 4.3).

Таблица 4.3. Реестр выученных уроков

ИД	Проблема/ошибка	Кто и когда обнаружил	Способ устранения	Выученные уроки
УИД для ссылок на ВУ из других артефактов/документов	Наблюдавшаяся проблема/ошибка описывается как можно более беспристрастно, обезличенно и максимально абстрагированно от конкретного частного случая там, где это возможно	Указываются фамилия человека, обнаружившего проблему/ошибку, и дата этого события. Это делается для того, чтобы можно было узнать подробности из первых уст	Описывается, как справлялись с последствиями проблемы/ошибки, что конкретно было сделано, чтобы нивелировать или уменьшить возможные негативные последствия	Записывается, что можно и нужно сделать для того, чтобы проблема/ошибка не возникла с самого начала, какие проверки надо делать периодически или после фиксации каких-то событий, чтобы своевременно и максимально проактивно отреагировать на проблему/ошибку



Ведут эту таблицу все члены команды. Любой из участников в любой момент может ее дополнить или поправить в режиме управления изменениями.

Менеджер проекта периодически — как правило, после окончания этапа проекта/итерации — проводит анализ выученных уроков и принимает решение о том, имеет ли смысл отдельные выученные уроки рекомендовать для рассмотрения офису управления проектом (Project Management Office — PMO) или комитету/группе по совершенствованию (инжинирингу) процессов (например, Software Engineering Process Group — SEPG).



Пример заполнения реестра выученных уроков приведен в табл. 4.4.

Таблица 4.4. Образец реестра выученных уроков

ИД	Проблема/ошибка	Кто и когда обнаружил	Способ устранения	Выученные уроки
111	Заказчик прислал официальное письмо с требованиями разъяснений по сорванным срокам, указанным в плане-графике проекта. МП поправил план-график и устно согласовал изменения с комиссией по управлению изменениями (ССВ) проекта, зафиксировав электронным протоколом договоренности и изменения в базовой версии плана-расписания	Иванов, 14.01.2013	Отправить официальное письмо с разъяснениями. В качестве приложения отослать новую базовую версию плана-графика и распечатку электронного протокола	При внесении изменений в любые проектные артефакты, которые в ПУД согласуются или утверждаются заказчиком, помимо электронного протокола, фиксировать изменения направлением официального письма с приложением измененного артефакта



После рассмотрения компетентным органом организации выученный урок проекта может стать выученным уроком отдела/проектного офиса/компании. В этом случае такие выученные уроки оформляются как «хорошие практики» и/или вносятся в регламенты и процедуры организации. Смотрите соответствующий паттерн.

ВУ не обязательно должны оформляться в виде таблицы. Вот еще один пример конкретного ВУ. В этой компании просто велась переписка с темой ВУ.

### Ситуация

*На проекте внедрения заказчик затягивал момент формирования рабочей группы (группы ответственных за ведение работ по проекту) со своей стороны.*

*Вплоть до непосредственного выезда нашего специалиста отдела внедрения на объект внедрения заказчик **не подписал** общий план проекта, а также детальный план этапа № 1.*

*В первый же день нахождения нашего специалиста отдела внедрения на объекте выяснилось, что заказчик имеет несогласованные вопросы и, по его мнению, срок решения должен быть сокращен.*

*Обозначенные заказчиком вопросы переводятся на высшее руководство.*

*У нас нет никаких формальных писем, подтверждающих то, что сроки формирования рабочей группы и подписания планов сдвигались по вине заказчика.*

### Проблема

*Заказчик **уверен**, что сроки мероприятий, проводимых на объекте, наступают значительно раньше, чем сроки, обозначенные нами в плане.*

*Так как нет **согласованного плана** и никаких формальных подтверждений о задержках со стороны заказчика по его согласованию, то высшее руководство нашей компании вынуждено решать проблемы на личностном уровне.*

*Формально заказчик прав в своих ожиданиях.*

*Данная ситуация может привести к срыву контракта и потере заказчика.*

### Выученные уроки

- 1. План этапа № 1 должен содержать работы по согласованию «Общего плана» со стороны заказчика.*
- 2. К моменту завершения работ по этапу № 1 (обследование) все сроки должны быть формально согласованы с заказчиком.*
- 3. В случае невозможности выполнения п. 2 руководитель проекта обязан получить формальные подтверждения о задержках со стороны заказчика в виде официально-го или как минимум электронного письма.*
- 4. В случае невозможности выполнения п. 3 руководитель проекта должен послать официальное письмо-запрос о представлении информации (на фирменном бланке) руководителю/ответственному со стороны заказчика.*

А вот еще один конкретный пример ВУ из другой компании (табл. 4.5).

**Таблица 4.5.** Еще один образец реестра выученных уроков

Дата	Описание проблемы	Комментарий от IT	Дата устранения	Предложения по предотвращению подобных ошибок	Тривиальная ошибка (да/нет)
03.03.2010	Недопонимание в алгоритмах, техническое задание (ТЗ) между IT и бизнесом. Заказчик запутался в терминах, коэффициентах и их назначении	Заказчик не знаком со всеми тонкостями работы продукта, не знает, какие значения на что влияют	05.03.2010	Составлять четкое ТЗ, выделять время на разъяснение непонятных моментов, написание документации, с которой должен быть ознакомлен заказчик	Нет

В данной таблице графа «Тривиальная ошибка» выступала в том числе и как инструмент контроля усвоения сотрудниками ранее выявленных ВУ и рекомендаций, оформленных в виде хороших практик и процедур. Если ошибка/проблема помечалась как тривиальная, то с большой вероятностью это означало, что уже есть не только ВУ, но и хорошие практики (ХП) и процедуры, но сотрудник по каким-то причинам не усвоил их. В этом случае первое, что имеет смысл делать, — проверить, насколько понятно изложены все эти ХП и процедуры. Затем следует обучать сотрудника тем ХП и процедурам, в которых он допустил тривиальную ошибку. И только после этого — наказывать. Если ситуация повторится.

Мы сознательно привели несколько разных вариантов паттерна ВУ, во-первых, чтобы еще раз проиллюстрировать идею паттерна как некой рамки или направления приложения сил, а во-вторых, чтобы показать, что конкретный паттерн ВУ может реализовываться в разных форматах, главное, чтобы команде было удобно с ним работать и чтобы он играл свою главную роль — помогал формировать квалификационные стандарты компании.

## 4.5. Паттерн «Хорошие практики»

Этот паттерн почти всегда применяется совместно с паттерном «Выученные уроки». Смысл этого паттерна в формировании единых для всей организации полезных знаний и навыков, которые заработаны потом и кровью на ранее выполненных проектах/работах.

Есть старая, но точная истина: все инструкции пишутся кровью. В случае, когда какая-то практика не может быть зафиксирована инструкцией, но является крайне полезной для применения, или когда в компании/проекте вообще не создаются инструкции (а такое бывает очень часто — некогда, нет на это ресурсов, потом запишем), паттерн «Хорошие практики» является хорошим инструментом.

Этот паттерн отлично подходит для формирования процессного свода/базы знаний, который как один из инструментов формирования проектной среды и управления ею обеспечивает единое информационное поле навыков и знаний команды.

Проще всего этот паттерн реализовать в виде таблицы (табл. 4.6).

**Таблица 4.6.** Реестр хороших практик

ИД	Хорошая практика	Контекст	Ссылка на ВУ
УИД для ссылок на ХП из других артефактов/документов	Здесь описывается суть ХП	Здесь описывается контекст, в котором эту практику полезно применять	Ссылка на ВУ, с которым связана эта практика



Приведем пример, который можно вынести из примера в паттерне «Выученные уроки» (табл. 4.7).

**Таблица 4.7.** Пример реестра хороших практик

ИД	Хорошая практика	Контекст	Ссылка на ВУ
222	При внесении изменений в любые проектные артефакты, имеющие базовые версии, а также утверждающиеся/согласующиеся заказчиком, фиксировать изменения формальным официальным протоколом с приложением измененного артефакта	Управление базовыми версиями	111.1

Как можно видеть, в результате применения паттерна сформировалась рекомендация, отличие которой от ВУ заключается в том, что если ВУ носит рекомендательный характер и отдается на усмотрение МП, то ХП носит строго рекомендательный характер, то есть может не применяться только при согласовании с компетентным органом (например, офисом управления проектом).



В заключение добавлю, что когда организация приходит к осознанию необходимости упорядочения процессов и подходов к работе, то первое, с чего стоит начать построение процессов, — это выявить ВУ (СММІ 2) и сформировать ХП (СММІ 3), после чего «накручивать» сверху регламентную базу, корпоративный портал и системы управления качеством (QMS) с шаблонами и рабочими инструкциями.

ВУ и ХП гораздо более гибки, чем методологически ориентированные процедуры и практики, с точки зрения жизнедеятельности и применимости в организации и более дешевы в поддержке и сопровождении.

## 4.6. Паттерн «Своевременное и полное информирование»

Еще один важный аспект формирования здоровой проектной среды — обратная связь. Проектная среда вашего проекта должна базироваться на единой оценке происходящего и своевременно получать информацию из первых рук.

В этом паттерне я выделяю две составляющие: статус проекта и информационный бюллетень.

### 4.6.1. Паттерн «Статус проекта»

Этот паттерн я применяю для того, чтобы обеспечить единое видение всеми заинтересованными лицами проекта текущего состояния и статуса проекта. Далее приведен конкретный пример из моей практики — информационное письмо.



Как правило, информационное письмо содержит обновленную базовую версию плана-расписания, конкретный скоуп работ на неделю, возникающие в связи с их принятием задачи и поручения, проблемы и риски, статус по ПУД. В письме, которое я привожу как пример этого паттерна, вы найдете все эти составляющие.

«Коллеги, во вложении — обновленный план работ.

42	<input type="checkbox"/>		<input type="checkbox"/> Бизнес тестирование	35 дней	25.06.13 17:00	13.08.13 17:00	
48	<input checked="" type="checkbox"/>		Устранить обнаруженные Заказчиком ошибки / ответить на вопросы	15 дней	23.07.13 17:00	13.08.13 17:00	47
49	<input type="checkbox"/>		<input type="checkbox"/> Полное тестирование	14 дней	18.07.13 17:00	07.08.13 17:00	
50	<input checked="" type="checkbox"/>		Тестирование по еще не протестированным требованиям в соответствии с утвержде	5 дней	18.07.13 17:00	25.07.13 17:00	46
51	<input checked="" type="checkbox"/>		Определить критичность багов	1 день	25.07.13 17:00	26.07.13 17:00	50
52	<input checked="" type="checkbox"/>		Устранить критичные баги	5 дней	26.07.13 17:00	02.08.13 17:00	51
53	<input checked="" type="checkbox"/>		Провести выборочные тесты на развернутой площадке Заказчика	3 дня	02.08.13 17:00	07.08.13 17:00	52
54	<input type="checkbox"/>		<input type="checkbox"/> Тестирование алгоритма расчета прогноза прибытия IC	10 дней	02.08.13 17:00	16.08.13 17:00	
59	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Тестирование производительности	8 дней	25.06.13 17:00	05.07.13 17:00	
62	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Интеграционное тестирование	5 дней	25.06.13 17:00	02.07.13 17:00	
66	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> Приемка кода	10 дней	25.06.13 17:00	09.07.13 17:00	
69	<input type="checkbox"/>		<input type="checkbox"/> Приемка пользовательской документации	6 дней	23.07.13 17:00	31.07.13 17:00	
70	<input checked="" type="checkbox"/>		Изучение	2 дня	23.07.13 17:00	25.07.13 17:00	47
71	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Провести демонстрацию веб-решения и МП	1 день	25.07.13 17:00	26.07.13 17:00	70
72	<input checked="" type="checkbox"/>		Выполнение бизнес-сценариев	3 дня	26.07.13 17:00	31.07.13 17:00	71

На этой неделе:

- ◆ Иван, прошу прислать/выложить на FTP актуальную версию пользовательской документации;
- ◆ демонстрацию веб-решения и МП считаю выполненной благодаря нашей с вами активной переписке по исследованию проблем МП в ходе бизнес-тестирования (но есть открытые вопросы во вложении);

- ◆ **Иван**, в аттаче также письма, на которые я не получил ответа;
- ◆ **Иван**, жду от Вас обновленного “ZZZ-СВОД-Спецификации и документы v.1.6”, чтобы не иметь разночтений.

*С уважением,  
Перерва А. Д.»*

Думаю, что данный паттерн меньше всего нуждается в пояснениях, тем не менее есть два важных момента: такое информационное письмо должно рассылаться всем заинтересованным лицам проекта как минимум один раз в две недели (зависит от того, на каком этапе проекта мы находимся и насколько интенсивно ведутся работы); обе стороны должны иметь возможность свободно править план-расписание и ПУД (в примере называется «СВОД-Спецификации и документы»). Смотрите связанный паттерн ПУД.

Еще один пример отчета о статусе проекта для всех заинтересованных лиц вы сможете найти, перейдя по ссылке [http://bit.ly/pmway\\_statereport](http://bit.ly/pmway_statereport).

## 4.6.2. Паттерн «Информационный бюллетень»

Этот паттерн особенно полезен:

- ◆ в случае антикризисного управления, когда надо зажечь людей;
- ◆ как инструмент вовлечения в совершенствование процессов;
- ◆ как инструмент открытости и прозрачности, если вы руководитель функционального отдела или руководитель разработки ПО.

Паттерн предназначен для широкого информирования сотрудников и руководства о процессах, происходящих в отделе/подразделении. Использование этого паттерна позволяет показать сотрудникам, что вся работа строится прозрачно, ничего от сотрудников и топ-менеджеров не скрывают, все видят одинаковую картину. Это особенно важно для больших проектов в критическом состоянии или проектов по совершенствованию процессов.

Далее привожу пример такого информационного бюллетеня.



---

### «Что значительного случилось за прошедшие две недели?»

1. На этой неделе в офис пришел новый сотрудник отдела, **Иван Иванов**. В его обязанности будет входить поддержка пользователей офиса (подготовка рабочих мест новых пользователей, выявление и устранение проблем на рабочих компьютерах пользователей). Я поздравляю Ивана с началом работы в компании, желаю профессиональных успехов и скорейшего становления в качестве полноценного члена нашей дружной команды!
2. **Начата реорганизация ЦР (центра разработки) ПО**, о чем вы можете прочитать в письме во вложении. Основные выгоды и цели, преследуемые реорганизацией:

- ♦ разгрузить **Петра Петрова** для того, чтобы максимально использовать его знания и опыт в разработке новых продуктов, направленных на достижение стратегических целей — создание новых продуктов. Петр опытный разработчик и уже достаточно сильный менеджер, что позволит ему придать новый импульс и форсировать развитие этого направления.  
Для компании это значит увеличить прибыль через создание нужных ей инструментов;
  - ♦ распределить усилия разработки в IT по функциональным областям и выполнять все задачи более **предсказуемо** и прозрачно.  
Для компании это значит повысить эффективность работы IT и, как следствие, **повысить эффективность работы основного производственного процесса**;
  - ♦ позволить построить более эффективные коммуникации с бизнесом за счет диверсификации обращений по разным задачам к разным людям.
3. Начата комплектация центра разработки ПО и отдела техподдержки (ТП) для уменьшения нагрузки и обеспечения ресурсами возможности выполнения текущих задач. Приняты решения по найму  $X$  разработчиков в ЦР ПО и  $X$  — в отдел ТП. Идет поиск кандидатов. Эти меры также направлены на цели обозначенные в п. 2.
  4. В проекте “**1С CRM**” **Владимиром Владимировым** и **Яковом Яковлевым** успешно выполнены задачи по доработке и настройке подсистемы для отдела телемаркетинга, также созданы инструкции и обучены руководители. Отдел телемаркетинга начал использование своей части системы в промышленной эксплуатации. Параллельно выполняется стабилизация кода и разработка остальной функциональности 1С CRM.
  5. **Начаты работы по решению проблемы** “Пересмотреть требования и сформулировать заявку на закупку оборудования для обеспечения нормальной работы Task Manager”. Данная система — основное средство коммуникации в ходе рабочих процессов между отделами техподдержки — оптимизации и менеджерами. Кроме этого, на данном оборудовании действует система ReadMine, обеспечивающая работу внутри отдела техподдержки и SVN со всеми версиями программных продуктов. Требуется подготовить спецификацию и обоснование потребности в приобретении нового сервера. Ответственные — **Т. Требовательный** и **Ж. Железячкин**.
  6. Введена хорошая практика по созданию отчетов о состоянии задач и проектов группы поддержки и срочных доработок ЦР ПО. Отчет создается еженедельно по вторникам с 11:00 до 13:00 и рассылается учредителям и техническому директору. Это обеспечивает своевременное информирование и прозрачность работ в группе поддержки и срочных доработок.
  7. Продолжает использоваться практика выученных уроков. ВУ являются частью нематериальных активов компании и позволяют учиться на ошибках и просчетах, допущенных в повседневной деятельности. Вы можете посмотреть уже существующие уроки и поделиться с другими членами команды своими уроками. Адрес: [https://выученные\\_уроки/index](https://выученные_уроки/index).

За прошедший период **добавилось несколько важных уроков**, на которые я рекомендую обратить внимание.

- ◆ Как избежать ошибок при введении в эксплуатацию новых модулей.
- ◆ Как более точно озвучивать сроки выполнения задач бизнеса, в которых участвует ИТ.
- ◆ Составление правильного ТЗ».



В подобных информационных бюллетенях очень важно:

- ◆ выдерживать нейтрально-доброжелательный тон;
- ◆ критические моменты описывать максимально беспристрастно и обезличенно и, напротив, положительные факторы всегда подкреплять примером работы конкретных сотрудников с указанием фамилии и описанием вклада, который тот внес в общее дело;
- ◆ показывать связь мероприятий и задач сотрудников с задачами и целями подразделения/проекта;
- ◆ показывать связь с финансовыми целями компании или иным способом информировать о значимости описываемой деятельности для компании.

Кроме этого, любая плановая активность, упомянутая в этом письме, должна сопровождаться указанием ответственного сотрудника.

Эти простые правила являются эффективными инструментами лидера, которые позволят вам сплотить команду в кризисные моменты и вовлечь сотрудников в общую созидательную и конструктивную деятельность.

## 4.7. Паттерн «Управление средой проекта»

А сейчас я хочу предложить вам посмотреть на управление проектом через призму управления средой проекта.

В замечательной книге Дуга ДеКарло «Экстремальное управление проектом» [10] проект рассматривается как экосистема. Позволю себе процитировать некоторые высказывания из этой книги.

*«Авторы книги *Surfing the Edge of Chaos* Джиоя, Миллеман и Паскаль рассказывают историю о молодой паре, получившей домик в наследство от своей тетушки. Работая над обстановкой в одной из комнат, они перепробовали различные отделочные материалы и цветовые решения, пытаясь добиться оптимального результата. Но все было тщетно. Однажды лампочка, установленная в этой комнате, перегорела. Когда ее стали менять на новую, обнаружилось, что колба лампочки была желтого цвета. Сразу стало понятно: желтый цвет придавал неприятный оттенок всей комнате, влияя на цветовую схему и не позволяя достичь запланированного эффекта. Как бы они ни пытались изменить интерьер комнаты, желтый цвет всегда оказывал на него пагубное влияние.*»

*В этой истории среда (желтый оттенок) оказывала большее влияние на результат, чем содержание (цвет обоев и материала). Из-за этого работа над содержанием комнаты была пустой тратой времени и денег и завела молодую пару в тупик.*



Чтобы осознать важное различие между содержанием и средой, представьте, что ваш проект — это цветущий сад, экосистема, состоящая из цветов, почвы и погодных условий (табл. 4.8).

В данном случае цветы являются содержанием, а почва и погодные условия представляют собой среду. И почва, и погодные условия оказывают решающее влияние на выживание, рост и красоту цветов. Если почва отвратлена, а погода негостеприимна, цветы погибнут, как бы о них ни заботились.



Основной ошибкой руководителей экстремальных проектов является нежелание управлять проектной средой, реагируя на условия, в которых проходит процесс разработки, — эмоции, мысли и отношения участников.

*На основе сказанного создается “должностная инструкция” руководителя экстремального проекта, сгруппированная по направлениям».*

**Таблица 4.8.** Проект как экосистема

	<b>Сад</b>	<b>Проект</b>
Среда: условия, окружающие содержание проекта	Погода: солнце, дождь, осадки, температура воздуха	Внешние условия: конкуренция, новые технологии, законодательство, сторонние поставщики, предпочтения заказчиков, экономика, политика
	Почва: химические удобрения, плодородие, глубина, токсичность	Внутренние условия: качество мыслей, эмоций и отношений, окружающих проект; ожидания участников, системная политика; общее отношение к проекту
Содержание	Цветы	Разрабатываемый продукт или услуга
Проектная роль	Фермер: очистить почву; создать парниковый эффект; следить за ростом цветов; посоветоваться с садовником насчет изменений (например, ирригации), связанных с погодой	Руководитель проекта: создать позитивное отношение к проекту; управлять энергией, направляя мысли, потоки и отношения; управлять проектом как бизнес-предприятием
Инструменты	Трактор, удобрения, орошение, защита	Управление отношениями; помощь, переговоры, межличностные и политические навыки, технология, инфраструктура
Техническая роль	Садовник: уход за цветами	Технический администратор/менеджер по разработкам: создание продукта и/или услуги

### **Управление политическим ландшафтом**

- ◆ Найти спонсора, с которым можно добиться успеха.
- ◆ Установить хорошие взаимоотношения и рабочие соглашения с ключевыми участниками.
- ◆ Убедиться, что спонсор и участники проекта выполняют условия успешной реализации проекта.
- ◆ Создать взаимовыгодные отношения с поставщиками и руководителями взаимозависимых проектов.
- ◆ Вести и влиять без властного давления.
- ◆ Договариваться (относительно людей, фондов, капитального оборудования, сроков и т. д.).

### **Создание результатов**

- ◆ Создайте инфраструктуру управления проектом (процессы, инструменты, шаблоны, общение в реальном времени).
- ◆ Поощряйте принятие решений.
- ◆ Демонстрируйте ранние и текущие результаты.
- ◆ Согласовывайте с заказчиком промежуточные и окончательные результаты проекта.
- ◆ Следите за достигнутым прогрессом.

### **Управление настройками**

- ◆ Превратите разрозненную группу в функциональную единицу.
- ◆ Создайте условия для работы, вдохновляющие команду.
- ◆ Как можно быстрее устраняйте препятствия, мешающие нормальной работе.
- ◆ Найдите «вау»-фактор — то, что делает проект интересным и запоминающимся, а его результаты — важными и даже революционными. Он раскрывает в людях стремление участвовать в проекте.
- ◆ Способствуйте разрешению межличностных конфликтов.
- ◆ Не оставляйте без внимания и поощряйте индивидуальные и групповые достижения.
- ◆ Учите и ободряйте.

### **Обеспечение ценности бизнеса**

- ◆ Поддерживайте стратегическую, а не локальную ориентацию.
- ◆ Сосредоточьтесь на экономическом эффекте.
- ◆ Постоянно задавайте четыре вопроса бизнеса.
  - ◆ Кому это нужно и зачем?
  - ◆ Что для этого нужно сделать?

- ◆ Сможем ли мы с этим справиться?
- ◆ Оно того стоит?

Давайте остановимся чуть подробнее на обеспечении ценности бизнеса. Это самое главное с точки зрения собственно существования проекта и это крайне важно с точки зрения управления средой проекта, так как эти четыре вопроса постоянно напоминают всем участникам о том, что проект в первую очередь — бизнес-предприятие: главной целью является достижение значимых результатов на каждом этапе, включая стадию реализации, которая наступает после достижения всех результатов проекта.

Дуг ДеКарло рекомендует постоянно держать в фокусе и постоянно задавать эти четыре вопроса, работая с ними следующим образом.

- ◆ Составьте перечень всех неизвестных факторов, которые могут оказать влияние на график работ или бюджет проекта. Используйте этот список, чтобы проинформировать спонсора проекта. Вы окажете ему большую услугу.
- ◆ Объясните, что ответ на второй вопрос должны дать вы сами.
- ◆ Настойчиво требуйте присутствия представителя заказчика на все время реализации проекта для определения истинных требований. Укажите, что это позволит дать ответ на первый вопрос и подготовит почву для ответа на второй.
- ◆ Объясните спонсору проекта, что на каждом этапе реализации проекта ему придется принимать решение о том, стоит ли двигаться дальше, и отвечать на четвертый вопрос («Оно того стоит?»), что, в свою очередь, позволит сохранить финансовый контроль над проектом.

Применение четырех вопросов бизнеса на практике означает постоянное обновление бизнес-планов для отражения текущих ожиданий и перспектив. А для выявления ожиданий и достижения согласованных позиций существует механизм достижения согласованных позиций, о котором я рассказывал в первой книге.

### Механизм достижения согласованных позиций




---

Я хочу поделиться с вами неоднократно проверенной на практике обобщенной WBS работ по согласованию позиций и ожиданий.

---

1. Формально определяйте цели согласования позиций.
2. Делайте это в форме программы проектов.
3. Определяйте устав программы проектов хотя бы в форме протокола совещания.
4. Определите перечень областей, в которых вы должны достичь согласованных позиций.
5. Определите по одному ответственному за каждую область.
6. Определите перечень заинтересованных лиц (ЗЛ), которые должны быть опрошены каждым ответственным. Могут появиться виртуальные ЗЛ, на-

пример методологии или стандарты. В этом случае в качестве ожиданий берутся положения этих материалов.

7. Ответственный за область должен встретиться с каждым из ЗЛ и сформировать единый и непротиворечивый перечень ожиданий ЗЛ от области, за которую он отвечает.
  - ◆ Ни одно из выявленных ожиданий не удаляется.
  - ◆ Если есть спорные ожидания, они помечаются как спорные, фиксируются обе точки зрения.
8. На общем собрании рассматриваются сформированные реестры ожиданий.
  - ◆ В первую очередь рассматриваются спорные ожидания. Этот процесс может занять много времени. Лучше выполнять эту деятельность итеративно.
  - ◆ Во вторую очередь все выявленные ожидания просматриваются, приоритизируются и утверждаются.
9. Намечайте инициативы для выполнения ожиданий в соответствии с приоритетами. Повторите п. 5–7 для инициатив. Проведите трассировки от ожиданий к инициативам.
10. Формируйте портфель проектов реализации согласованных инициатив.
11. Выполнение проектов передайте в офис управления проектами (Project Management Office – PMO).
12. Формируйте реестр рисков уровня программы проектов и управляйте им на периодической основе в тесном контакте с менеджерами проектов.

Для того чтобы ответить на первый вопрос бизнеса «Кому это нужно и зачем?», вам придется регулярно сталкиваться с противоречивыми ожиданиями, целями и мотивами. Приведенный механизм поможет не просто получить ответ на данный вопрос, но и в процессе формулировки ответов добиться максимально согласованной со всеми единой позиции.

Резюмирую сказанное.



---

Для достижения успеха проекта вам необходимо будет интегрировать политические аспекты и аспекты бизнеса с технической стороной предприятия. Технический аспект связан с содержанием проекта — разрабатываемыми продуктами и услугами. Политические аспекты и аспекты бизнеса связаны со средой проекта (окружающим миром), в которой происходит разработка новых продуктов и услуг.

Управление средой, таким образом, сводится к управлению ожиданиями участников проекта. Иными словами, это управление отношениями. И я показываю вам эффективный механизм, который поможет применять паттерн четырех вопросов бизнеса и должностной инструкции менеджера экстремального проекта в своей деятельности.

---

В заключение добавлю несколько слов о поведенческом аспекте данного паттерна.

Модель экстремального управления проектами по Дугу ДеКарло состоит из наборов правил, ценностей, умений, инструментов и практик, основанных на принципе изменений и неопределенности и составляющих программную и аппаратную часть экстремального управления проектами:

- ◆ четыре ускорителя — принципы, которые дают свободу мотивации и инновациям;
- ◆ десять общих ценностей — набор ценностей, устанавливающих доверие между заинтересованными сторонами;
- ◆ четыре вопроса бизнеса — вопросы, ответы на которые помогают часто и быстро выдавать заказчику ценные результаты;
- ◆ пять критических факторов успеха — навыки и инструменты, а также организационная поддержка, играющие ключевую роль в достижении успеха.

Я не буду в этой книге останавливаться на всех моментах подробно — для этого лучше обратиться к первоисточнику, и я настоятельно рекомендую вам это сделать. Раскрою здесь только то, что нужно для целостного восприятия читателем паттерна «Управление проектной средой». Организационный аспект паттерна я рассмотрел ранее.

Поведенческий аспект этого паттерна формируют четыре ускорителя и десять общих ценностей.



#### Четыре ускорителя

1. **Подружитесь с изменениями.** Изменения воспринимаются как **новая возможность**, анализ и реагирование на изменения повышает шансы на **достижение желаемого результата** (который может существенно отличаться от запланированного). Фраза «Подружитесь с изменениями» означает, что вы должны принять изменения и двигаться дальше.
2. **Играйте на страстях людей.** Слово «проект» содержит в себе некий угнетающий подтекст. Люди будут работать с большим энтузиазмом, если будут знать, что **выполняют определенную миссию**, если они будут рассматривать проект не как «проект», а как причину их действий. Применение на практике второго ускорителя означает: **вы должны показать людям, что их работа является частью чего-то большего, дав им четкое представление о целях и средствах.**
3. **Сделайте людей хозяевами достигнутых результатов.** Люди заботятся о том, что они создают. Вы должны **доверять профессиональным способностям** людей и **дать им возможность повлиять на достижение успеха проекта в целом**, включая измерение производительности проекта.
4. **Не усложняйте процесс.** Для экстремального проекта старый добрый принцип «чем меньше, тем лучше» — не пустой звук. Это очень серьезно. На практике меньшее становится большим: **меньше процессов, меньше управления, меньше политик и стандартных процедур.**

Как видите, во многом я уже писал об этих принципах в паттернах «Хорошие практики», «Выученные уроки» и «Своевременное и полное информирование». Найдите несколько минут и проанализируйте перечисленные паттерны на предмет применения в них четырех ускорителей. Это крайне полезное упражнение.

**Десять общих ценностей — это система ценностей**, основанная на непоколебимой вере участников проекта в то, что, работая сообща, они обязательно добьются успеха даже в неблагоприятных изменчивых условиях.

◆ **Человеческие ценности.**

1. **Свободное общение** — действуйте последовательно и открыто говорите правду о хорошем, плохом и ужасном, не боясь наказания.
2. **Главное — люди** — устраните все препятствия и дайте людям возможность качественно выполнять свою работу.
3. **Качество жизни** — убедитесь в том, что соблюден необходимый баланс между работой и личной жизнью.
4. **Отвага** — действуйте несмотря на свой страх; действуйте наперекор страху, если знаете, что поступаете правильно.

◆ **Ценности процесса.**

5. **Взаимодействие с заказчиками** — постоянное взаимодействие и общение с заказчиками в ходе всего предприятия в противовес простому получению требований и отсутствию обратной связи.
6. **Быстрые отказы** — поиск кратчайшего пути к провалу проекта посредством выполнения самой трудной, рискованной или важной работы на раннем этапе реализации проекта.
7. **Прозрачность** — держите на виду у всех все, что можно: проектные планы, достигнутый прогресс, разработанные продукты, отчеты, ответственность за выполненную работу.

◆ **Ценности бизнеса.**

8. **Ясность целей** — необходимо видеть не только цель проекта, но и всю картину в целом, то, для чего в первую очередь был инициирован проект.
9. **Ориентация на результат** — сосредоточьтесь на достижении результатов, а не на текущих задачах.
10. **Ранний результат** — дать заказчику нечто, что он сможет использовать как можно быстрее.

Найдите еще несколько минут и проанализируйте паттерны этой книги на предмет применения в них этих десяти общих ценностей. Это еще одно крайне полезное упражнение.

Не читайте следующее предложение, пока не выполните упражнение. Выполнили?

Отлично! Значит, вы знаете «секрет» паттернов этой книги — практически во всех них так или иначе встречаются эти четыре принципа и десять общих ценностей.

Если примеры и описания паттернов из этой книги помогут вам дополнить гениальные формулировки Дуга ДеКарло и послужат конкретными прикладными примерами и инструментами, я буду считать миссию книги выполненной.

Не забудьте, пожалуйста, сообщить мне об этом на страничке «Путь аналитика» в Facebook: <https://www.facebook.com/pages/путь-аналитика/582969531717404> — или на сайте книги <http://bp4you.ru>.

Итак, в этой главе я рассмотрел некую философскую основу управления и формирования проектной среды и управления ею, вспомнил про инструмент достижения согласованных позиций (или универсальной WBS, как я сам часто его называю при общении с коллегами). Мы вникли в глубокий смысл и основополагающую сущность ускорителей, вопросов бизнеса и факторов от Дуга ДеКарло. Пожалуй, этот паттерн менее всех остальных похож на адаптируемое руководство к действию. Он начнет работать, только если вы прочувствуете его, посмотрите сами и поймете, как перечисленные понятия раскрываются с самых разных сторон в разных паттернах, видоизменяя свою форму, но не изменяя своей целевой функции и не теряя практической пользы.

## 4.8. Паттерн «Открытые вопросы»

Рассмотрим еще один очень полезный паттерн формирования проектной среды — «Открытые вопросы». Он принесет максимальную пользу, если вы подхватываете уже разрабатываемый проект и вам нужно в кратчайшие сроки разобраться в его цели, предмете и среде.

Решая очевидную задачу накопления вами знаний, этот паттерн позволяет также в качестве сопутствующей цели добиться сплоченности команды и ориентированности на результат. Это получается за счет вовлечения всей команды в обсуждение спорных, проблемных и открытых вопросов.

Документ «Открытые вопросы» создается в свободной форме, однако я рекомендую иметь сводную таблицу управления статусами вопросов следующего формата (табл. 4.9).

**Таблица 4.9.** Реестр открытых вопросов

ИД вопроса	Суть вопроса	Ответственный за вопрос	Кто обнаружил	Когда обнаружил	Ссылка на ответ/ответ на вопрос	Кто дал ответ	Статус вопроса

Статус открытого вопроса в простейшем случае имеет состояния «открыт», «пояснен», «закрыт». В статус «пояснен» вопрос переводит только ответственный за вопрос, которого назначил менеджер проекта. То есть если вопрос касается тре-

бований, то ответственным за вопрос будет назначен системный аналитик или архитектор, если управления — менеджер проектов. В статус «закрыт» вопрос переводит только менеджер проекта по факту всестороннего анализа рисков.



Таким образом, паттерн «Открытые вопросы» дополняет паттерн ПУД и реестр рисков. В цепочке управления проектом при создании ПУД полезно крайне внимательно по каждому объекту поставки сформировать перечень открытых вопросов и систематизировать их в документе «Открытые вопросы». По мере работы с открытыми вопросами рекомендуется анализировать возможные риски и их последствия и работать с реестром рисков.

В моей практике этот паттерн отлично себя зарекомендовал, когда мне приходилось подхватывать серьезные и масштабные проекты на заключительных этапах их реализации и стояла задача как можно быстрее вникнуть в проект и его предмет. Этот паттерн не только позволял снять действительно открытые вопросы, но и вовлечь даже достаточно неконструктивно настроенные стороны в дискуссию, что в конечном итоге помогало сформировать реально работающую команду.

## 4.9. Паттерн «Система авторизации работ»

Паттерн «Система авторизации работ (Work Authorization System (WAS))» — последний паттерн в группе общих паттернов. Его цель — обеспечить выполнение нужной работы в правильные сроки.

Основные цели WAS:

- ◆ вертикальная и горизонтальная прозрачность работ;
- ◆ возможность предупредительных действий со стороны руководителя функционального подразделения/менеджера проекта в начале рабочей недели за счет эффективного контроля планируемых к выполнению работ в течение недели и времени на ее выполнение со стороны сотрудников;
- ◆ возможность корректирующих действий со стороны руководителя функционального подразделения/менеджера проекта по окончании рабочей недели посредством оперативного контроля отметок о фактическом времени выполнения запланированных работ, сделанных сотрудниками рабочей группы;
- ◆ самоорганизация и планирование сотрудниками собственной работы;
- ◆ возможность переназначения задачи другому исполнителю без потери времени на передачу знаний;
- ◆ формирование единого рабочего информационного пространства, объединяющего все артефакты и работы в рамках всех проектов, рабочих групп и отделов в связанное целое;

- ◆ эффективное использование и постоянное усиление синергетического эффекта единого информационного пространства WAS;
- ◆ возможность снимать метрики и принимать управленческие решения, основанные на количественных измерениях на уровне проекта/рабочей группы/подразделения;
- ◆ возможность снимать метрики и принимать управленческие решения, основанные на количественных измерениях на уровне организации.

В самом простом виде паттерн WAS может быть реализован с помощью реестра работ (табл. 4.10) (см. также паттерн «Управление через реестры»).

**Таблица 4.10.** Реестр работ

Название	Назначение	Состав	Комментарий по работе с реестром
Реестр работ	Управление работами проектной группы, собственными работами	№ по порядку. Наименование работы. Описание работы. Ответственный. Состояние. Основание авторизации работы. Плановая дата завершения. Фактическая дата окончания. Дата создания. Приоритет. Отчет о ходе выполнения. Номер связанной работы	Описание работы и отчет о ходе ее выполнения — поля, в которых в свободной форме, начиная с более поздней даты, приводятся комментарии по выполнению работы. Я рекомендую при занесении нового комментария всегда указывать дату и с красной строки начинать сам комментарий. Это позволит проводить ретроспективный анализ. Этот реестр может вести любой участник рабочей группы, но никто не имеет право удалять из него информацию. Неактуальным работам просто присваивается статус «прекращена»

Даже в таком виде основная цель паттерна — выполнять в нужный момент времени необходимые работы, авторизованные уполномоченным на принятие решений о старте работ лицом, — будет достигнута.

## 4.10. Паттерн «Эффективный переговорщик»

В свое время мне посчастливилось посетить тренинг по агрессивным переговорам. Сейчас я, к сожалению, не вспомню, кто именно его проводил. Это был 24-часовой тренинг, на котором члены группы не слушали лекции в пассивном режиме, а активно вели переговоры друг с другом. Одна часть группы продавала слона другой, продавали себя группе лиц, продавали идеи представителям инвестора и т. п.

В наследство остались опыт, знания и материалы, которые я дополнил и переработал в небольшой, но действенный паттерн — шаблон документа по подготовке и проведению переговоров, которым я и хочу с вами поделиться на страницах этой книги. Вы можете скачать этот шаблон на сайте книги, перейдя по ссылке [http://bit.ly/pmway\\_negotiation](http://bit.ly/pmway_negotiation).

Основа паттерна — в понимании структуры переговоров. Большинство переговоров можно структурировать следующим образом.

1. Подготовка к переговорам.
2. Начало переговоров.
3. Предложение.
4. Торг.
5. Резюме договоренностей.

Не претендуя на освещение всех тонкостей переговорного процесса, расскажу только о некоторых основных моментах, которые я выделил лично для себя и которые часто помогали мне в переговорах на самых разных уровнях.

**Подготовка к переговорам.** На данном этапе вы должны четко понять, каковы цели переговоров и возможные цели противоположной стороны, выработать стратегию проведения переговоров и продумать ее альтернативы, выработать план проведения переговоров, узнать состав участников переговоров с противоположной стороны, создать команду по проведению переговоров, принимая во внимание личные и профессиональные характеристики членов команды и противоположной стороны, и проработать с командой важные моменты, которые могут возникнуть на переговорах.

**Начало переговоров.** Цель этого этапа — создание атмосферы переговоров. В идеале это атмосфера доверия и взаимного уважения. Точки зрения могут расходиться, одна сторона может стремиться продать подороже, в то время как другая — купить подешевле, но атмосфера взаимного уважения и доверия, основывающаяся на том факте, что стороны собрались, чтобы достичь прогресса и договориться к удовлетворению обеих сторон, — залог если не успешных, то как минимум интересных и комфортных переговоров.

На этом этапе встречаются по одежке, рукопожатию, улыбке, здесь важно уметь поговорить о погоде по-английски. Желательно больше слушать вторую сторону, раскрывать ее. Этот этап похож на разведку: нужно наблюдать и анализировать. Хорошо, если вы на переговорах не один и ваш партнер поможет вам в их проведении. Проверяйте свои догадки, анализируйте потребности и ожидания, которые озвучивает вторая сторона. Важный результат этого этапа переговоров — сформулированная или уточненная повестка встречи.

**Предложение.** Цель этого этапа — презентовать предложение, выделив при этом наиболее интересные второй стороне свойства и характеристики продукта, подав их в нужном и выгодном вам свете. Выделить интересные характеристики нужно, аккуратно и очень деликатно проведя сравнение с конкурентами и не обидев их при этом.

В ходе презентации предложения нужно определить спорные вопросы и границы возможного обсуждения. Очень важно на этом этапе быть конкретным и привязывать свои предложения к потребностям второй стороны. Но самое важное — это формула рассказа о своем товаре, услугах и о себе:

$$\begin{aligned} & \text{свойство/характеристика/знания} + \\ & + \text{факт/портфолио/подтвержденный опыт} = \\ & = \text{выгода для второй стороны.} \end{aligned}$$

На этом этапе мы впервые сталкиваемся с понятиями верхней и нижней границ. Верхняя граница — это то, что в идеале хотите получить вы. Нижняя граница — это те условия, ниже которых вы ни при каких обстоятельствах не опускаетесь. Крайне важно заранее, еще до встречи определить список потребностей и выгод для второй стороны. Это именно те догадки, которые надо проверять по возможности на первом этапе, и это основной костяк, вокруг которого имеет смысл вести переговоры на втором этапе, уточняя и дополняя выявленные вами выгоды второй стороны. И снова вспоминаем о пользе помощника на переговорах. Он поможет зафиксировать то, что вы не сможете, так как будете слишком вовлечены в беседу. Кроме того, в ходе переговоров хорошо менять роли сопровождающих вас людей, участвующих в переговорах. Один высказался, выслушали другую сторону. Высказался второй — со своими аргументами, со своей формой подачи материала. Это оживляет процесс переговоров, а когда о чем-то говорят несколько человек, психологически это всегда убедительнее, чем когда говорит один.

Далее в ходе проведения переговоров мы переходим к описанию возможностей нашего товара. Ранее я уже упомянул основную формулу рассказа о товаре, здесь она максимально эффективна: выгода — свойство — факт. В коммуникациях должны быть задействованы как информационная, так и эмоциональная составляющая переговоров. Не стесняйтесь, старайтесь эмоционально рассказывать о своем товаре, своих проектах. Это всегда помогает — увлекает собеседника и увеличивает доверительное отношение друг к другу, возникшее на этапе начала переговоров.

Магические формулы здесь следующие: «И вы и я понимаем...», «Я готов, если вы...». Для этой стадии требуется заготовить «домашнюю работу» в виде таблицы с перечнем сильных и слабых сторон вашего предложения. Готовьте аргументацию и риторические приемы заранее.

Теперь мы плавно переходим к стадии «оценка предмета переговоров». «Домашняя работа» для нее заключается в заполненной таблице вида: «Почему я»; «Какие риски». Следует разобраться в том, что станет наилучшей альтернативой переговорам: что сделает вторая сторона, если вы не договоритесь, и что в таком случае сделаете вы. Это понимание является частью общей картины и поможет вовремя скорректировать русло переговоров или вообще почувствовать и уловить еще более интересные возможности.

Например, проходили переговоры, в ходе которых нужно было убедить заказчика в целесообразности сдвига сроков проекта. В качестве одного из вариантов

наилучшей альтернативы было отмечено привлечение в проект дополнительного ресурса по разработке за их счет. Когда переговоры коснулись этой альтернативы и наша сторона была вынуждена озвучить ее, удалось договориться с заказчиком о том, что разработчиков мы привлекаем в рамках установленного бюджета, а заказчик выделяет для проекта ресурсы на тестирование продукта и готов участвовать в подготовке детальных тест-кейсов. Таким образом было достигнуто «вин-вин»-решение, которое, формально говоря, не привело к буквальному достижению цели переговоров с нашей стороны — сроки не были сдвинуты. Но наш выигрыш заключался в том, что мы получили «бесплатный» ресурс, а выигрыш заказчика — в том, что мы по-прежнему старались уложиться в сроки.

**Торг** — следующий этап переговоров. Я не очень люблю его, как и большинство технарей, но несколько простых рекомендаций помогли мне договариваться и с теми, кого называют продавцами от Бога. Вот одна из них: следует выяснять причину любого несогласия.



---

Вообще главный вопрос на этом этапе не «Сколько стоит?», как могло бы показаться, а «Почему?». «Меня не устраивает ваша цена». — «Почему? Вы считаете, что можно создать продукт с желаемыми вами характеристиками и качеством за значительно меньшие деньги? Вероятно, мы можем поговорить о сроках?..»

---

«Почему?» — вот главный вопрос данного этапа. И только узнав причину несогласия, стоит делать ответное предложение. Из первой рекомендации логически вытекает следующая: разбивать уступки на маленькие кусочки, повышая при этом значимость уступок с вашей стороны. Особенно первой. Еще два золотых правила этого этапа: после предложения всегда следует условие; спуститься (по цене, по условиям) можно всегда, подняться — никогда!

«Домашнее задание» для данного этапа переговоров выглядит как таблица «Чего хочу я» с графами «Я хочу! (Верхняя граница)», «Допустимо пожертвовать/отдать», «Нижняя граница». Как это ни парадоксально, но часто в переговорах четкое и ясное обозначение именно нижней границы является более важным, чем обозначение верхней. Потому что, опустившись ниже этой границы, можно не только проиграть текущие переговоры, но и создать серьезные проблемы для всего бизнеса.

**Резюме договоренностей** — это заключительный этап переговоров. Здесь кратко формулируются все ключевые договоренности и намечаются дальнейшие шаги, обязательно со сроками и конкретными фамилиями тех, кто будет заниматься этими вопросами с обеих сторон.



---

Напоминаю, что на сайте книги вы найдете шаблон документа «Подготовка к переговорам», в котором есть еще целый ряд рекомендаций, дополняющих изложенное.

---

# 5. Паттерны разработки ПО

---

---

В этой главе мы рассмотрим сложные вопросы, которые выходят за рамки общих паттернов формирования проектной среды, но тем не менее являются именно паттернами этой серии, так как позволяют сформировать единое поле выполнения проекта в команде.

## 5.1. Паттерн «Итеративная разработка ПО»

Процесс итеративной разработки/создания продукта в настоящее время приобрел массу поклонников даже среди тех, кто не вполне понимает, в чем заключается суть итеративного подхода. Особенно популярно выражение «итеративная разработка» среди последователей Agile-подхода, что, несомненно, правильно, учитывая то, что этот подход изначально итеративен. Но все ли читатели понимают, что такое итерация? Итак...

Прежде чем вы продолжите чтение, ответьте сами себе на вопросы.

- ◆ Что такое итерация?
- ◆ Как можно определить границы итерации?
- ◆ Зачем вообще нужны границы итерации, ведь цели итерации можно обозначить и по вехам проекта?

Единственно правильный ответ на них и одновременно суть итеративного подхода состоит в том, что вы на  $(i + 1)$ -й итерации повторяете ту же деятельность, которой занимались на  $i$ -й итерации, только имея обновленный и обогащенный багаж знаний.

Если в вашем понимании первая итерация — это основная функциональность, вторая итерация — это уточнение вспомогательных/неосновных функций, а третья итерация — наведение красоты в интерфейсе, то вы с большой вероятностью не следуете принципу итеративной разработки ПО.

Принцип итеративной разработки означает, что на первой итерации вы реализуете часть основной функциональности, а на второй итерации в первую очередь пересматриваете то, что сделано на первой итерации, дорабатываете

основную функциональность и только потом переходите к вспомогательным/неосновным функциям. То есть каждая  $i + 1$  итерация — это в первую очередь уточнение наших знаний о системе на  $i$ -й итерации.



А отсюда вытекает основной принцип итеративной разработки: на ранних итерациях — «взгляд с высоты птичьего полета». То есть даже если вы знаете какие-то частности и детали уже на 1-й итерации, не нужно это знание детально описывать. Лучше смотреть на всю систему комплексно, шире, оставаясь на едином уровне детализации и знаний.

Мне очень нравится, как Джефф Паттон показывает принцип итеративной разработки в своей замечательной презентации User Story Mapping ([http://www.agileproductdesign.com/downloads/patton\\_user\\_story\\_mapping.ppt](http://www.agileproductdesign.com/downloads/patton_user_story_mapping.ppt)).

Очень часто приходится сталкиваться с тем, что в качестве итеративной разработки внедряется процесс, похожий рис. 5.1: команда долго, последовательно и мучительно «вылизывает» в мельчайших деталях отдельные фичи, при этом понимание того, что будет представлять собой решение в целом, появится очень нескоро. Следствием является то, что получение реального опыта эксплуатации системы откладывается до поздних этапов проекта. На ранних этапах система просто не будет предоставлять минимально необходимого набора функций.



Рис. 5.1. Неитеративная разработка

А вот другой подход к реализации продуктов, который, с моей точки зрения, и является итеративным (рис. 5.2).

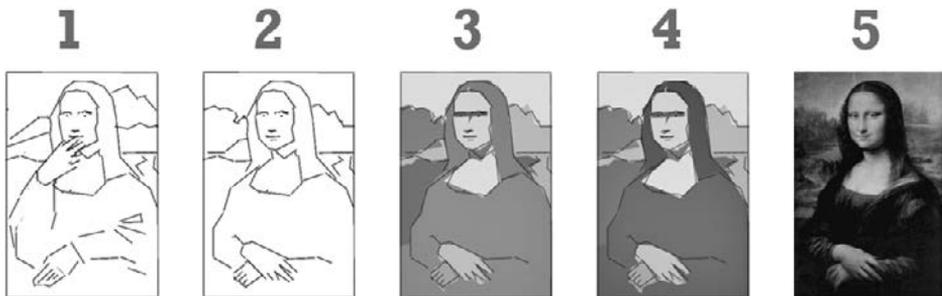


Рис. 5.2. Итеративная разработка

Смотрите, какая между ними разница: в первом варианте только на предпоследней итерации стала ясна художественная композиция — лицо, положение рук, фон. Во втором варианте мы получили общее представление о картине сразу, на второй итерации откорректировали положение рук, потом попробовали обозначить цветовую гамму решения, не понравилось — переделали на четвертой итерации, а на пятой довели решение до конца.

Очень хорошо описан итеративный подход к разработке в практике Story Mapping, изобретенной уже упомянутым Джефом Паттоном. Для полноценного ознакомления с практикой Story Mapping я рекомендую обратиться к первоисточнику — сайту <http://www.agileproductdesign.com>.



Здесь упомяну только предлагаемый способ разделения функционала на слои (помните, как в первой части «Шрека»: «У лука есть слои, у людоедов есть слои...»). Паттон также предлагает разделить функциональность продукта на слои.

Например, нам нужно реализовать сценарий ввода пользователем регистрационных данных: e-mail, пароль, контактный телефон, регион. На какие слои мы можем его разбить?

- ◆ Голая необходимость — минимальный функционал, реализующий нужную фичу. Например, простой набор текстовых полей без валидации данных, сложных элементов управления, реализации альтернативных сценариев и т. д.
- ◆ Вариативность — расширение функционала, добавление поддержки альтернативных сценариев.
- ◆ Достоверность данных — под достоверностью здесь подразумевается целостность, согласованность, непротиворечивость и т. д. Именно на этом слое появляются валидация и проверка корректности ввода, предотвращающие введение пользователями некорректных данных.
- ◆ Удобство, производительность и т. д. — сюда попадает все, что не относится к базовому скелету фичи, но помогает сделать ее более привлекательной для использования: увеличенная скорость работы, красивый интерфейс, всяческого рода автозаполнения данных, которые помогают облегчить жизнь пользователя.

Распределение работ в проекте, ведущемся в соответствии с итеративным подходом, может строиться следующим образом.

**Начало проекта.** Строится функциональный скелет продукта — основной функционал в простейшем варианте реализации. Например, для ввода данных могут применяться простые текстовые поля без валидации, форматирования и автоподстановки. Уже на этом этапе можно проводить тестирование, чтобы получить первые отклики пользователей и других заинтересованных лиц. Также в конце этого этапа можно оценивать производительность и масштабируемость решения.

**Середина проекта.** На этом этапе завершается работа над функционалом, а функционал, реализованный ранее, становится более удобным и полным.

**Конец проекта.** Продукт готовится к выпуску. Продолжается его проверка на реальных пользователей и получение обратной связи. Выполняется тестирование производительности и масштабируемости.



Умение отсекаать лишнее — одно из главных качеств аналитика, и в данном случае это качество должно стать первым качеством менеджера проекта. Тем более что это связано и со скоупом, и со стоимостью работ. Вы как менеджер должны уметь выдать «прототип MS Word» в таком виде, чтобы я как заказчик понял красоту заложенных концепций, но не погряз в вопросах, Ctrl+Z или Ctrl+Backspace мне надо нажимать, чтобы отменить последнее событие. Да и какая разница? На этой итерации достаточно, чтобы все представляли себе, что какая-то возможность будет концептуально. И это будет концептуальная итерация. Потому что уже буквально на следующей итерации я, как заказчик, могу сказать: «Нет, не покупаю такую фичу, дорого». И не будет больше такой функции. А значит, нет смысла прорабатывать детально блоки информации и строить диаграммы последовательности. Равно как и не стоит тратить 20 человеко-часов на то, чтобы запрограммировать для меня эту функцию, чтобы на очередном скрам-митинге я сказал, что это мне не нужно.

В этой книге я не буду проводить сравнительный анализ существующих методологий разработки ПО. Это сделано в первой книге. На основании выводов по каждой из методологий, представленных в книге [1], составлена сводная таблица применимости методологий в зависимости от параметров разрабатываемых системы или продукта, требований к разработке и рисков, связанных с разработкой (табл. 5.1). Эта таблица не претендует на сравнительный анализ методологий, ее цель — отобразить значимые, с точки зрения авторов, различия для выбора методологии разработки в качестве основы.

**Таблица 5.1.** Сводная таблица методологий разработки

Требования к разработке свойства продукта (системы). Начало проекта	XP	Scrum	ICONIX	RUP
1	2	3	4	5
Риск наличия профессиональной и сработавшейся команды	+			
Риск наличия профессиональной команды инициативных разработчиков	+	+		
Сложность системы:				
• Низкая	+	+	[+]	[+]
• Средняя	(+)	+	+	+
• Высокая		(+)	+	+
Размер системы:				
• Маленькая	+	+	[+]	[+]
• Средняя	(+)	+	+	+
• Большая		(+)	+	+

Продолжение ⇨

Таблица 5.1 (продолжение)

1	2	3	4	5
Длительность проекта:				
• Малая	+	+	[+]	[+]
• Средняя	(+)	+	+	[+]
• Большая			+	+
Сильная вовлеченность клиента в тестирование	+			
Сильная вовлеченность клиента в процесс разработки	+	+		
Сильная вовлеченность клиента в процесс разработки модели предметной области			+	[+]
Разработка, управляемая моделью			+	+
Разработка несколькими компаниями-разработчиками			(+)	+
Детальные спецификации на систему			(+)	+
Доработка продукта производится несколькими компаниями-разработчиками			(+)	+
Поддержка продукта сторонней компанией			(+)	+

Здесь использованы следующие обозначения: + – методология удовлетворяет требованиям/способна выполнять проекты с описанными свойствами/включает в себя риски; (+) – методология удовлетворяет требованиям при дополнительных действиях; [+] – методология удовлетворяет требованиям, но не всегда является оптимальной.

Из табл. 5.1 можно сделать следующие выводы.

Для маленьких по объему проектов с очевидной функциональностью, которые не будут постоянно дорабатываться или будут дорабатываться по минимуму одной и той же командой, имеет смысл рассматривать методологию XP.

Риски:

- ◆ отсутствие профессиональной и сработавшейся команды;
- ◆ отсутствие зафиксированных требований к системе;
- ◆ неоптимальный дизайн из-за отсутствия фазы системного анализа и дизайна системы.

Для менее очевидных по функциональности проектов и более долгосрочных, больших и сложных проектов, которые не будут постоянно дорабатываться или будут дорабатываться по минимуму одной и той же командой, имеет смысл рассматривать методологию Scrum.

Риски:

- ◆ отсутствие профессиональной команды инициативных разработчиков;
- ◆ отсутствие спецификаций требований к системе;
- ◆ риск постоянного расширения скоупа проекта и требований со стороны заказчика.

Для больших по объему, длительности сложных проектов имеет смысл рассматривать методологию ICONIX. В случае необходимости документирования спецификаций для обеспечения поддержки или разработки другими компаниями-разработчиками методология дополняется отдельными практиками или артефактами из методологии RUP.

Риски:

- ◆ отсутствие детальных спецификаций требований к системе;
- ◆ недостаточность квалификации системных аналитиков и архитектора.

Для разработки долгосрочных, сложных и больших по объему проектов, особенно в случае, если разработкой, доработкой и поддержкой продукта будут заниматься несколько компаний-разработчиков и если требуется наличие детальных спецификаций на систему, следует рассматривать методологию RUP.

Риски:

- ◆ большой объем документации;
- ◆ отсутствие рабочих процедур и инструкций, реализующих процессы RUP;
- ◆ недостаточность квалификации системных аналитиков и архитектора.

В качестве основы методологии разработки ПО предлагается выбрать методологию ICONIX как гибкую и масштабируемую методологию, с одной стороны, и как наиболее точно отвечающую требованию необходимого и достаточного для разработки качественного продукта минимума формальных документов и спецификаций — с другой.



Процесс ICONIX рассматривает разработку системы в динамике и статике, которые гибко используются в итеративном подходе: вы можете в рамках одной итерации пройти весь процесс ICONIX для небольшой партии прецедентов/вариантов использования, возможно даже для какой-то части (отдельных шагов сценария, например happy path) прецедентов, закончив итерацию исходным кодом и модульными тестами.

По этой причине процесс ICONIX хорошо подходит для экстремальных проектов, где крайне важна быстрая обратная связь в зависимости от таких факторов, как требования, проектирование и оценка полученного прототипа.

## 1. Требования.

- ◆ **Функциональные требования:** определите, что система должна делать. В зависимости от того, как организован ваш проект, или вы будете вовлечены в создание функциональных требований, или требования будут сформулированы в самых общих чертах клиентом или командой бизнес-аналитиков.
- ◆ **Моделирование предметной области:** описать и смоделировать предметную область в виде однозначно трактуемой модели.
- ◆ **Поведенческие требования:** определить, как будут взаимодействовать пользователь и система (сценарии использования). ICONIX рекомендует

начать с GUI-прототипа (раскадровка графического интерфейса) и выявить все случаи использования, которые вы собираетесь реализовать, или по крайней мере очертить скоуп прецедентов для первой итерации, который с меньшей вероятностью будет изменяться по мере того, как вы выявите и сформулируете требования с большей глубиной.

**Веха 1 — ревью требований.** Убедитесь, что вариант использования текста совпадает с ожиданиями клиента. Обратите внимание на то, что вы могли бы рассматривать варианты использования небольшими партиями непосредственно перед их разработкой. Тогда на каждой итерации (то есть для небольшой партии прецедентов) вы постепенно выполните действия, описанные далее.

## 2. Анализ/эскизный проект.

- ◆ Анализ надежности: нарисуйте диаграмму пригодности/робастности (модель объектов из шагов в прецеденте), при необходимости переписав текст варианта использования, если он «не укладывается» в диаграмму пригодности. Диаграмма пригодности — одна из ключевых особенностей ICONIX, позволяющая проверить качество и системность сформулированного прецедента. Вы просто не сможете построить диаграмму, если прецедент прописан некачественно. Я рекомендую использовать этот инструмент в критических или системных вариантах.
- ◆ Обновляйте модель предметной области по мере создания описания прецедентов и рисования диаграмм пригодности. Вы откроете для себя пропущенные классы, скорректируете описания, устраните двусмысленности, добавите атрибуты для объектов предметной области (например, определите, что объект «книга» имеет название, автора, синопсис и т. д.).
- ◆ Поименуйте все логические функции программного обеспечения (контроллеры), необходимые для того, чтобы реализовать прецеденты.
- ◆ Скорректируйте первые драфты вариантов использования.

**Веха 2 — ревью предварительного дизайна.** Убедитесь в том, что модель предметной области актуализирована, уточнена, не противоречит вариантам использования и полна с точки зрения диаграмм пригодности вариантов использования.

## 3. Рабочий проект.

- ◆ Нарисуйте диаграмму последовательности (одну для прецедента), чтобы показать в деталях, как вы собираетесь реализовать вариант использования. Основная цель этих диаграмм — разбиение поведения системы по классам.
- ◆ Обновляйте модель предметной области по мере разработки диаграмм последовательности, а также добавьте выявленные методы к объектам домена. На этом этапе объекты домена — действительно классы домена, или сущности, и эта модель предметной области быстро становится статической моделью или диаграммой классов — важной частью вашего рабочего проекта.

- ◆ Очистите статическую модель. Поименуйте более корректно атрибуты и методы, исправьте методы и их параметры с точки зрения реализации системы. Уточните кратность связей между классами.

**Вежа 3 – критическое Design Review (CDR).** Анализ соответствия динамической составляющей системы (диаграмм последовательности) статической составляющей (диаграмме классов), анализ и корректировка распределения поведения системы по классам с точки зрения хороших практик реализации кода.

#### 4. Внедрение.

- ◆ Кодирование/модульное тестирование: написать код и модульные тесты. Или в зависимости от ваших предпочтений написать модульные тесты, а затем код.
- ◆ Интеграция и сценарий тестирования: интеграционные тесты на базе прецедентов, покрывающие как основной поток (happy path), так и альтернативный поток сценариев.

**Вежа 4 – ревью кода и обновление модели.** Для подготовки к следующей итерации синхронизируйте модель и код. Удалите методы и атрибуты, которых в коде нет, или исправьте код.

**Вежа 5 – уточнение требований.** Проведите расширенное интервью по ранее неглубоко выявленным моментам пользовательских историй/раскадровки графического интерфейса, покажите прототип системы как можно раньше, соберите замечания и уточнения.

Выполните новую итерацию, начиная с шага 1.

Жизненный цикл разработки программного обеспечения по ICONIX приведен на рис. 5.3.

Методология должна учитывать хорошо зарекомендовавшую себя практику Agile, предусматривающую итеративность выпуска прототипов продукта с целью минимизации рисков при выяснении требований к продукту и управлению ими, а также вовлечения заказчика в процесс разработки на ранних стадиях. Методология должна предусматривать возможность расширения процессными практиками и артефактами RUP для формирования спецификаций на систему требуемого уровня детализации.

Как один из вариантов, который реально работает, с целью минимизации рисков при выявлении новых требований на поздних стадиях разработки, минимизации рисков по «плавающим» требованиям, а также с целью вовлечения заказчика в процесс разработки на ранних стадиях имеет смысл использовать Agile-подход к итеративности разработки, взяв за основу методологию ICONIX.

Принцип итеративности базируется на законе Парето: с одной стороны, 20 % функционала покрывают 80 % основных потребностей заказчика, с другой — 20 % функционала (критические use cases) производят 80 % ошибок.

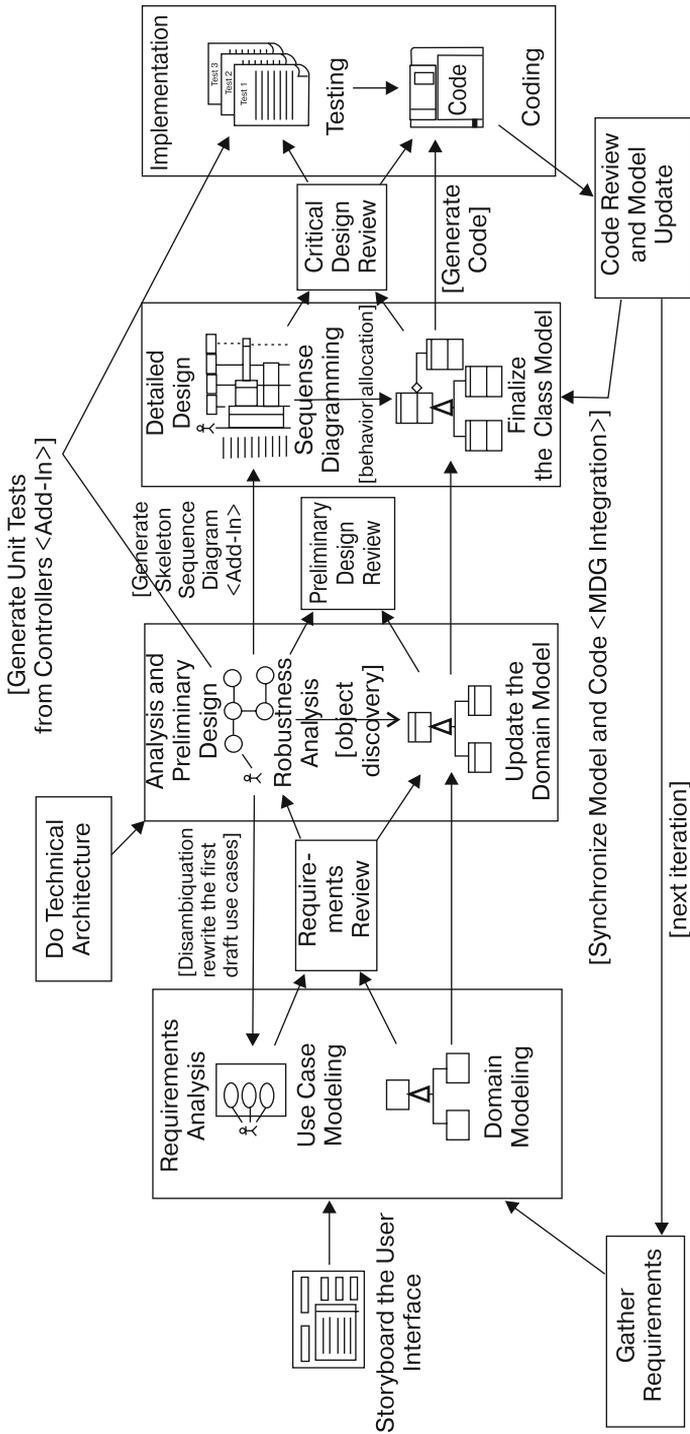


Рис. 5.3. Жизненный цикл ICONIX

Итеративность оперирует понятием «прототип». Так называют условно рабочую версию системы с заведомо неполно реализованным функционалом и заведомо большим количеством ошибок. Прототип поставляют заказчику, последний исследует его на предмет анализа собственных ожиданий от системы, правильности общего направления разработки продукта, корректности интерфейсных взаимодействий и удобства использования будущей системы. При анализе прототипа для достижения указанных целей ошибки реализации игнорируются как неважные.

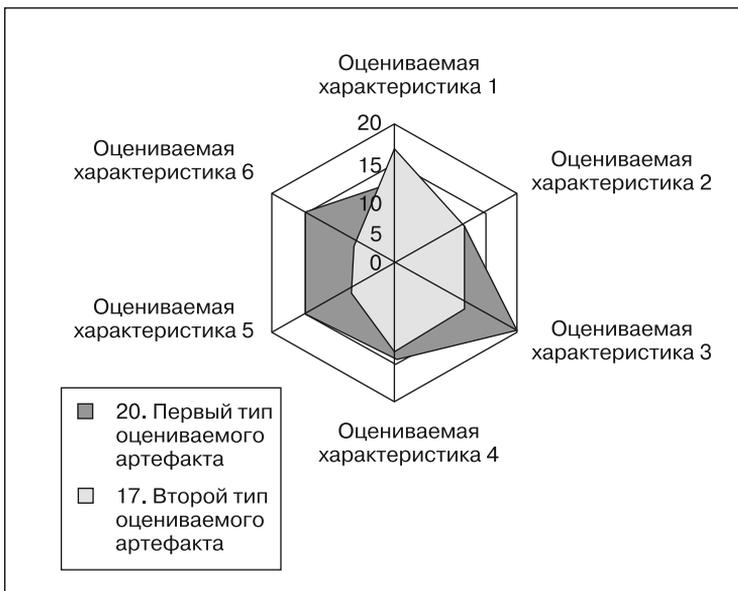
Процесс разработки ПО можно рассматривать как процесс узнавания и моделирования (программной реализации) предметной области, начиная с точки абсолютного незнания и заканчивая полным знанием всей предметной области.

В качестве иллюстрации меры знания о предметной области используются радарные диаграммы, которые строятся на основании табл. 5.2.

**Таблица 5.2.** Радарная диаграмма меры знания о предметной области

Количество	Оцениваемый артефакт	Оцениваемая характеристика					
		1	2	3	4	5	6
20	Первый тип	12	12	20	14	15	15
17	Второй тип	17	12	12	12	7	6

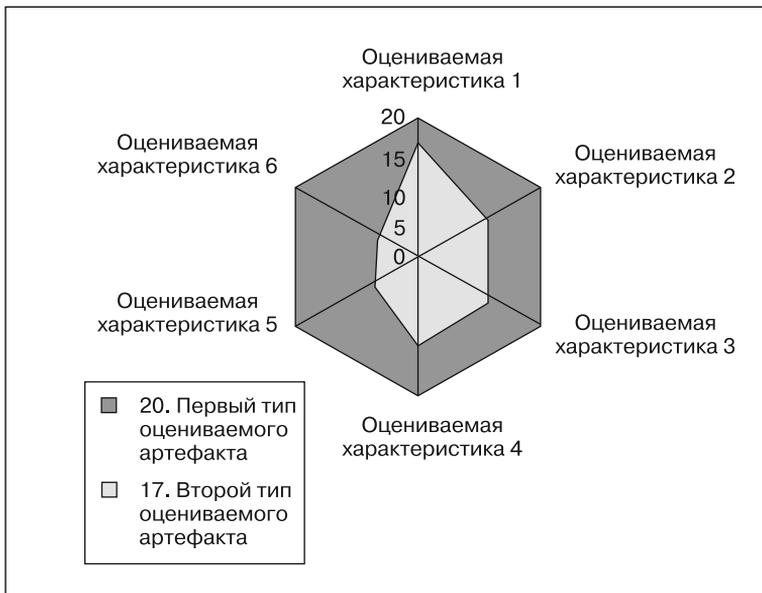
Радарная диаграмма отображает количественные оценки для каждой из оцениваемых характеристик и каждого типа оцениваемого артефакта в виде, показанном на рис. 5.4.



**Рис. 5.4.** Радарная диаграмма. Пример

На диаграмме видно покрытие для каждого типа оцениваемого артефакта, представляющее собой плоскость, окрашенную в цвет, соответствующий типу артефакта, и проходящую через конкретные количественные оценки каждой из оцениваемых характеристик.

С помощью этой диаграммы можно увидеть, какие характеристики, для каких типов артефактов необходимо уточнять. Конечная цель — покрытие всей области знаний плоскостями. Например, из диаграммы, представленной на рис. 5.5, можно сделать вывод, что мы знаем все о первом типе оцениваемого артефакта и примерно 40 % — о втором типе.



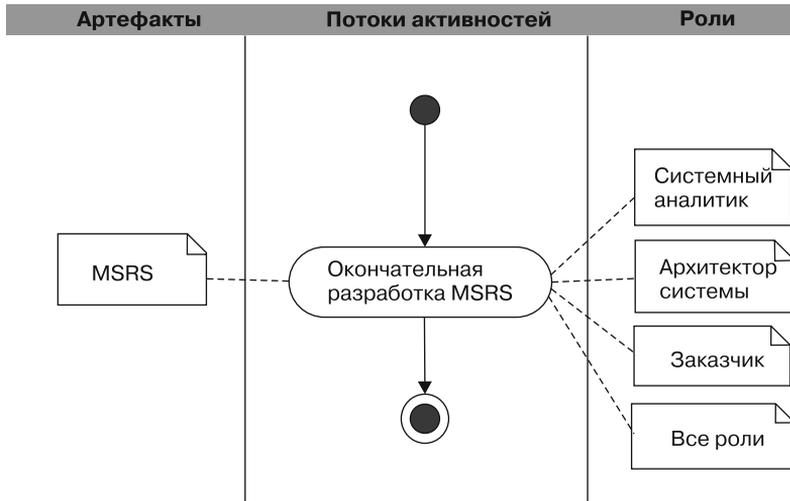
**Рис. 5.5.** Радарная диаграмма

Радарные диаграммы, использованные в качестве иллюстраций в разделах, описывающих итерации, отображают представление двух типов артефактов: критических use cases и обычных use cases, исходя из допущения, что в системе всего 100 use cases, из них критических — 20, обычных — 80.

Итерации иллюстрируются диаграммами активностей, которые представляют собой отображение активностей и их взаимосвязей, а также список артефактов, участвующих в каждой из активностей, и список ролей, выполняющие активность. Каждая диаграмма состоит из трех уровней: «Артефакты», «Поток активностей» и «Роли», и на каждом уровне отображаются соответствующие компоненты (рис. 5.6).



Все представленные диаграммы активностей являются первыми приближениями, цель которых — помочь понять процессы предлагаемой методологии, и не прорабатывались детально.



**Рис. 5.6.** Нотация диаграмм активностей для описания бизнес-процессов ЖЦ разработки

Предлагается выполнять итерации, которые будут рассмотрены далее.

### 5.1.1. Нулевая итерация (предварительный анализ)

#### Описание итерации

- ◆ Выяснить, каковы свойства, характеристики и функциональность системы и эксперты предметной области, методом интервьюирования заказчика продукта, который должен ответить на вопрос «Чем конкретно я доволен при использовании этой системы?» с позиции разных ролей, использующих создаваемую систему, в предположении, что система уже готова и все роли активно ее используют. Здесь же выполняется (если это не было сделано раньше) краткий бизнес-анализ — исследование процессов и операций, краткое их описание, уточнение критических факторов успеха и метрик процессов, показателей эффективности выполнения процессов и достижения целей. Ответы приоритизируются с ТЗ важности совместно с заказчиком (1-2 дня).
- ◆ Дать грубые оценки трудоемкости (0,5 дня). Эти оценки используются при управлении скоупом и стоимостью проекта, переговорах с заинтересованными лицами о том, нужна ли такая функциональность за такую цену в принципе. Здесь не важна точность оценок, важно то, что все они делаются с примерно одинаковой точностью и появляется возможность сравнить функциональность по «весу цены» в единицах измерения «трудоемкость».

- ◆ Выделение use cases и построить use case-модели (0,5–1 день).
- ◆ Совместно с заказчиком приоритизировать use cases и назначить срочность для каждого use case (1 день). На практике это неплохо работает для подразделений внутренней автоматизации, хуже – в случае крупных заказных разработок и разработок продуктов. Зачастую заказчики «от бизнеса» могут определить лишь минимальные рамки релизов, но не более того. Приоритизацией внутри релизов они не занимаются: «Нам нужно это все, в каком порядке – неважно». Так было, к сожалению, на всех моих местах работы и проектах. Исключение – ситуации, когда заказчик сам вышел из разработчиков. Борьба с такой ситуацией можно путем прессинга заказчика, выдавливания из него частоты использования сценариев, а также согласования с заинтересованными лицами плана демонстрации прототипов системы.
- ◆ Проанализировать функциональность и характеристики системы.
  - ◆ Описать основной поток всех критических use cases (0,5 дня). Чтобы уложиться в этот срок, сначала на одинаково высоком уровне описывают основные потоки, затем подробнее описывают критические из критических, если остается время, описывают остальные. На этой итерации важна не глубина и точность описания, а согласованность понимания каких-то основных моментов, пусть и прописанных концептуально и пунктирно.
  - ◆ Проанализировать первые 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, – определение и краткое описание основного пути выполнения прецедента, а также формирование списка пожеланий к use case с точки зрения каждой роли (максимум 1 день).
  - ◆ Проанализировать следующие 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, чтобы сформировать только список пожеланий к use case с точки зрения каждой роли (максимум 0,25 дня).
  - ◆ Проанализировать оставшиеся 60 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, с целью выявления списка характеристик системы и экспертов предметной области (максимум 0,125 дня).
- ◆ Ознакомить архитектора системы и разработчика с выработанными материалами с целью формирования списка вопросов для обсуждения с заказчиком (на все эти действия должно уйти не больше семи рабочих дней).
- ◆ Обсудить вопросы с заказчиком.
- ◆ Дать уточненные оценки трудоемкости (1 день).

Диаграмма активности нулевой итерации приведена на рис. 5.7.

Радарная диаграмма покрытия предметной области знаниями о ней после нулевой итерации изображена на рис. 5.8.

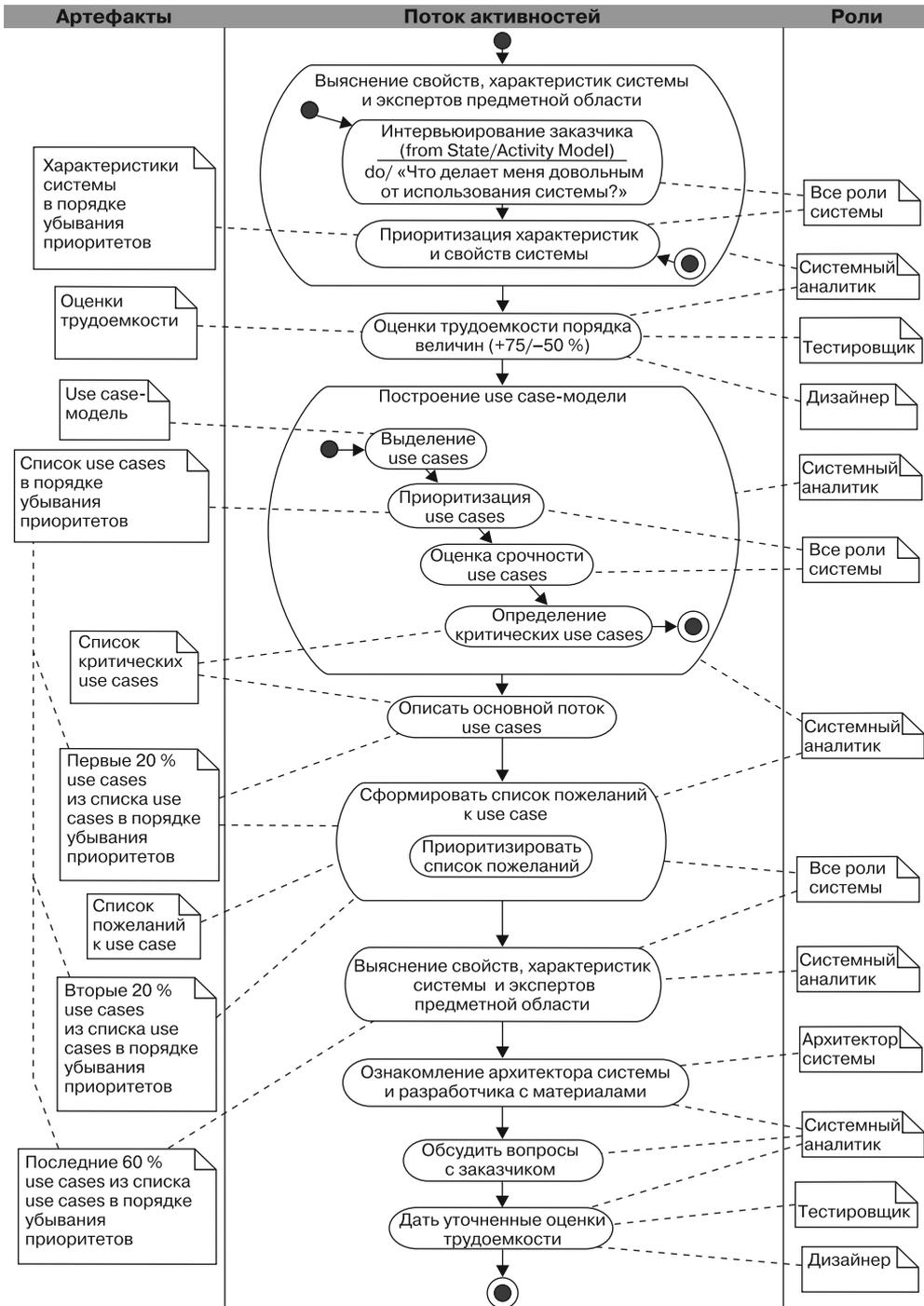


Рис. 5.7. Диаграмма активности нулевой итерации

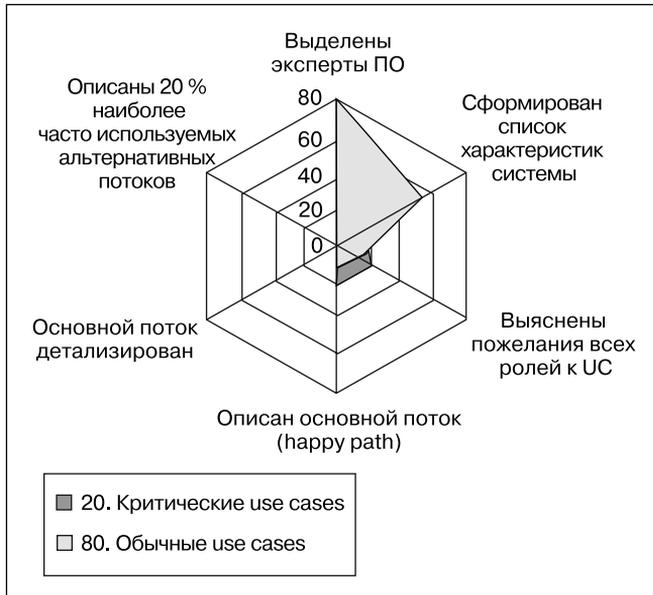


Рис. 5.8. Покрытие предметной области знаниями после нулевой итерации

## 5.1.2. Первая итерация

### Описание итерации

- ◆ Проанализировать функциональность и характеристики системы.
  - ◆ По результатам обсуждения с заказчиком обновить use case-модель и детализировать описания основного и альтернативных потоков выполнения всех критических use cases — use cases, без которых система не может существовать. Альтернативные потоки должны быть перечислены все, детализируются только первые 20 % альтернативных потоков из списка таких потоков, отсортированного в порядке убывания частоты использования (1–3 дня).
  - ◆ Детализировать описания основных потоков первых 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов. Сформировать список пожеланий к use case с точки зрения каждой роли (1–2 дня).
  - ◆ Проанализировать все оставшиеся use cases с целью выявления списка характеристик системы и экспертов предметной области (максимум 0,125 дня).
- ◆ Поставить первый прототип, реализующий:
  - ◆ первые 20 % ожиданий заказчика из списка ответов на вопрос «Чем конкретно я доволен при использовании этой системы?», отсортированных по убыванию важности, без глубокой детализации;

- ◆ основные потоки всех критических use cases;
  - ◆ первичную концептуальную модель системы;
  - ◆ первичную модель архитектуры;
  - ◆ первичный дизайн;
  - ◆ первичные прототипы интерфейсов.
- ◆ Презентовать прототип заказчику.
  - ◆ Собрать и проанализировать замечания к системе, уточнить старые и выявить новые ожидания от системы и требования к ней.
  - ◆ Дать уточненные оценки трудоемкости.

Диаграмма активности первой итерации приведена на рис. 5.9.

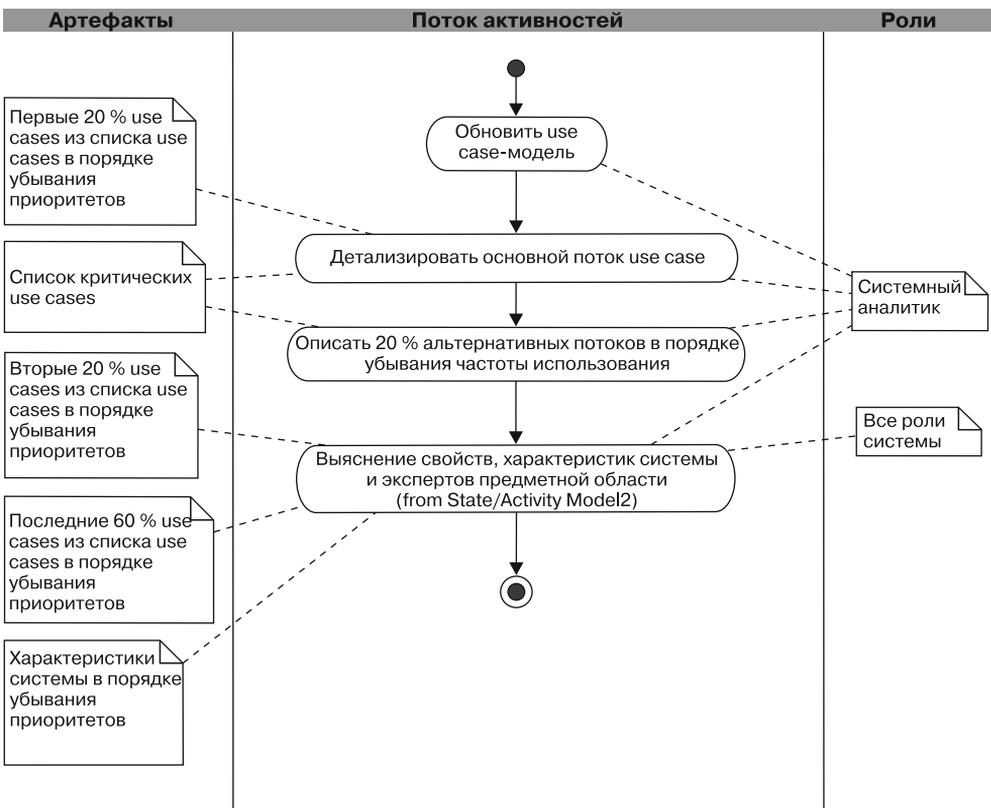


Рис. 5.9. Диаграмма активности первой итерации

Радарная диаграмма покрытия предметной области знаниями о ней после первой итерации изображена на рис. 5.10.

Следует обратить внимание на то, что после первой итерации покрытие для критических use cases полное, так как их всего 20. Иначе говоря, мы знаем все о критических use cases.

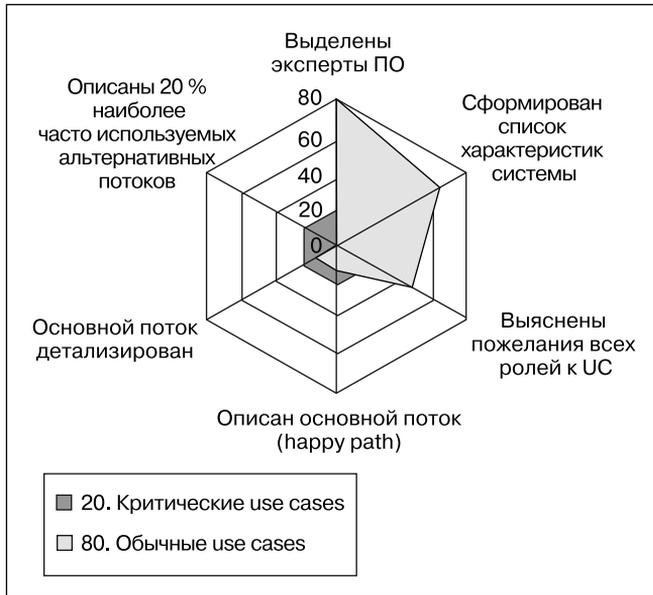


Рис. 5.10. Покрытие предметной области знаниями после первой итерации

## 5.1.3. Вторая итерация

### Описание итерации

- ◆ Проанализировать функциональность и характеристики системы.
  - ◆ По результатам обсуждения с заказчиком обновить use case-модель и детализировать описания основного потока для следующих 20 % из списка use cases, отсортированного в порядке убывания приоритетов, и создать описание альтернативных потоков выполнения 40 % use cases. Альтернативные потоки должны быть перечислены все, детализируются только первые 20 % альтернативных потоков из списка таких потоков, отсортированного в порядке убывания частоты использования (1–3 дня).
  - ◆ Проанализировать остальные use cases и сформировать список пожеланий к ним с точки зрения каждой роли (2–5 дней).
- ◆ Поставить второй прототип, реализующий:
  - ◆ следующие 20 % ожиданий заказчика из списка ответов на вопрос «Чем конкретно я доволен при использовании этой системы?», отсортированных по убыванию важности, без глубокой детализации;
  - ◆ альтернативные потоки всех критических use cases;
  - ◆ основные потоки первых 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов;
  - ◆ уточненную концептуальную модель системы;

- ◆ уточненную модель архитектуры;
- ◆ уточненный дизайн;
- ◆ уточненные прототипы интерфейсов.
- ◆ Презентовать прототип заказчику.
- ◆ Собрать и проанализировать замечания к системе, уточнить старые и выявить новые ожидания от системы и требования к ней.
- ◆ Дать уточненные оценки трудоемкости.

Диаграмма активности второй итерации приведена на рис. 5.11.

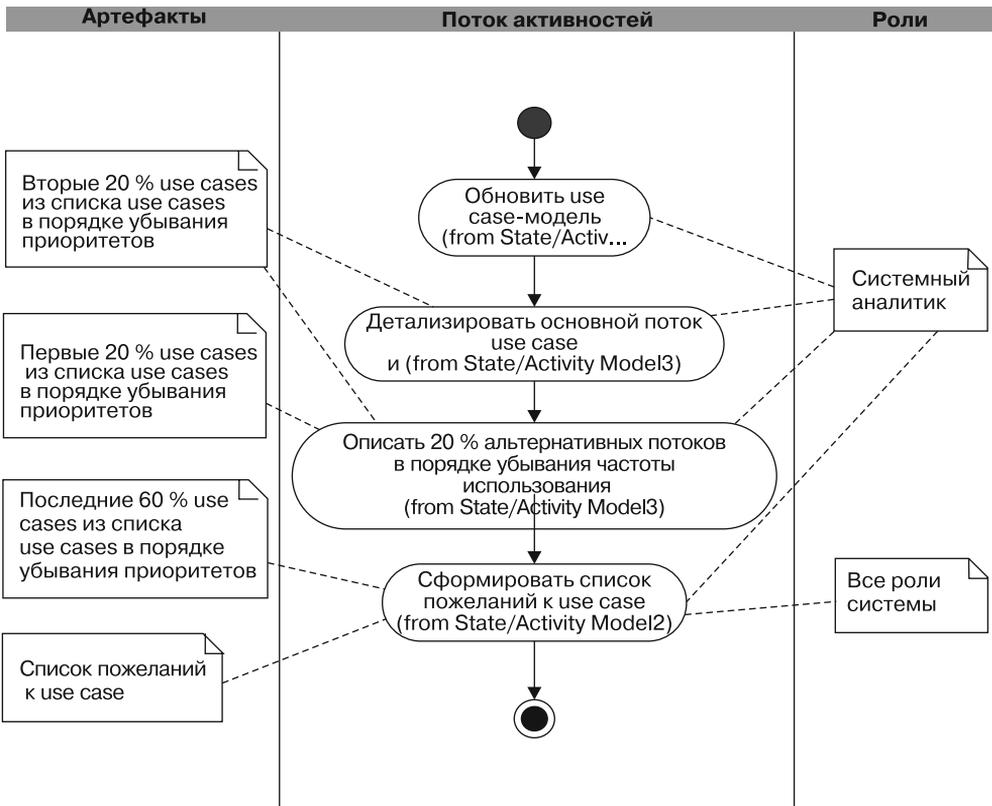


Рис. 5.11. Диаграмма активности второй итерации

Радарная диаграмма покрытия предметной области знаниями о ней после второй итерации изображена на рис. 5.12.

Покрытие критических use cases находится «под» покрытием обычных use cases — мы узнаем о предметной области больше и больше, поэтому покрытие знаниями об обычных use cases количественно перекрывает наши знания о критических use cases.

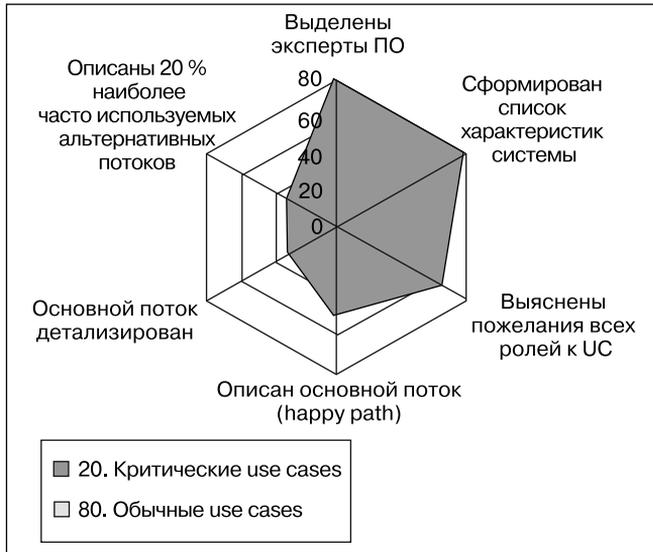


Рис. 5.12. Покрытие предметной области знаниями после второй итерации

## 5.1.4. Третья итерация

### Описание итерации

- ◆ Создать первую версию пакета спецификаций требований, включающего:
  - ◆ обзор use case-модели;
  - ◆ детальные use case-спецификации;
  - ◆ все требования к системе: дополнительные функциональные требования; нефункциональные требования; требования к архитектуре; требования к дизайну; требования к реализации;
  - ◆ прототипы интерфейсов;
  - ◆ концептуальную модель предметной области/системы.
- ◆ Выполнить детальный дизайн системы — создать логическую модель системы.
- ◆ Разработать и выполнить JUnit-тестирование программного кода.
- ◆ Поставить третий прототип, реализующий:
  - ◆ все критические use cases;
  - ◆ 40 % обычных use cases (при покрытии знаниями 60–80 % всех use cases);
  - ◆ 40 % ожиданий заказчика от всего функционала системы.
- ◆ Собрать и проанализировать замечания к системе, уточнить старые и выявить новые ожидания от системы и требования к ней.
- ◆ Дать точные оценки трудоемкости.

Диаграмма активности третьей итерации приведена на рис. 5.13.

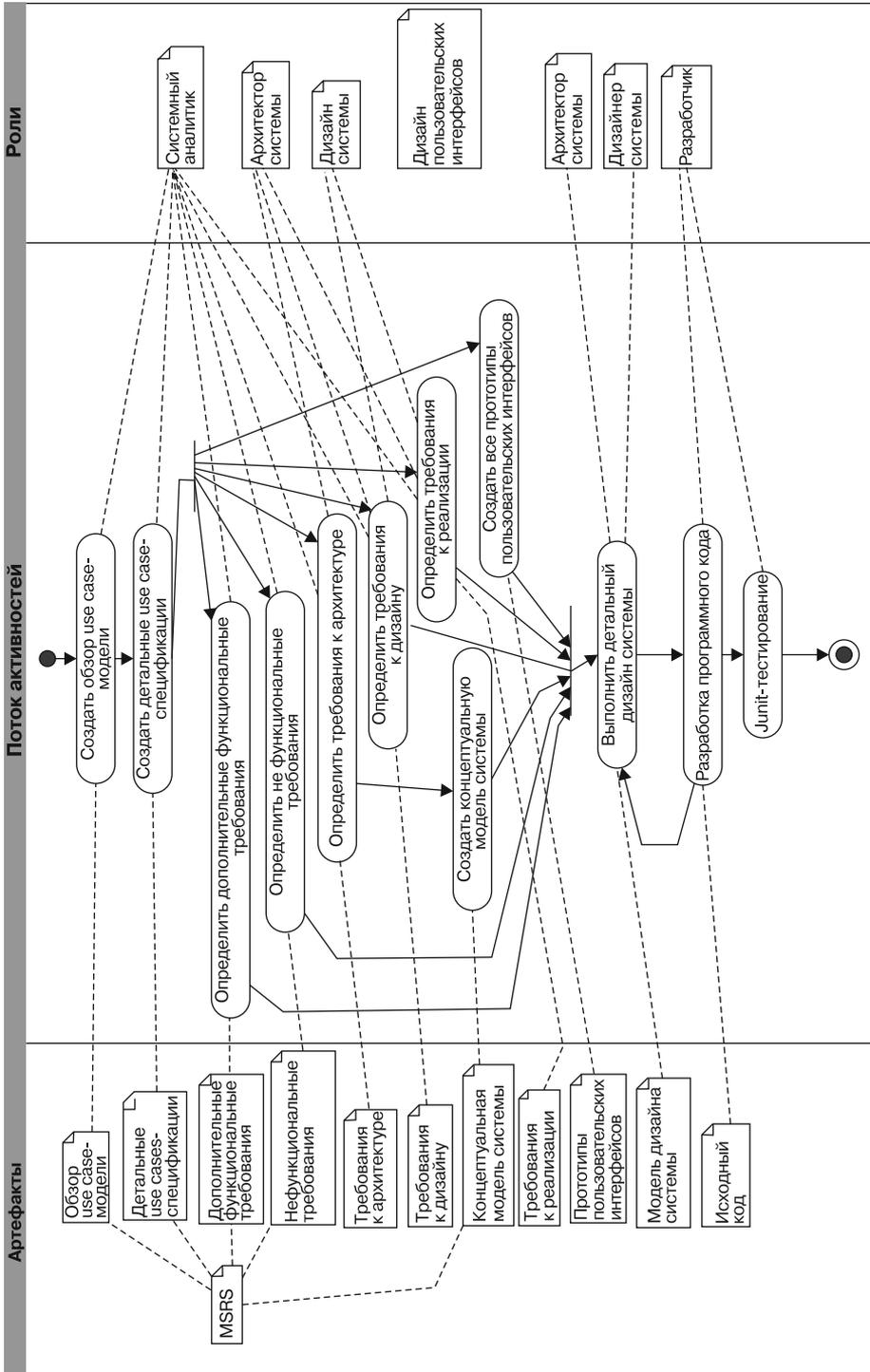


Рис. 5.13. Диаграмма активности третьей итерации

Радарная диаграмма покрытия предметной области знаниями о ней после третьей итерации изображена на рис. 5.14.

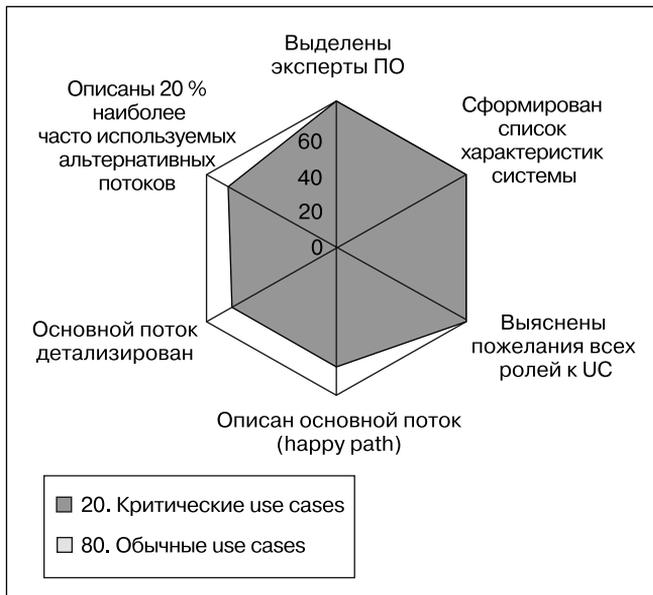


Рис. 5.14. Покрытие предметной области знаниями после третьей итерации

## 5.1.5. Четвертая итерация

**Описание итерации.** Окончательное уточнение и утверждение заказчиком полного пакета спецификаций требований (MSRS).

На данном этапе итеративность заключается в последовательном уточнении оставшейся функциональности по тому же принципу – в выделении блоков информации (use cases), каждый из которых составляет 20 % от оставшейся для анализа информации. Параллельно идет разработка, и по мере уточнения оставшихся use cases стартуют новые потоки работ по разработке.

Эта итерация выполняется циклически до 100%-ного покрытия знаниями всей предметной области.

Диаграмма активностей четвертой итерации приведена на рис. 5.15.

Радарная диаграмма покрытия предметной области знаниями о ней после четвертой итерации изображена на рис. 5.16.

Необходимо отметить, что реальное покрытие предметной области знаниями шире, чем представленное на диаграмме, так как в этой итерации выделены дополнительные функциональные требования, нефункциональные требования и т. д. Для упрощения восприятия эти моменты не отражены на диаграмме.

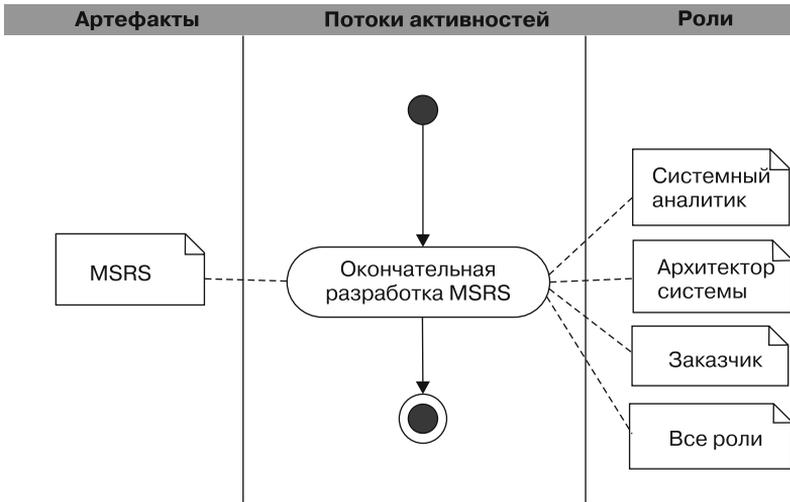


Рис. 5.15. Диаграмма активности четвертой итерации

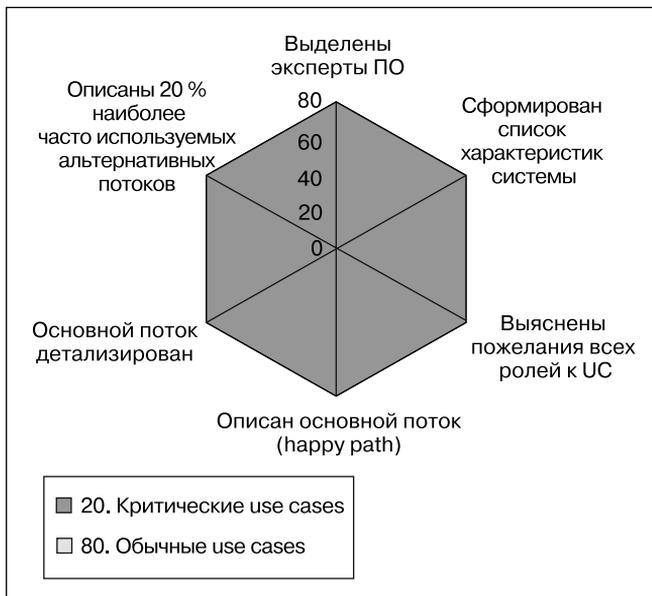


Рис. 5.16. Покрытие предметной области знаниями после четвертой итерации

### 5.1.6. Сводная таблица итераций

В табл. 5.3 представлена краткая информация о сути каждой итерации с иллюстрацией в виде радарной диаграммы.

Таблица 5.3. Резюме паттерна «Итеративная разработка ПО»

Основная работа в итерации	Иллюстрация
<p><b>1</b></p> <p>Основной поток всех критических use cases (0,5 дня).                      Анализ первых 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, — определение и краткое описание основного пути выполнения прецедента, а также формирование списка пожеланий к use case с точки зрения каждой роли (максимум 1 день).                      Анализ следующих 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, с целью сформировать только список пожеланий к use case с точки зрения каждой роли (максимум 0,25 дня).                      Анализ оставшихся 60 % use cases из списка use cases, отсортированного в порядке убывания приоритетов, с целью выявления списка характеристик системы и экспертов предметной области (максимум 0,125 дня)</p>	<p><b>2</b></p> <p>Выделены эксперты ПО</p> <p>Описаны 20 % наиболее часто используемых альтернативных потоков</p> <p>Сформирован список характеристик системы</p> <p>Основной поток детализирован</p> <p>Выявлены пожелания всех ролей к UC</p> <p>Описан основной поток (happy path)</p> <p>■ 20. Критические use cases                      □ 80. Обычные use cases</p>
<p>По результатам обсуждения с заказчиком обновить use case-модель и детализировать описания основного и альтернативных потоков выполнения всех критических use cases, без которых система не может существовать. Альтернативные потоки должны быть перечислены все, детализируются только первые 20 % альтернативных потоков из списка альтернативных потоков, отсортированного в порядке убывания частоты использования (1–3 дня).                      Детализировать описания основных потоков первых 20 % use cases из списка use cases, отсортированного в порядке убывания приоритетов. Сформировать список пожеланий к use case с точки зрения каждой роли (1–2 дня).</p>	<p>Выделены эксперты ПО</p> <p>Описаны 20 % наиболее часто используемых альтернативных потоков</p> <p>Сформирован список характеристик системы</p> <p>Основной поток детализирован</p> <p>Выявлены пожелания всех ролей к UC</p> <p>Описан основной поток (happy path)</p> <p>■ 20. Критические use cases                      □ 80. Обычные use cases</p>

1	2
<p>Анализ оставшихся use cases с целью выявления списка характеристик системы и экспертов предметной области (максимум 0,125 дня)</p> <p>По результатам обсуждения с заказчиком обновить use case-модель и детализировать описание основного пути для следующих 20 % из списка use cases, отсортированного в порядке убывания приоритетов, и создать описание альтернативных потоков выполнения 40 % use cases. Альтернативные потоки должны быть перечислены все, детализируются только первые 20 % альтернативных потоков из списка альтернативных потоков, отсортированного в порядке убывания частоты использования (1-3 дня). Проанализировать остальные use cases и сформировать список пожеланий к ним с точки зрения каждой роли (2-5 дней)</p>	<p>Выделены эксперты ПО</p> <p>Описаны 20 % наиболее часто используемых альтернативных потоков</p> <p>Сформирован список характеристик системы</p> <p>Выявлены пожелания всех ролей к UC</p> <p>Основной поток детализирован</p> <p>Описан основной поток (happy path)</p> <p>■ 20. Критические use cases □ 80. Обычные use cases</p>
<p>Создание первой версии пакета спецификаций требований (MSRS), включающего в себя:</p> <ul style="list-style-type: none"> <li>• обзор use case-модели;</li> <li>• детальные use case-спецификации;</li> <li>• все требования к системе;</li> <li>• дополнительные функциональные требования;</li> <li>• нефункциональные требования;</li> <li>• требования к архитектуре;</li> <li>• требования к дизайну;</li> <li>• требования к реализации;</li> <li>• прототипы интерфейсов;</li> <li>• концептуальную модель предметной области/ системы;</li> <li>• детальный дизайн системы;</li> <li>• логическую модель системы</li> </ul>	<p>Выделены эксперты ПО</p> <p>Описаны 20 % наиболее часто используемых альтернативных потоков</p> <p>Сформирован список характеристик системы</p> <p>Выявлены пожелания всех ролей к UC</p> <p>Основной поток детализирован</p> <p>Описан основной поток (happy path)</p> <p>■ 20. Критические use cases □ 80. Обычные use cases</p>

Таблица 5.3 (продолжение)

1	2
<p>Окончательное уточнение и утверждение заказчиком полного пакета спецификаций требований (MSRS).</p> <p>На данном этапе итеративность заключается в последовательном уточнении оставшейся функциональности по тому же принципу — выделению блоков информации, каждый из которых составляет 20 % от оставшейся для анализа информации. Параллельно идет разработка, и по мере уточнения оставшихся use cases начинают новые потоки работ по разработке</p>	<p>Выделены эксперты ПО</p> <p>Описаны 20 % наиболее часто используемых альтернативных потоков</p> <p>Сформирован список характеристик системы</p> <p>Выявлены пожелания всех ролей к UC</p> <p>Основной поток детализирован</p> <p>Описан основной поток (happy path)</p> <p>■ 20. Критические use cases □ 80. Обычные use cases</p>

## 5.1.7. Дальнейшая разработка

Дальше по необходимости выполняются последовательные действия по разработке: окончательный дизайн системы, разработка программного кода, тестирование (Unit-тестирование, функциональное тестирование, интеграционное тестирование, нагрузочное тестирование и т. д.), устранение дефектов, выявленных при тестировании, приемочные испытания, устранение дефектов, выявленных во время приемочных испытаний, и подписание акта сдачи продукта/услуги.

## 5.1.8. Резюме паттерна

Вы рассмотрели один из вариантов итеративной разработки ПО. В чем и до какой степени его адаптировать, зависит от вашего опыта, среды проекта и целей. Главное, что вы должны понять, прочитав эту главу, — это суть итераций не как последовательность действий, ориентированных на достигнутый ранее результат, и не как последовательность действий, корректирующих результаты предыдущей итерации, но как последовательность действий, детализирующих и последовательно развивающих результаты предыдущей итерации.

## 5.2. Паттерн «Релизное управление»

Этот паттерн позволяет эффективно управлять границами проектных работ, договариваясь с бизнесом о том, какие конкретно ожидания в виде законченных функциональных блоков системы будут реализованы в какой последовательности, с каким качеством и в какие сроки.

В моей практике были случаи, когда бизнес-дивизион и производственный департамент не могли эффективно управлять развитием продукта по одной простой причине — у них были совершенно разные взгляды на продукт, которые они формулировали каждый в своих терминах.

Действительно, бизнес интересуется набором некоторых полезных функций продукта, который он может продать своим клиентам и который позволит клиентам достигать своих целей с использованием продукта. Производство же интересуется функциональностью системы, которую они должны выпустить к определенному сроку с требуемым качеством. Основная причина непонимания в том, что функции, интересующие бизнес, и функции системы в понятийном поле, в котором мыслит производство, — разные вещи.

Более того, в конкретном кейсе, о котором я говорю, была заведена практика управления производством через задачи — использовался паттерн, похожий на паттерн WAS. Беда в том, что задачи ставились аналитикам, разработчикам и тестировщикам без какой-либо объединительной сущности, хотя задачи естественно были связаны между собой и ссылались друг на друга. В итоге даже на простые вопросы: «А какие конкретно функции системы мы реализуем к концу этого месяца?» — не было ответов. Крайне непросто было понять, какой объем задач будет выполнен не только аналитиками, но и разработчиками и тестировщиками. Хотя бы потому, что задач разработчиков, не говоря уже о задачах тестировщиков, зачастую попросту еще не было в системе управления задачами в тот промежуток времени, когда аналитики работали над своими задачами. Естественно, что в таких условиях бизнес совершенно не понимал, что делает производство, а производство совершенно не понимало причин, по которым бизнес не может разобраться в том, что же они делают.



---

В подобных ситуациях очень хорошо работает паттерн «Релизное управление».

Для начала следует ввести в словарь и разграничить сущности и понятия «бизнес-функция», «функция системы» и «задача». Бизнес-функция — это та самая «неотвратимая польза», которую принесет продукт конечному заказчику и которую продает ему бизнес. Каждая бизнес-функция может реализовываться одной или несколькими функциями системы. При этом каждая функция может быть реализована одной или несколькими более мелкими функциями. Для реализации одной функции нужно выполнить одну или несколько задач. В то же время одна и та же задача может быть необходимым условием для реализации нескольких функций.

---

Безусловно, если бы мои коллеги владели понятийной базой методологий разработки ПО, то все перечисленное можно было бы описать по аналогии, возможно, более привычной вам: business use case — feature — use case. В книге «Путь аналитика» я писал о том, что первично — курица или яйцо, то есть feature (фича) или use case (блок информации). Не буду здесь на этом вопросе останавливаться подробно. Просто давайте примем как данность, что сначала формулируется некая фича, а потом она декомпозируется на описание поведения системы. При этом один use case (одна какая-то часть описания поведения системы) может «покрывать» несколько фич.

Итак, в качестве понятной для обсуждения с двух сторон сущности мы выбрали фичу. Было не очень просто найти грань, до достижения которой бизнес еще понимал, о чем говорят эти «люди с планеты Железяка», и не терял фокуса достижения бизнес-целей, и на которой производство уже начинало понимать, что же потребуется сделать в системе, не углубляясь при этом в детали настолько, чтобы потерять из виду бизнес-назначение фичи и сосредоточенность на целях.

После этого мы ввели понятие релизного планирования. То есть ведущий аналитик постарался сформулировать все фичи «на грани», о которой я только что рассказал, после чего бизнес совместно с архитектором и ведущим аналитиком присвоили им приоритеты с точки зрения очередности и срочности реализации, а также важности для клиентов и для стабилизации/улучшения архитектуры приложения.



---

Затем совет по продуктам компании определил цели и примерные сроки выпуска ближайших релизов продукта. Обратите внимание: релизом здесь я называю не промежуточный выпуск, а конечную (на текущий момент) проверенную и стабильную версию продукта. Во многих компаниях это называют версией, а релизом — что-то близкое к внутренним итерациям в процессе производства этой версии. Это непринципиально и неважно. Нужно только договориться о терминах. Давайте также примем как данность, что в контексте этого паттерна релиз — это совокупность фич, которые направлены на удовлетворение заявленного перечня бизнес-функций и ожиданий клиента и заинтересованных лиц.

---

Далее фичи сформированных релизов поступали на оценку. Аналитики, разработчики и тестировщики вынуждены совместно анализировать свои трудозатраты, в каком порядке эти фичи лучше реализовывать и какие риски могут возникнуть по каждому из направлений, чтобы ответить на вопрос: «Сколько будет стоить реализация фичи?» Таким образом появлялась релизная оценка с перечнем допущений, ограничений и рисков.

*В начале использования этого паттерна коллеги часто спрашивали меня: «Но ведь оценка здесь делается с большой погрешностью, так как ни системный анализ, ни детальная проработка требований не проводились. Какой смысл в такой оценке?»*

*Ответ на данный вопрос уже дан ранее: да, эти оценки порядка величин, то есть +75 %... -50 %. Но все — все без исключения — фичи имеют ровно такую же погреш-*

*ность. А значит, с точки зрения понимания того, «дорого» или «дешево» и так ли это «срочно», если так «дорого», цель достигается.*

Данная релизная оценка поступала на совет компании по продуктам, который оценивал, нужна ли действительно такая «дорогая» (по трудозатратам) фича именно в этом релизе, или, может быть, вместо нее использовать в нем, скажем, вот эти три менее дорогие и чуть более низкоприоритетные фичи.

Если такое решение принималось, то релизы переформировывались. Иногда снова требовалась переоценка. В итоге получался релизный план продукта, в котором и бизнес, и производство видели одно и то же. Этот же план становился основой приемки релиза, так как напротив каждой фичи сразу прописывались критерии приемки (примерный аналог паттерна «Смартинг»).

Выделяются два типа релизов: релиз развития и релиз поставки. Разница между ними в следующем. В релиз развития добавляют только фундаментальные фичи, которые являются абсолютно новым функционалом системы, общим для всех клиентов, зачастую значительно влияющим на архитектуру системы. Релиз же поставки включает в себя преимущественно фичи, которые нужны конкретному клиенту. Также релиз развития чаще требует более высококлассного программирования и тестирования, так как цена ошибки в нем крайне высока.



Иначе говоря, релиз развития направлен в основном на уже существующих и будущих клиентов компании, для которых компания стремится сделать продукт еще более удобным и функциональным. Этот релиз связан со стратегическим развитием продукта. Поэтому в него включают критические фичи, общие для всех клиентов. Отметим также, что фичи релиза развития (по крайней мере первые их версии) могут реализовываться фичами релизов поставки. При необходимости оптимизации или тиражирования этой фичи на всех клиентов формируется новая фича по развитию реализованной функциональности, и уже она часто выполняется в рамках релиза развития.

Релиз поставки же направлен строго на выполнение текущих контрактных обязательств компании по автоматизации деятельности новых клиентов.

Перечисленные понятия и отношения между ними наглядно проиллюстрированы диаграммой UML, приведенной на рис. 5.17.

К счастью, компания не пыталась параллельно развивать несколько веток продукта, актуальных для разных типов клиентов. В таком случае приведенная картинка стала бы многомерной и гораздо более сложной. Здесь я не буду рассказывать об этом подробно. Возможно, в третьей книге серии «Путь IT-профессионала» мы коснемся вопроса одновременного управления несколькими версиями одного и того же продукта. Сейчас я только дам ключик: в этом случае нужно стремиться иметь базовую версию продукта, развивающуюся в рамках релизов развития, и специфические «веточки», которые

в сумме с базовой версией дают клиентскую версию (релиз поставки). То есть архитектору придется изрядно попотеть, чтобы выделить базовую версию, если архитектура системы изначально некомпонентная или нет возможности выделить ядро или базовые компоненты/ код, которые используются во всех вариациях продукта.

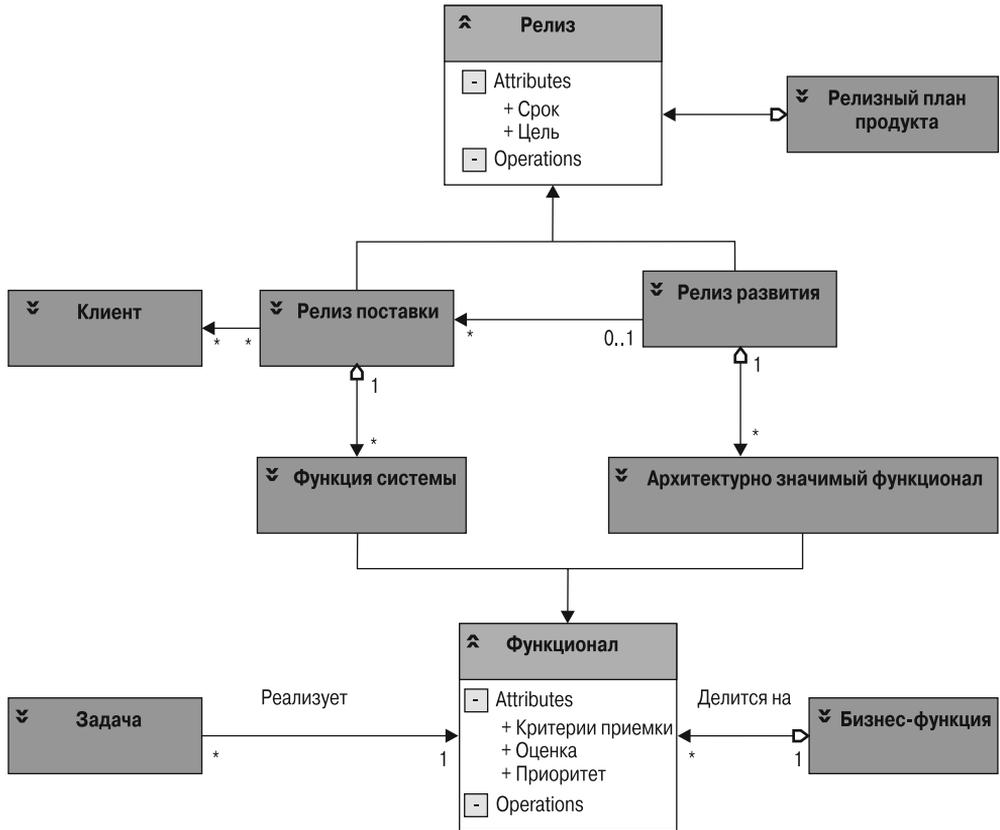


Рис. 5.17. Концептуальная модель понятий релизной модели



Обратите внимание на то, что мы поддерживали три высокоуровневых документа уровня всей системы: стратегию развития, функциональную концепцию и архитектурную концепцию. Это были «живые» документы уровня продукта, три кита развития продукта. Они не являются частью проектных спецификаций требований и архитектурных спецификаций. Более детально принципы стратегического управления описаны в паттерне «Стратегия и планирование в СУПр» в разделе 6.5.

Также акцентирую ваше внимание на том, что паттерн «Выученные уроки» применялся и на уровне стратегии, и на уровне архитектурных задач, и на уровне функциональных задач.

В целом место релизного планирования в укрупненном ЖЦ работы с клиентом компании выглядело примерно так (рис. 5.18).



**Рис. 5.18.** Укрупненный ЖЦ работы с клиентом

Однако, как я только что сказал, параллельно с этим процессом постоянно актуализировались три кита развития. Соответственно, на стадии бизнес-анализа рассматривалось влияние стратегии развития и в случае необходимости в нее вносились изменения, а на стадии анализа реализуемости проводился импакт-анализ функциональной и архитектурной концепций. Естественно, перед тем как быть принятыми, изменения также проходили цикл импакт-анализа на другие уже выполняющиеся проекты в компании.

Для того чтобы все это не расплзлось по швам, в компании были выделены роль функционального архитектора, которую выполнял ведущий системный аналитик, и роль внутреннего заказчика, которую выполнял ведущий методолог. Первый отвечал за импакт на стадии анализа реализуемости, а второй — на стадии бизнес-анализа, хотя это очень условное разделение.

Основное условие для использования этого паттерна — командная работа. С самого начала до самого конца задействованы все роли ЖЦ производства компании.

Да, пока мы не двинулись дальше, отвечу на вопрос, который, скорее всего, вертится у вас на языке: «А как же итерации?» Ну конечно же, итерации в этом ЖЦ предусмотрены.

На сайте книги, доступном по ссылке [http://bit.ly/pmway\\_lceexample](http://bit.ly/pmway_lceexample), в качестве примера нотации вы можете скачать ЖЦ производства в формате MS Visio. Это близкий к реальному ЖЦ одной из компаний-поставщиков комплексных услуг по автоматизации процессов формирования бюджета государственных органов на базе продукта собственной разработки. На этой схеме вы увидите высокоуровневую и концептуальную картину, не отражающую всех связей и последовательностей выполнения работ, но тем не менее дающую общее представление о подходе к ним. Это может быть полезно главным образом с точки зрения понимания того, где может использоваться паттерн «Релизное управление». Нотация, применяемая на диаграмме, в самом верхнем блоке показывает роли, участвующие в деятельности, затем идет блок с кратким названием деятельности, а нижний блок отвечает на вопрос, какие проектные артефакты создаются в результате деятельности.

Общие рекомендации по выбору жизненного цикла приведены на сайте книги, доступном по ссылке [http://bit.ly/pmway\\_choiceoflc](http://bit.ly/pmway_choiceoflc).

## 5.3. Паттерн «Обеспечение качества и анализ рисков»

Для каждого разрабатываемого продукта выдвигаются разные требования к качеству. Отвечает за выдвижение таких требований бизнес-дивизион, в частности менеджер по развитию продукта.



Рекомендую выдвигать требования к качеству в соответствии с ГОСТ 28195–89 «Оценка качества программных средств». Кроме того, может быть полезен стандарт ISO 9126, который представляет собой набор стандартов и определяет шесть основных категорий качества программного продукта: Functionality, Reliability, Usability, Efficiency, Maintainability, Portability.



В зависимости от выдвинутых требований к качеству и характеристик разрабатываемой системы могут меняться методология разработки ПО и иерархическая структура работ проекта (Work Breakdown Structure — WBS). На сайте книги вы можете скачать дополнительный материал «Выбор ЖЦ проектов в зависимости от характеристик системы и требований к качеству продукта», который поможет вам с выбором жизненного цикла разработки ПО. В этом разделе мы поговорим о WBS.

Каждая работа в WBS должна быть нацелена на выполнение требований к качеству системы либо на полное или частичное снятие рисков разработки некачественного продукта, например таких, как неудовлетворенность заказчика или противоречие ожиданий заказчика правилам его бизнеса.



Риски качества конечного продукта:

- ◆ бизнес-цели заказчика не достигнуты;
- ◆ систему невозможно применить в реальном бизнесе заказчика;
- ◆ заказчик остался неудовлетворен;
- ◆ ожидания заказчика противоречат правилам его бизнеса;
- ◆ поведение системы понято неправильно;
- ◆ нет полного понимания общего поведения системы;
- ◆ система не реализует все бизнес-цели заказчика;
- ◆ поведение системы не соответствует ожиданиям заказчика;
- ◆ поведение системы реализовано неправильно;
- ◆ ожидаемые функции не реализованы;
- ◆ варианты использования пропущены;
- ◆ возникают отторжение внедрения системы и сопротивление ему;
- ◆ реализованы требования, ненужные бизнесу;

- ♦ работа системы неудовлетворительна по скорости, масштабированию и нагрузке;
- ♦ реализация поведения неоптимальна;
- ♦ в системе имеется «голд платина» — реализация никому не нужных «дорогих» функций;
- ♦ низкое качество тестирования;
- ♦ невозможность проведения приемочных испытаний;
- ♦ сложности при поддержке и сопровождении продукта.

Подумайте, какие работы надо выполнить, какие артефакты (модели, требования, трассировки) построить, чтобы полностью или частично снять перечисленные риски.

Если вы испытываете затруднения, возможно, имеет смысл прочитать первую книгу серии «Путь IT-профессионала», особенно подразделы «Концептуальная модель типов требований», «Моделирование» и «Трассировки» раздела «Профессиональные и специальные навыки» главы «Аналитик».



Выполнением работ по разработке ПО невозможно снять некоторые риски, например риск изменения заказчиком утвержденных требований или риск невыполнения заказчиком своих обязательств. С такими управленческими рисками должны справляться менеджеры проекта и продукта.

Теперь проанализируем возможные причины возникновения названных рисков (табл. 5.4).

**Таблица 5.4.** Риски качества и возможные их причины

<b>Риск, ухудшающий качество конечного продукта</b>	<b>Вероятная причина наступления риска</b>
<b>1</b>	<b>2</b>
Бизнес-цели заказчика не достигнуты	Не выявлены и не утверждены заказчиком все его бизнес-требования
Систему невозможно применить в реальном бизнесе заказчика	Не выявлены и не утверждены заказчиком все бизнес-правила, действующие в его бизнесе
Заказчик остался не удовлетворен	Не выявлены и не утверждены заказчиком все его ожидания. Не выполнена трассировка правил бизнеса на ожидания заказчика
Ожидания заказчика противоречат правилам его бизнеса	Не выявлены и не утверждены заказчиком все варианты использования системы
Поведение системы понято неправильно	Завышенный уровень описания выявленных вариантов использования
Нет полного понимания общего поведения системы	Не выполнена трассировка бизнес-требований на варианты использования
Система не реализует все бизнес-цели заказчика	Не выполнена трассировка ожиданий заказчика и пользовательских требований на варианты использования

Продолжение ➤

Таблица 5.4 (продолжение)

1	2
Поведение системы не соответствует ожиданиям заказчика	Не выявлены и не утверждены в проектной команде все варианты использования системы
Поведение системы реализовано неправильно	Не выявлены и не утверждены в проектной команде все функциональные требования к системе
Ожидаемые функции не реализованы	Не выполнена трассировка от вариантов использования к функциональным требованиям
Варианты использования пропущены	
Возникают отторжение внедрения системы и сопротивление ему	Не выявлены и не утверждены заказчиком все требования пользователей системы
Реализованы требования, ненужные бизнесу	Не выполнена трассировка требований бизнеса на требования пользователей
Работа системы неудовлетворительна по скорости, масштабированию и нагрузке	Не выявлены и не утверждены в проектной команде нефункциональные требования
Реализация поведения неоптимальна	<p>Не выполнена трассировка нефункциональных требований на варианты использования.</p> <p>Не выполнена трассировка пользовательских требований на варианты использования.</p> <p>Не выполнена трассировка функциональных требований на нефункциональные.</p> <p>Не выполнена трассировка пользовательских требований на нефункциональные.</p> <p>Не выполнена трассировка бизнес-требований на нефункциональные.</p> <p>Не показана реализация каждого значимого варианта использования системы конкретными ее классами (оторванная от поведения архитектура может привести к излишней сложности архитектуры, кроме того, существует риск не заметить очевидных пересечений одной и той же функциональности при реализации разными классами). Без углубленного анализа поведения системы (построения диаграмм пригодности вариантов использования, активности, последовательности, модели анализа, логической модели системы) есть риск несвоевременного выявления нефункциональных требований, важных с точки зрения реализации продукта</p>
В системе имеется «голд платина» — реализация никому не нужных «дорогих» функций	Не выполнена трассировка утвержденных требований пользователя, бизнес-требований, бизнес-правил и ожиданий заказчика на варианты использования
Низкое качество тестирования	Отсутствие базы для создания тестовых спецификаций — не выявлены функциональные требования, нет описанных вариантов использования системы
Невозможность проведения приемочных испытаний	Нет утвержденных функциональных требований для формальной сдачи-приемки системы заказчику
Сложности при поддержке и сопровождении продукта	Отсутствие материалов для написания пользовательской документации, разработки рекомендаций и обучающих курсов

После определения вероятных причин наступления риска становится понятно, какие работы надо включить в WBS проекта, чтобы максимально снизить риски.



Далее приведен состав WBS, который на 90 % снижает вероятность возникновения перечисленных рисков (табл. 5.5). Примерно такую же таблицу вы видели в книге «Путь аналитика». Здесь я попытался показать, какие работы/паттерны лучше всего снимают риски.

**Таблица 5.5.** Управление рисками, ухудшающими качество

Риски, ухудшающие качество конечного продукта	Паттерны управления проектом и состав WBS, частично снимающие риски
<b>1</b>	<b>2</b>
<p>Бизнес-цели заказчика не достигнуты — не выявлены и не утверждены заказчиком все его бизнес-требования</p>	<p>Паттерн «Управление средой проекта»</p>
<p>Систему невозможно применить в реальном бизнесе заказчика — не выявлены и не утверждены заказчиком все бизнес-правила, действующие в его бизнесе</p>	<p>Паттерн «Управление средой проекта»</p>
<p>Заказчик остался не удовлетворен — не выявлены и не утверждены заказчиком все его ожидания.</p> <p>Нет полного понимания общего поведения системы — уровень описания выявленных вариантов использования очень высокий.</p> <p>Возникают отторжение внедрения системы и сопротивление ему — не выявлены и не утверждены заказчиком все требования пользователей системы.</p> <p>Ожидания заказчика противоречат правилам его бизнеса — не выполнена трассировка правил бизнеса на ожидания заказчика.</p> <p>Система не реализует все бизнес-цели заказчика — не выполнена трассировка бизнес-требований на варианты использования.</p> <p>Поведение системы понято неправильно — не выявлены и не утверждены заказчиком все варианты использования системы.</p> <p>Поведение системы не соответствует ожиданиям заказчика и даже противоречит им — не выполнена трассировка ожиданий заказчика и пользовательских требований на варианты использования.</p> <p>Поведение системы реализовано неправильно — не выявлены и не утверждены в проектной команде все варианты использования системы.</p> <p>Ожидаемые функции не реализованы — не выявлены и не утверждены в проектной команде все функциональные требования к системе.</p> <p>Варианты использования пропущены — не выполнена трассировка от вариантов использования к функциональным требованиям.</p>	<p>Паттерн «Итеративная разработка ПО»</p> <p>WBS включает в себя работы по созданию:</p> <ul style="list-style-type: none"> <li>• моделей: <ul style="list-style-type: none"> <li>— концептуальной модели предметной области;</li> <li>— концептуальной модели системы;</li> <li>— use case-модели на уровне «название + цель кейса + основной поток + альтернативные потоки» (для кейсов, выбранных аналитиком, архитектором, МП и МПР);</li> <li>— диаграммы пригодности для выбранных кейсов с альтернативными потоками;</li> <li>— логической модели системы;</li> <li>— диаграммы активности выбранных кейсов с детальной спецификацией;</li> <li>— (по совместному решению аналитика, архитектора, МП);</li> <li>— диаграммы последовательности для выбранных кейсов с детальной спецификацией;</li> <li>— диаграммы активности выбранных кейсов с альтернативными потоками;</li> <li>— диаграммы последовательности для выбранных кейсов с альтернативными потоками;</li> <li>— модели данных;</li> </ul> </li> <li>• проекта требований, интегрированного с моделью, в котором выделяются: <ul style="list-style-type: none"> <li>— бизнес-требования;</li> <li>— бизнес-правила;</li> <li>— ожидания ЗП;</li> <li>— пользовательские требования;</li> <li>— ограничения;</li> <li>— допущения;</li> </ul> </li> </ul>

Продолжение ⇨

Таблица 5.5 (продолжение)

1	2
<p>Реализованы требования, ненужные бизнесу, — не выполнена трассировка требований бизнеса на требования пользователей.</p> <p>Работа системы неудовлетворительна по скорости, масштабированию и нагрузке — не выявлены и не утверждены в проектной команде нефункциональные требования.</p> <p>Реализация поведения неоптимальна:</p> <ul style="list-style-type: none"> <li>• не выполнена трассировка нефункциональных требований на варианты использования;</li> <li>• не выполнена трассировка пользовательских требований на варианты использования;</li> <li>• не выполнена трассировка функциональных требований на нефункциональные;</li> <li>• не выполнена трассировка пользовательских требований на нефункциональные требования;</li> <li>• не выполнена трассировка бизнес-требований на нефункциональные требования;</li> <li>• не показана реализация каждого значимого варианта использования системы конкретными ее классами (оторванная от поведения архитектура может привести к излишней сложности архитектуры, кроме того, существует риск не заметить очевидных пересечений одной и той же функциональности при реализации разными классами). Без углубленного анализа поведения системы (построения диаграмм пригодности вариантов использования, активности, последовательности, модели анализа, логической модели системы) есть риск несвоевременного выявления нефункциональных требований, важных с точки зрения реализации продукта.</li> </ul> <p>В системе имеется «голд платина» — реализация никому не нужных «дорогих» функций, — не выполнена трассировка утвержденных требований пользователя, бизнес-требований, бизнес-правил и ожиданий заказчика на варианты использования.</p> <p>Низкое качество тестирования — отсутствие базы для создания тестовых спецификаций — не выявлены функциональные требования, нет описанных вариантов использования системы.</p> <p>Невозможность проведения приемочных испытаний — нет утвержденных функциональных требований для формальной сдачи-приемки системы заказчику.</p> <p>Сложности при поддержке и сопровождении продукта — отсутствие материалов для написания пользовательской документации, разработки рекомендаций и обучающих курсов</p>	<ul style="list-style-type: none"> <li>— перечень поддерживаемых ОС;</li> <li>— use case;</li> <li>— функциональные требования;</li> <li>— нефункциональные требования;</li> <li>— трассировка бизнес-требований на пользовательские требования;</li> <li>— трассировка бизнес-требований на use cases;</li> <li>— трассировка бизнес-правил на use cases;</li> <li>— трассировка бизнес-правил на нефункциональные требования;</li> <li>— трассировка пользовательских требований на use cases;</li> <li>— трассировка use cases на функциональные требования;</li> <li>— трассировка ограничений на нефункциональные требования;</li> <li>— трассировка допущений на функциональные требования;</li> <li>— трассировка допущений на нефункциональные требования.</li> </ul> <p>По совместному решению аналитика, МП и менеджера продукта выполняются:</p> <ul style="list-style-type: none"> <li>— трассировка бизнес-требований на ожидания ЗЛ;</li> <li>— трассировка ожиданий ЗЛ на use cases;</li> <li>— трассировка нефункциональных требований на варианты использования.</li> </ul> <p>По совместному решению аналитика, архитектора, МП выполняются:</p> <ul style="list-style-type: none"> <li>— трассировка бизнес-требований на нефункциональные требования;</li> <li>— трассировка пользовательских требований на нефункциональные требования;</li> <li>— трассировка ожиданий ЗЛ на нефункциональные требования;</li> <li>— трассировка функциональных требований на нефункциональные требования;</li> </ul> <ul style="list-style-type: none"> <li>• спецификаций: <ul style="list-style-type: none"> <li>— «Словарь терминов и понятий»;</li> <li>— «Бизнес-правила и бизнес-требования»;</li> <li>— «Обзор вариантов использования»;</li> <li>— «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР);</li> <li>— «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР и утвержденных комиссией по контролю за изменениями);</li> <li>— «Требования к системе»;</li> </ul> </li> </ul>

1	2
	<ul style="list-style-type: none"> <li>– «Обзор решения»;</li> <li>– «Словарь данных».</li> </ul> Полноценное управление требованиями: <ul style="list-style-type: none"> <li>• статус требований «предложено – одобрено (feasibility) – согласовано – утверждено – реализовано»;</li> <li>• обсуждение в виде дискуссий.</li> </ul> Паттерн «Открытые вопросы». Паттерн «С ног на голову, или ПМП ПСР». SLA в соответствии со стратегией сервисного обслуживания. Часто используется метод дифференциации клиентов

Как видите, можно разработать WBS, которая полностью или частично снимет максимально возможное количество рисков, однако обусловит значительное увеличение стоимости и времени выполнения проекта. Поэтому необходимо управлять рисками и требованиями к качеству.

Менеджер продукта должен выдвигать требования к качеству продукта, аналитик — анализировать их и формировать список рисков, мешающих достижению требуемого качества продукта. Менеджер продукта должен ранжировать риски как критические, приоритетные, некритические, то есть фактически какие-то риски принять, а какие-то отметить как те, что крайне необходимо устранить.

Менеджер проекта должен управлять рисками — снимать вероятность их возникновения путем включения соответствующих работ в WBS проекта.

Время выполнения проектных работ может быть сокращено менеджерами продукта, программы проектов и проекта в целом.

Менеджер продукта и менеджер программы проектов должны стараться не заказывать создание спецификаций требований, если нет серьезной потребности в их наличии в виде документа. Кроме того, время и стоимость выполнения аналитических работ могут быть уменьшены путем снижения требований к качеству конкретного продукта, а также за счет принятия определенных рисков.

Менеджер проекта должен планировать и проводить работы по повышению уровня знаний и навыков членов проектной команды по работе с аналитическими артефактами. В идеале у проектной команды должна быть такая квалификация, чтобы ее членам было удобнее работать с моделями и проектом требований, чем со спецификациями требований. Уменьшению количества времени и стоимости выполнения аналитических работ будут способствовать также эффективные коммуникации внутри команды (обсуждения, встречи, процедуры согласования и утверждения, разрешение конфликтных ситуаций).

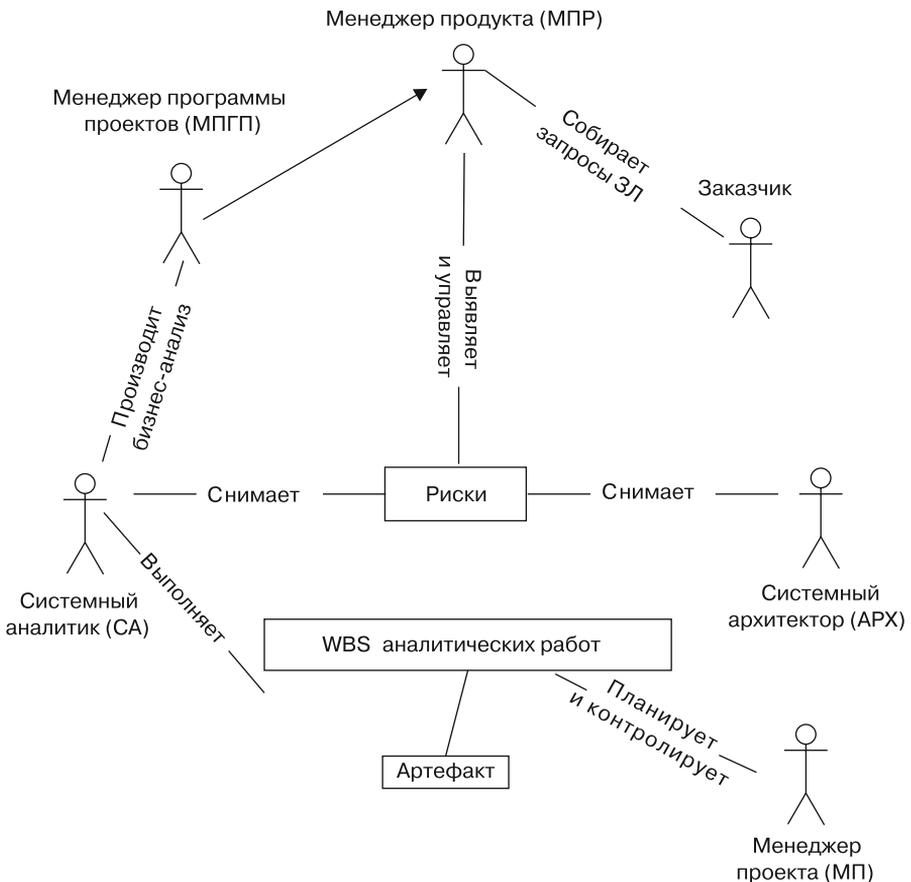


Обобщим сказанное, обозначив ответственность ролей, участвующих в управлении требованиями к качеству продукта (табл. 5.6).

**Таблица 5.6.** Ответственность ролей в проекте

Роль	Ответственность
Менеджер продукта, менеджер программы проектов	Определить требования к качеству конкретного продукта. Выявлять риски и управлять ими. Заказывать проектные работы, а не спецификации
Менеджер проекта	Выбрать форму фиксации результатов работ аналитика. Обеспечить эффективные коммуникации внутри команды (обсуждения, встречи, процедуры согласования и утверждения, разрешение конфликтных ситуаций). Планировать и проводить мероприятия по повышению уровня знаний и навыков членов проектной команды по работе с аналитическими артефактами
Системный аналитик	Снять риски посредством выполнения аналитических работ

Проиллюстрируем сказанное (рис. 5.19).



**Рис. 5.19.** Ответственность и координация проектных ролей с точки зрения рисков проекта



Здесь важно отметить, что риски достижения качества продукта снимает не только аналитик, но и архитектор и менеджер проекта. Архитектор выявляет из требований к качеству продукта архитектурные риски и отвечает за то, чтобы максимально снизить их. Менеджер обязан (!) определить, какие риски являются неснимаемыми, а какие — снимаемыми по возможности.

В заключение я хочу еще раз привлечь ваше внимание к важным особенностям обеспечения качества продуктов посредством требований к качеству продуктов, которые вы должны понимать и учитывать в своей деятельности.

1. Все требования к продукту выявляются из проекта в проект (из релиза в релиз), начиная с самого первого релиза, и в момент создания  $i$ -го релиза все актуальные требования являются требованиями продукта. Тем не менее, когда мы планируем создавать новый продукт или развивать старый, мы сразу можем сказать, какие требования качества выдвигаем к продукту. Например, если продукт — система банковских счетов, то мы обязательно выдвинем требование: «Должны быть определены требования к защите и проверке данных», а если продукт — калькулятор, то нет.
2. Следует разделять документацию и цель выполнения какой-то работы по выявлению и анализу информации. Документация — всего лишь способ закрепить результат анализа.
3. В зависимости от наличия каждого из предлагаемых требований качества будут выполняться определенные работы из WBS аналитика — это основной критерий внесения требования качества в список требований к качеству продукта.
4. Согласование требований с заказчиком — длительный процесс, который увеличивает время аналитических работ и сказывается на стоимости продукта. Появление почти каждого требования к качеству приводит к необходимости согласования с заказчиком.

Также крайне полезно напомнить читателю о существовании практики ревью. Ревью может быть одноранговым — его производят специалисты одной квалификации — и согласовательным — это делает специалист, отвечающий за согласование/утверждение артефакта. Одноранговое ревью — прекрасный паттерн повышения качества при подготовке практически любых проектных артефактов. Одноранговое ревью предполагает не детальную вычитку, а скорее чтение по диагонали, но даже оно часто позволяет заметить нестыковки и неочевидные для глаза автора документа/артефакта моменты — преодолеть эффект «замыливания глаза».

Применительно к программному коду крайне полезно проводить код ревью и архитектурное ревью. Причем если речь идет о доработках, влияющих на архитектуру системы или связанных с ней или ее основными технологическими особенностями, то лучше, если такое ревью проведут внешние ревьюеры — или специалисты по разработке другого продукта (в крупной компании), или вообще сторонняя компания. При этом в случае архитектурного ревью основной

упор делается на архитектурные спецификации, диаграммы классов, последовательности, описание алгоритмов и проверку реализации всего этого в коде. То есть такое ревью требует значительных затрат труда и средств. Но это крайне полезный способ объективно повысить качество продукта.

## 5.4. Паттерн «Пакеты спецификаций требований»

Во многих компаниях, ориентированных на результат, процессный подход не всегда идеален, а иногда его крайне сложно принимают сотрудники. В таком случае можно выстроить процессы через документооборот. Для таких компаний очень актуален данный паттерн.

Суть паттерна в том, что формируются некие правила, согласно которым в разных типах проектов готовятся разные типы спецификаций (портфель спецификаций). При этом каждый портфель — это четко определенные работы и коммуникации по обсуждению, согласованию и утверждению артефактов. Кроме того, артефакты связаны по смыслу друг с другом и значит, чтобы я согласовал какой-то один артефакт, часто уже должен существовать какой-то другой артефакт (какие-то другие артефакты), логически связанный с анализируемым. То есть через работу с артефактами и выстраивается процесс разработки. Этот паттерн хорошо работает вместе с паттерном ПУД, описанным ранее.

Можно выделить следующие основные принципы формирования пакетов спецификаций требований для производственных проектов.

- ◆ Общая трудоемкость работ по проекту зависит от сложности и объема WBS, которая напрямую зависит от желаемого состава пакета спецификаций.
- ◆ Возможность выбрать заранее предлагаемый или сформировать собственный пакет спецификаций для каждого проекта (типа проекта). Возможны исключения, когда для какого-то проекта требуется более детальный пакет спецификаций.
- ◆ Для каждого продукта в качестве наиболее полного с точки зрения минимальной достаточности для качественной разработки систем будет рекомендован один из вариантов пакетов спецификаций.
- ◆ Вместо одного рекомендованного для продукта пакета требований в зависимости от выводов о достаточности или избыточности пакета в проектах по развитию данного продукта с течением времени может быть рекомендован другой пакет требований.
- ◆ Как и для продукта, для каждого типа проектов имеется минимально допустимый и рекомендуемый пакеты спецификаций требований и WBS.
- ◆ Вместо одного рекомендованного для типа проектов пакета требований в зависимости от выводов о достаточности или избыточности пакета в проектах

конкретного типа с течением времени может быть рекомендован другой пакет требований.

- ◆ Приоритетной рекомендацией по выбору пакета требований в проекте по развитию продукта является рекомендованный пакет требований для продукта в целом.
- ◆ Второй по приоритету рекомендацией является рекомендация выбора пакета спецификаций требований для конкретного типа проекта.
- ◆ В случае если в проекте по развитию продукта принимается решение реализовывать менее полный состав пакета требований, чем рекомендованный для продукта в целом, при закрытии проекта с санкции ССВ продукта пакет требований проекта должен быть переработан в пакет требований, рекомендованный для продукта.
- ◆ Для помощи в формировании собственного пакета спецификаций в СУП будут созданы матрица зависимостей артефактов (спецификаций требований) друг от друга и матрица происхождения артефактов из WBS.

### 5.4.1. Состав пакетов спецификаций требований



---

В данном разделе приведено описание состава каждого пакета спецификаций и краткое описание основных работ для получения каждого пакета спецификаций требований.

---

### 5.4.2. Пакет Extreme

Пакет включает в себя:

- ◆ устав проекта;
- ◆ документ «Концепция системы», расширенный разделом «общее описание функциональности», — общее описание поведения системы в простой неструктурированной форме.

По окончании проекта необходим максимально детальный анализ для выявления и слияния требований с бейзлайном требований.

### 5.4.3. Пакет Simple

Пакет включает в себя:

- ◆ устав проекта;
- ◆ план управления проектом (в виде протокола совещания, в котором зафиксированы вопросы управления требованиями, качеством, рисками, коммуникациями);

- ◆ список вех;
- ◆ простой **проект требований**, в котором выделяются:
  - ◆ ограничения;
  - ◆ допущения;
  - ◆ перечень поддерживаемых ОС;
  - ◆ функциональные требования;
  - ◆ нефункциональные требования;
  - ◆ трассировки функциональных требований на нефункциональные, выполняемые по решению МП;
- ◆ **спецификацию** «Требования к системе», которая имеет следующие особенности:
  - ◆ управление требованиями минимальное;
  - ◆ статус требования «предложено — утверждено»;
  - ◆ проходит обсуждение в виде дискуссий.

По окончании проекта необходим детальный анализ для выявления и слияния требований с бейзлайном требований.

#### 5.4.4. Пакет Basic

Это базовый пакет спецификаций требований. Он включает в себя:

- ◆ устав проекта;
- ◆ план управления проектом;
- ◆ план управления качеством;
- ◆ ПУД;
- ◆ план-расписание проекта;
- ◆ реестр проектных рисков;
- ◆ отчеты по проекту и информационные письма;
- ◆ **модель use case** на уровне «название + цель кейса + основной поток» (для кейсов, выбранных аналитиком, архитектором, МП и МПР);
- ◆ **проект требований**, интегрированный с моделью, в котором выделяются:
  - ◆ ограничения;
  - ◆ допущения;
  - ◆ перечень поддерживаемых ОС;
  - ◆ бизнес-требования;
  - ◆ use case;
  - ◆ функциональные требования;

- ◆ нефункциональные требования;
- ◆ трассировки, выполняемые по совместному решению аналитика, архитектора, МП:
  - трассировка use cases на функциональные требования;
  - трассировка функциональных требований на нефункциональные;
  - трассировка ограничений на функциональные требования;
  - трассировка ограничений на нефункциональные требования;
  - трассировка допущений на функциональные требования;
  - трассировка допущений на нефункциональные требования;
- ◆ трассировка бизнес-требований на use cases, выполняемая по совместному решению аналитика, МП и менеджера продукта;
- ◆ **спецификации:**
  - ◆ «Словарь терминов и понятий»;
  - ◆ «Обзор вариантов использования»;
  - ◆ «Требования к системе»;
- ◆ управление требованиями средней сложности;
- ◆ статус требования «предложено – одобрено (feasibility) – утверждено»;
- ◆ обсуждение в виде дискуссий.

По окончании проекта нужны анализ для более глубокой детализации функциональных требований и слияние требований с бейзлайном требований.

### 5.4.5. Пакет Advanced

Пакет включает в себя:

- ◆ устав проекта;
- ◆ план управления проектом;
- ◆ ПУД;
- ◆ план-расписание проекта;
- ◆ реестр проектных рисков;
- ◆ реестр работ (или WAS) с возможностью всем заинтересованным лицам в любое время увидеть текущее состояние работ проекта;
- ◆ **модели use case** на уровне «название + цель кейса + основной поток (для кейсов, выбранных аналитиком, архитектором, МП и МПР) + альтернативные потоки» (для кейсов, выбранных аналитиком, архитектором, МП и МПР);
- ◆ диаграммы пригодности для выбранных кейсов с детальной спецификацией;
- ◆ выполняемые по совместному решению аналитика, архитектора, МП:

- ◆ логическая модель системы;
- ◆ диаграммы активности выбранных кейсов с детальной спецификацией;
- ◆ диаграммы последовательности для выбранных кейсов с детальной спецификацией;
- ◆ **проект требований**, интегрированный с моделью, в котором выделяются:
  - ◆ бизнес-требования;
  - ◆ бизнес-правила;
  - ◆ пользовательские требования;
  - ◆ ограничения;
  - ◆ допущения;
  - ◆ перечень поддерживаемых ОС;
  - ◆ use case;
  - ◆ функциональные требования;
  - ◆ нефункциональные требования;
  - ◆ трассировка use cases на функциональные требования;
  - ◆ трассировки, выполняемые по совместному решению аналитика, архитектора, МП:
    - трассировка пользовательских требований на use cases;
    - трассировка пользовательских требований на нефункциональные требования;
    - трассировка функциональных требований на нефункциональные требования;
    - трассировка ограничений на функциональные требования;
    - трассировка ограничений на нефункциональные требования;
    - трассировка допущений на функциональные требования;
    - трассировка допущений на нефункциональные требования;
  - ◆ трассировки, выполняемые по совместному решению аналитика, МП и менеджера продукта:
    - трассировка бизнес-требований на use cases;
    - трассировка бизнес-требований на нефункциональные требования;
- ◆ **спецификации:**
  - ◆ «Словарь терминов и понятий»;
  - ◆ «Бизнес правила и бизнес требования»;
  - ◆ «Обзор вариантов использования»;
  - ◆ «Спецификация варианта использования системы» (для кейсов выбранных аналитиком, архитектором, МП и МПР) — детальная спецификация основных потоков;

- ◆ «Спецификация варианта использования системы» (для кейсов выбранных аналитиком, архитектором, МП и МПР и утвержденных комиссией по контролю за изменениями) — полная и детальная спецификация;
- ◆ «Требования к системе»;
- ◆ полноценное управление требованиями:
  - ◆ статус требований «предложено — одобрено (feasibility) — согласовано — утверждено — реализовано»;
  - ◆ обсуждение в виде дискуссий.

По окончании проекта нужны анализ функциональных требований и слияние требований с бейзлайном требований.

## 5.4.6. Пакет Gold

Пакет включает в себя:

- ◆ устав проекта;
- ◆ план управления проектом;
- ◆ план обучения сотрудников;
- ◆ план совершенствования процессов;
- ◆ план управления качеством;
- ◆ ПУД;
- ◆ план-расписание проекта;
- ◆ реестр проектных рисков;
- ◆ WAS — с возможностью сбора и анализа статистики, получения отчетов и просмотра ЗЛ в любой момент времени состояния дел по работам и артефактам;
- ◆ модели:
  - ◆ концептуальную модель предметной области;
  - ◆ концептуальную модель системы;
  - ◆ use case-модель на уровне «название + цель кейса + основной поток + альтернативные потоки (для кейсов, выбранных аналитиком, архитектором, МП и МПР);
  - ◆ диаграммы пригодности для выбранных кейсов с альтернативными потоками;
  - ◆ логическую модель системы;
  - ◆ диаграммы активности выбранных кейсов с детальной спецификацией;
  - ◆ диаграммы последовательности для выбранных кейсов с детальной спецификацией;
  - ◆ выполняемые по совместному решению аналитика, архитектора, МП:

- диаграммы активности выбранных кейсов с альтернативными потоками;
- диаграммы последовательности для выбранных кейсов с альтернативными потоками;
- ◆ **проект требований**, интегрированный с моделью, в котором выделяются:
  - ◆ бизнес-требования;
  - ◆ бизнес-правила;
  - ◆ пользовательские требования;
  - ◆ ограничения;
  - ◆ допущения;
  - ◆ перечень поддерживаемых ОС;
  - ◆ use case;
  - ◆ функциональные требования;
  - ◆ нефункциональные требования;
  - ◆ трассировка пользовательских требований на use cases;
  - ◆ трассировка use cases на функциональные требования;
  - ◆ трассировка ограничений на функциональные требования;
  - ◆ трассировка ограничений на нефункциональные требования;
  - ◆ трассировка допущений на функциональные требования;
  - ◆ трассировка допущений на нефункциональные требования;
  - ◆ трассировки, выполняемые по совместному решению аналитика, МП и менеджера продукта:
    - трассировка бизнес-требований на пользовательские требования;
    - трассировка бизнес-требований на use cases;
    - трассировка бизнес-требований на нефункциональные требования;
    - трассировка бизнес-правил на use cases;
    - трассировка бизнес-правил на нефункциональные требования;
  - ◆ трассировки, выполняемые по совместному решению аналитика, архитектора и МП:
    - трассировка бизнес-требований на нефункциональные требования;
    - трассировка функциональных требований на нефункциональные требования;
- ◆ **спецификации:**
  - ◆ «Словарь терминов и понятий»;
  - ◆ «Бизнес-правила и бизнес-требования»;
  - ◆ «Обзор вариантов использования»;

- ◆ основных потоков «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР) — детальная спецификация;
- ◆ «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР и утвержденных комиссией по контролю за изменениями) — полная и детальная спецификация;
- ◆ «Требования к системе»;
- ◆ «Обзор решения»;
- ◆ полноценное управление требованиями:
  - ◆ статус требований «предложено — одобрено (feasibility) — согласовано — утверждено — реализовано»;
  - ◆ обсуждение в виде дискуссий.

По окончании проекта нужно слияние требований с бейзлайном требований.

## 5.4.7. Пакет VIP

Пакет включает в себя:

- ◆ устав проекта;
- ◆ план управления проектом;
- ◆ план обучения сотрудников;
- ◆ план совершенствования процессов;
- ◆ план управления качеством;
- ◆ ПУД;
- ◆ план-расписание проекта;
- ◆ реестр проектных рисков;
- ◆ WAS;
- ◆ Document Management System (часто применяется как подсистема WAS) с возможностью сбора и анализа статистики, получения отчетов и просмотра заинтересованными лицами в любой момент состояния дел по работам и артефактам;
- ◆ **модели:**
  - ◆ концептуальную модель предметной области;
  - ◆ концептуальную модель системы;
  - ◆ use case-модель на уровне «название + цель кейса + основной поток + альтернативные потоки» (для кейсов, выбранных аналитиком, архитектором, МП и МПР);
  - ◆ диаграммы пригодности для выбранных кейсов с альтернативными потоками;

- ◆ логическую модель системы;
- ◆ диаграммы активности выбранных кейсов с детальной спецификацией;
- ◆ диаграммы последовательности для выбранных кейсов с детальной спецификацией;
- ◆ выполняемые по совместному решению аналитика, архитектора и МП:
  - диаграммы активности выбранных кейсов с альтернативными потоками;
  - диаграммы последовательности для выбранных кейсов с альтернативными потоками;
  - модель данных;
- ◆ **проект требований**, интегрированный с моделью, в котором выделяются:
  - ◆ бизнес-требования;
  - ◆ бизнес-правила;
  - ◆ ожидания заинтересованных лиц;
  - ◆ пользовательские требования;
  - ◆ ограничения;
  - ◆ допущения;
  - ◆ перечень поддерживаемых ОС;
  - ◆ use case;
  - ◆ функциональные требования;
  - ◆ нефункциональные требования;
  - ◆ трассировка бизнес-требований на пользовательские требования;
  - ◆ трассировка бизнес-требований на use cases;
  - ◆ трассировка бизнес-правил на use cases;
  - ◆ трассировка бизнес-правил на нефункциональные требования;
  - ◆ трассировка пользовательских требований на use cases;
  - ◆ трассировка use cases на функциональные требования;
  - ◆ трассировка ограничений на нефункциональные требования;
  - ◆ трассировка допущений на функциональные требования;
  - ◆ трассировка допущений на нефункциональные требования;
  - ◆ трассировки, выполняемые по совместному решению аналитика, МП и менеджера продукта:
    - трассировка бизнес-требований на ожидания заинтересованных лиц;
    - трассировка ожиданий заинтересованных лиц на use cases;
  - ◆ трассировки, выполняемые по совместному решению аналитика, архитектора и МП:

- трассировка бизнес-требований на нефункциональные требования;
- трассировка пользовательских требований на нефункциональные требования;
- трассировка ожиданий заинтересованных лиц на нефункциональные требования;
- трассировка функциональных требований на нефункциональные требования;

◆ **спецификации:**

- ◆ «Словарь терминов и понятий»;
- ◆ «Бизнес-правила и бизнес-требования»;
- ◆ «Обзор вариантов использования»;
- ◆ «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР) — детальная спецификация основных потоков;
- ◆ «Спецификация варианта использования системы» (для кейсов, выбранных аналитиком, архитектором, МП и МПР и утвержденных комиссией по контролю за изменениями) — полная и детальная спецификация;
- ◆ «Требования к системе»;
- ◆ «Обзор решения»;
- ◆ «Словарь данных»;

◆ **полноценное управление требованиями:**

- ◆ статус требований «предложено — одобрено (feasibility) — согласовано — утверждено — реализовано»;
- ◆ обсуждение в виде дискуссий.

По окончании проекта необходимо слияние требований с бейзлайном требований.

## 5.4.8. Расширения пакетов спецификаций требований

Каждый из рассмотренных пакетов может расширяться следующими требованиями и методами анализа.

### Требования к взаимодействию с пользователем

Пакет дополняют:

◆ **модели:**

- ◆ модель взаимодействия интерфейсных форм;
- ◆ модели, выполняемые по совместному решению аналитика, GUI дизайнера, МП:

- состава интерфейсных форм;
- состава отчетов;
- ◆ добавления к **проекту требований**:
  - ◆ требование типа «интерфейсная форма» (перечень интерфейсных форм);
  - ◆ требования к интерфейсным формам (внешний вид, поведение);
  - ◆ требования к отчетам;
  - ◆ трассировки, выполняемые по совместному решению аналитика, менеджера по локализации, GUI дизайнера, МП:
    - трассировка use cases на интерфейсные формы;
    - трассировка use cases на требования к интерфейсным формам;
    - трассировка use cases на требования к отчетам;
- ◆ **спецификации**:
  - ◆ «Спецификация прототипов пользовательских интерфейсов»;
  - ◆ «Спецификация отчетов системы».

По окончании проекта нужен анализ добавленных к пакету требований для анализа связи с функциональными требованиями и для слияния требований с бейзлайном требований.

## Требования к взаимодействию компонентов и/или к взаимодействию с внешними системами

Пакет дополняют:

- ◆ **модели** — компонентная диаграмма;
- ◆ добавления к **проекту требований**:
  - ◆ требование типа «интерфейсная форма» (перечень интерфейсных форм);
  - ◆ требования к интерфейсу взаимодействия;
  - ◆ трассировки, выполняемые по совместному решению аналитика, архитектора, МП:
    - ◆ трассировка use cases на требования к взаимодействию;
    - ◆ требования к взаимодействию на нефункциональные требования;
- ◆ «Спецификация интерфейсов системы».

По окончании проекта нужно проанализировать добавленные к пакету требования на предмет связи с функциональными требованиями и для слияния требований с бейзлайном требований к продукту.

Это один из самых сложных паттернов в этой книге.

Если вы дочитали до этого места — честь вам и хвала. Этот материал был сложен и для изложения, и для восприятия. Надеюсь, у меня получилось как можно проще и понятнее изложить суть этого паттерна.

Если же вы не поняли, о чем идет речь, я рекомендую вам прочитать первую книгу из серии «Путь IT-профессионала». Возможно, у вас недостаточно знаний о существующих артефактах разработки ПО.

Если у вас остались вопросы, обратитесь ко мне на Facebook или на сайте книги «Путь аналитика» (<http://bp4you.ru>) — буду рад ответить. На сайте книги также доступна презентация по работе с пакетами требований ([http://bit.ly/pmway\\_reqpackage](http://bit.ly/pmway_reqpackage)).

## 5.5. Паттерн CMMI

CMMI как модель, проверенная годами, сама по себе отлично работает как паттерн. В этом смысле приведенные ранее паттерны «Пакеты спецификаций требований» и «Обеспечение качества и анализ рисков» в чем-то схожи со CMMI, так как также содержат в себе рекомендации по составу работ и перечню артефактов, которые можно использовать в проекте в зависимости от требований к качеству продукта, масштабов проекта, рисков, среды и уровня зрелости команды.

В книге «Путь аналитика» я довольно подробно рассказывал про опыт построения процессов с использованием модели CMMI (DAR, концептуальная модель СУПр) и в этой книге подробно на этом останавливаться не буду.



В CMMI существуют общие цели — некоторые общие принципы для всех групп процессов и специфические цели каждой процессной группы — основные цели ее процессов. Для некоторых процессных групп общие цели дополнительно детализируются с точки зрения того, что конкретно нужно выполнить в рамках совершенствования процессов этой процессной группы для достижения общей цели в рамках компании. Для каждой цели CMMI рекомендует практики для ее достижения.



С точки зрения применения модели как паттерна крайне полезно построить простую таблицу (табл. 5.7) по всем общим и специфическим целям и использовать ее как рабочий инструмент.

**Таблица 5.7.** Планирование и управление реализацией паттерна CMMI

УИД	Вид	Текст	Очередь	Сделано в целевой СУ	Сделано в СУПр
1	2	3	4	5	6
SP 1.3	SP	Управлять изменениями требований по мере их возникновения в проекте	1, 2, 3		
	SP Q	Источник каждого нового требования или изменения существующего требования известен и документирован	1	СУТ:: STRK, BR, Brule + трассировки + статус требования + автоматическая история изменения в ReqPRO	СУП::процедура управления изменениями

Продолжение ⇨

Таблица 5.7 (продолжение)

1	2	3	4	5	6
	SP Q	Менеджер проекта имеет возможность проследить измерения изменчивости требований, чтобы решить, нужно ли изменять способы контроля и управления	3		СУТ, СУП, СККП::метрики и автоматизация
	SP Q	По мере появления дополнительных требований все требования обновляются, при необходимости меняются существующие требования	1		
	WP	Требования имеют статус (например, «предложено», «согласовано», «утверждено», «разработано», «протестировано»)	2		
	WP	Требования хранятся в базе данных	1		
	WP	Решения по управлению требованиями хранятся в базе данных	2	СУТ::обсуждения требований в ReqPRO, служебные поля «Обоснование», «Комментарии» и т. п.	СУП::процедура управления изменениями
	Sub	Все требования и изменения в них, выдвинутые к проекту или выдвинутые самим проектом, документируются	2	СУТ::обсуждение требований в ReqPRO, служебные поля «Обоснование», «Комментарии» и т. п.	СУП::процедура управления изменениями
	Sub	История изменений требований, а также причины этих изменений сохраняются	2		
	Sub	Влияние изменения требований оценивается с точки зрения заинтересованных лиц	3		СУП::процедура управления изменениями; СУАР + СККП + СУП + СУТ::анализ влияния запрошенных измерений (импакт-анализ); СУП::оценка затрат
	Sub	Данные о требованиях и их изменениях делаются доступными для проекта	3		СУП::коммуникации в проекте

В графе «Очередь» указана очередность, в которой мы собираемся реализовать необходимые мероприятия для того, чтобы такая практика появилась в компании.

В нашем примере для выполнения практики SP1.3 области Requirement Management «Manage changes to the requirements as they evolve during the project» («Управлять изменениями в требованиях по мере выявления их в проекте») будем последовательно выполнять следующие рекомендации.

### **Первая очередь**

- ◆ Источник каждого нового требования или изменения существующего требования известен и документирован.
- ◆ По мере появления дополнительных требований все требования обновляются, при необходимости меняются существующие требования.
- ◆ Требования хранятся в базе данных.

### **Вторая очередь**

- ◆ Требования имеют статус (например, «предложено», «согласовано», «утверждено», «разработано», «протестировано»).
- ◆ Решения по управлению требованиями хранятся в базе данных.
- ◆ История изменений требований, а также причины этих изменений сохраняются.

### **Третья очередь**

- ◆ Менеджер проекта имеет возможность проследить измерения изменчивости требований, чтобы решить, нужно ли изменять способы контроля и управления.
- ◆ Влияние изменения требований оценивается с точки зрения заинтересованных лиц.
- ◆ Данные о требованиях и их изменениях делаются доступными для проекта.

С точки зрения управления это значит, что сначала рекомендации первой очереди становятся как минимум хорошими практиками (смотри шаблон ХП), затем при необходимости они регламентируются, и только потом вы начинаете институцинировать в команде рекомендации второй очереди.

«Сделано в целевой СУ» — графа с указанием того, какие конкретно действия надо выполнять для реализации рекомендации (ХП) или дополнительных, расширяющих ХП.

Графы «Сделано в целевой СУ» и «Сделано в СУПр» нужны для того, чтобы акцентировать внимание на минимально необходимых практиках («Сделано в целевой СУ») и обеспечивающих/вспомогательных практиках («Сделано в СУПр»).

Самое сложное в применении этого паттерна — корректно перевести текст CMMI с английского на русский язык, так как буквальный перевод не подойдет

в силу того, что потеряются некоторые языковые и смысловые нюансы. В идеале, если команда сможет, стоит работать с этим паттерном на английском языке. Если же это невозможно, то нужно провести несколько сессий обсуждения получившегося перевода практик СММИ с целью формирования одинакового понимания смысла практик и рекомендаций у всех заинтересованных лиц. Это крайне важный этап в применении этого паттерна.



---

Вторая важная особенность паттерна — это управление, основанное на изменениях. Даже если ваша компания находится на втором или третьем уровне зрелости, когда сама СММИ не требует и не подразумевает снятие метрик, я рекомендую как можно раньше начать применять эту часть паттерна. Как минимум это позволит вам понять, какие метрики нужны в принципе, и задуматься о том, где вы их будете брать. А значит, и какие инструменты вам нужны.

---

Далее приведена табл. 5.8 с примером анализа метрик и измерений.

Графы «О чем можно судить» и «О чем может говорить» по сути являются целеполаганием для всех дальнейших рассуждений. Иначе говоря, применяя этот паттерн, вы сначала понимаете, какие цели ставите перед собой и что для вас сейчас важнее, а что менее важно. Затем определяете формулу расчета и составляющие формулы. И отсюда приходите к метрикам и измерениям, которые надо уметь делать, а значит, к требованиям к инструментам и средствам автоматизации деятельности.

На сайте книги ([http://bit.ly/pmway\\_projectmetrics](http://bit.ly/pmway_projectmetrics)) вы найдете довольно подробный список возможных метрик и пример анализа, проведенного в одном из проектов.

Для изучения модели СММИ и работы с ней я рекомендую использовать *wibas* СММИ Browser, доступный по адресу: [http://bit.ly/pmway\\_cmimi](http://bit.ly/pmway_cmimi) или <http://www.cmmi.de/#el=CMMI-DEV/0/HEAD/folder/Processcollection.00012> — тут в интерактивном режиме очень наглядно и удобно представлен СММИ.

## 5.6. Паттерн IDEAL



---

В моей практике было несколько серьезных проектов по совершенствованию и построению процессов разработки ПО в компании в соответствии с моделью зрелости СММИ того или иного уровня. И на сегодняшний день многие владельцы бизнесов и компаний знают про модель СММИ.

И немногие знают про модель IDEAL. Это незаслуженно. По моему мнению, модель IDEAL SEI заслуживает не меньшего внимания с точки зрения построения процессов разработки ПО в компании и сама по себе является хорошим паттерном управления.

---

Общая схема жизненного цикла паттерна IDEAL приведена на рис. 5.20.

Таблица 5.8. Пример анализа метрик и измерений

№	Тип	О чем можно судить	О чем можно говорить	Параметр	Формула расчета	Части формулы	Определение	Пример	Какие системы развивать	Что еще развивать
1	Эффективность	Об общей эффективности создания кода всей командой	Стали ли мы более эффективно разрабатывать код как проектная команда, выходящая за пределы проектной активности?	Software development efficiency	A/B	A	Сколько человеко-часов мы тратим на разработку 1 KSLOC?	0,5	WAS	Классификатор проектов
				Software development efficiency	A/B	B		10		
	Эффективность	Об общей эффективности создания кода всей командой	Стали ли мы более эффективно разрабатывать код как проектная команда, выходящая за пределы проектной активности?	Software development efficiency	A/B	B	Product Size Added финальный в KSLOC. Размер в logical KSLOC оригинального (то есть созданного или модифицированного) исходного кода, который передается заказчику в качестве результата проектных активностей по объекту измерений. Никакой технологический или вспомогательный код, а также рабочие альтернативные версии кода, не подлежащие передаче заказчику, не учитываются	20	CodeSize	

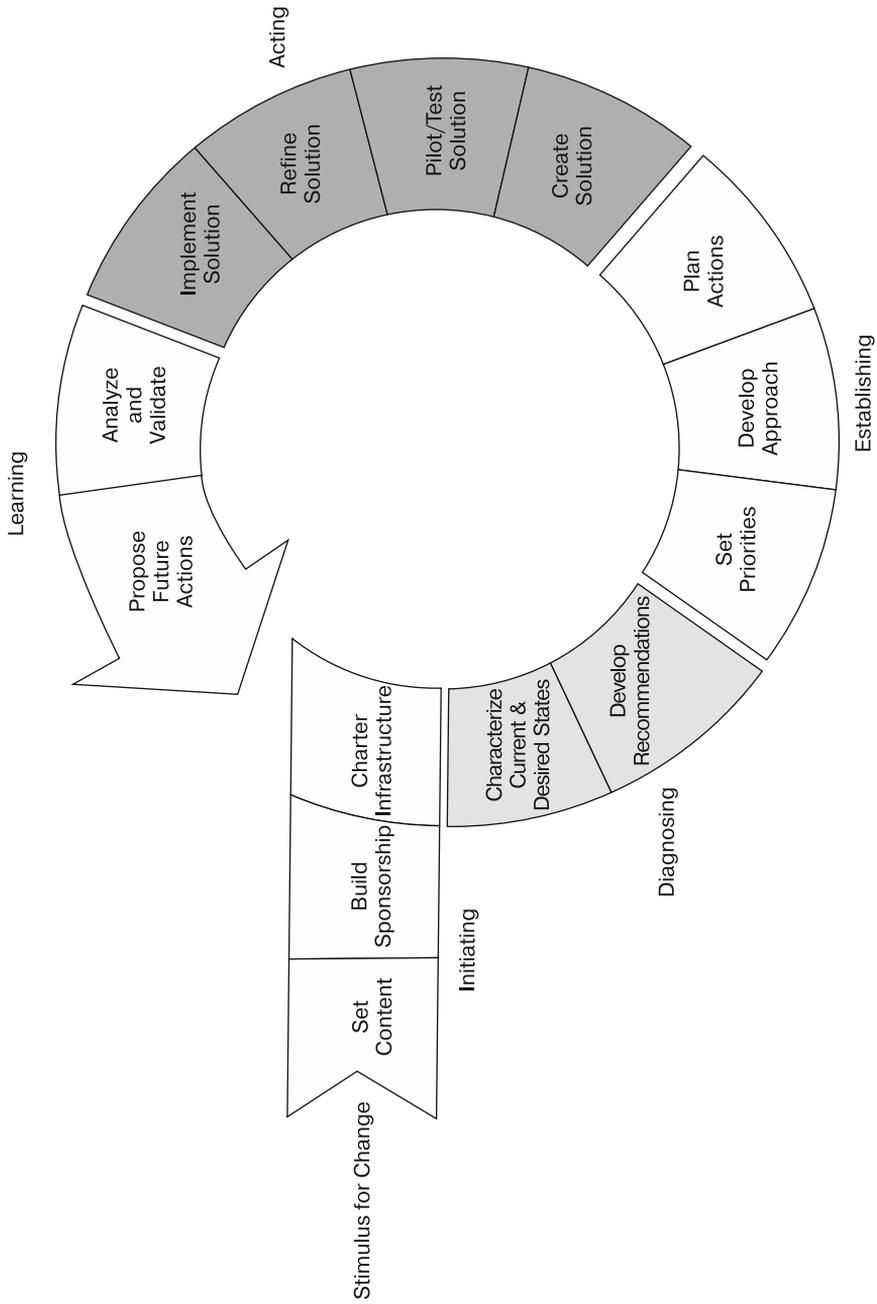


Рис. 5.20. Цикл функционирования программы совершенствования процессов

Описание фаз цикла IDEAL я привожу в табл. 5.9, которая может использоваться как паттерн по управлению проектами в довольно широком смысле.

**Таблица 5.9.** Описание фаз паттерна IDEAL

Фаза	Цель фазы
<b>1</b>	<b>2</b>
Initiating (инициация)	<p>Понять потребности в стимулах для изменений и основную причину их возникновения.</p> <p>Проанализировать окружающую обстановку (контекст).</p> <p>Заручиться поддержкой администрации (топ-менеджмента) при выполнении программы.</p> <p>Определить готовность к выполнению работ.</p> <p>Определить и установить необходимые требования и знания для старта.</p> <p>Заручиться разрешением администрации на создание инфраструктуры для проведения работ.</p> <p>Спланировать и подтвердить следующие шаги</p>
Diagnosing (диагностика)	<p>Понять, каково существующее состояние продукта, и определить его соответствие текущим бизнес-целям и ожиданиям.</p> <p>Определить желаемое состояние продукта.</p> <p>Для стратегического планирования собрать информацию о сильных сторонах продукта и возможностях совершенствования его технических и архитектурных характеристик.</p> <p>Выбрать эталонную модель разработки и управления.</p> <p>Разработать подходы к достижению желаемого состояния продукта.</p> <p>Определить стартовые значения для дальнейшего измерения степени совершенствования продукта.</p> <p>Обеспечить вовлечение в работу всех сотрудников – от руководителей команд до исполнителей</p>
Establishing (разработка, планирование)	<p>Разработать стратегический план.</p> <p>Интегрировать в стратегический план знания и информацию, полученные на предыдущих фазах.</p> <p>Установить приоритеты активностей в разработанных подходах.</p> <p>Согласовать стратегический план с активностями, связанными с разработкой других продуктов, а также с проектами, связанными с повышением качества работ и улучшения процессов.</p> <p>Обеспечить соответствие плана целям и миссии выпуска следующей версии продукта.</p> <p>Разработать краткосрочные и измеримые цели, достижимые конкретной версией продукта.</p> <p>Создать план действия с приоритизацией действий. Этот план включает расписание, задания, вехи, точки принятия решений, ресурсы, ответственность, измерения, механизмы прослеживания, риски и стратегии их минимизации, а также любые другие элементы, требуемые организацией</p>
Acting (внедрение)	<p>Определить состав программы проектов и создать планы этих проектов.</p> <p>Интегрировать план действия с планами проектов.</p> <p>Разработать решения/продукт.</p> <p>Проверить разработанные решения/продукт в одном пилотном проекте.</p> <p>Уточнить разработанные решения/продукт на основании результатов пилотного проекта.</p>

Продолжение ⇨

Таблица 5.9 (продолжение)

1	2
	Внедрить уточненные решения/продукт через пилотные проекты. Контролировать и поддерживать клиентов, переведенных на новый продукт. Принимать участие в необходимых активностях процессной группы и комитета по продуктам (планирование, выбор проектов, оценка результатов и т. д.)
Learning (оценка)	Собрать результаты выполнения предыдущих фаз. Убедиться в достижении желаемого состояния продукта, определенного на фазе инициации. Достичь наглядности в оценке качества продукта и соответствия целям. Получить поддержку администрации для дальнейшего развития продукта. Установить цели для следующего уровня. Провести выученные уроки и выявить хорошие практики

Мне кажется, что табл. 5.9 не нуждается в дополнительных пояснениях. В целом можно сказать, что данный подход в чем-то пересекается с широко известным PDCA. Но, на мой взгляд, IDEAL более органичен и полезен с точки зрения развития продукта и управления развитием продуктов в целом.



IDEAL также отлично подходит как паттерн управления программой проектов по совершенствованию процессов в компании. В том числе и по модели CMMI.

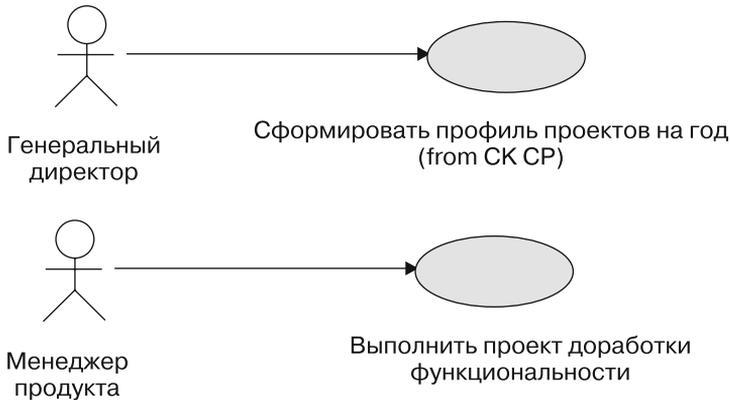
Кроме рассмотренных в этой книге IDEAL и CMMI, я рекомендую найти следующие методы и модели: «Циклическая модель ПРУР (принципов разработки управленческих решений)», «Модель ПРУР по Карпову», «Модель ПРУР по Минцбергу», «Модель принятия решений Врума – Йеттона» и познакомиться с ними.

## 5.7. Паттерн «Система управления производством ПО (СУПр)»

Как я уже упоминал, в первой книге я довольно подробно рассказывал про опыт построения процессов с использованием модели CMMI (DAR, концептуальная модель СУПр), так что не хочу повторять то, что уже было изложено, однако совсем пройти мимо этого паттерна было бы неправильно. Поэтому изложу здесь лишь некоторые основные тезисы, а более детально вы можете изучить его в первой книге.

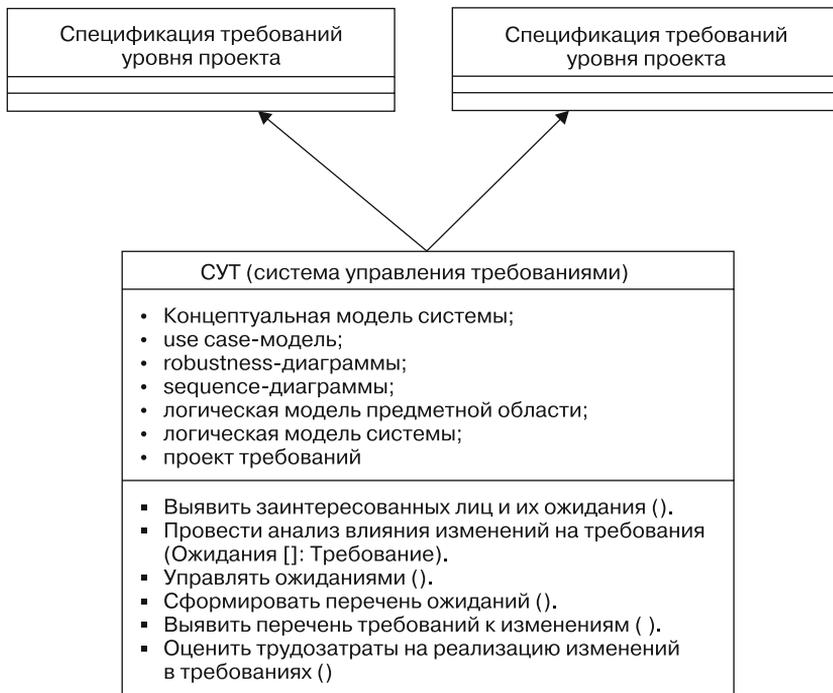
Паттерн СУПр по сути своей есть применение практик разработки ПО для построения управленческих процессов и, как следствие, формирования и управления средой проекта.

- ◆ Моделируем концептуальную модель предметной области СУПр (рис. 5.21).
- ◆ Выявляем варианты использования СУПр.



**Рис. 5.21.** Паттерн СУПр. Бизнес-кейс-модель

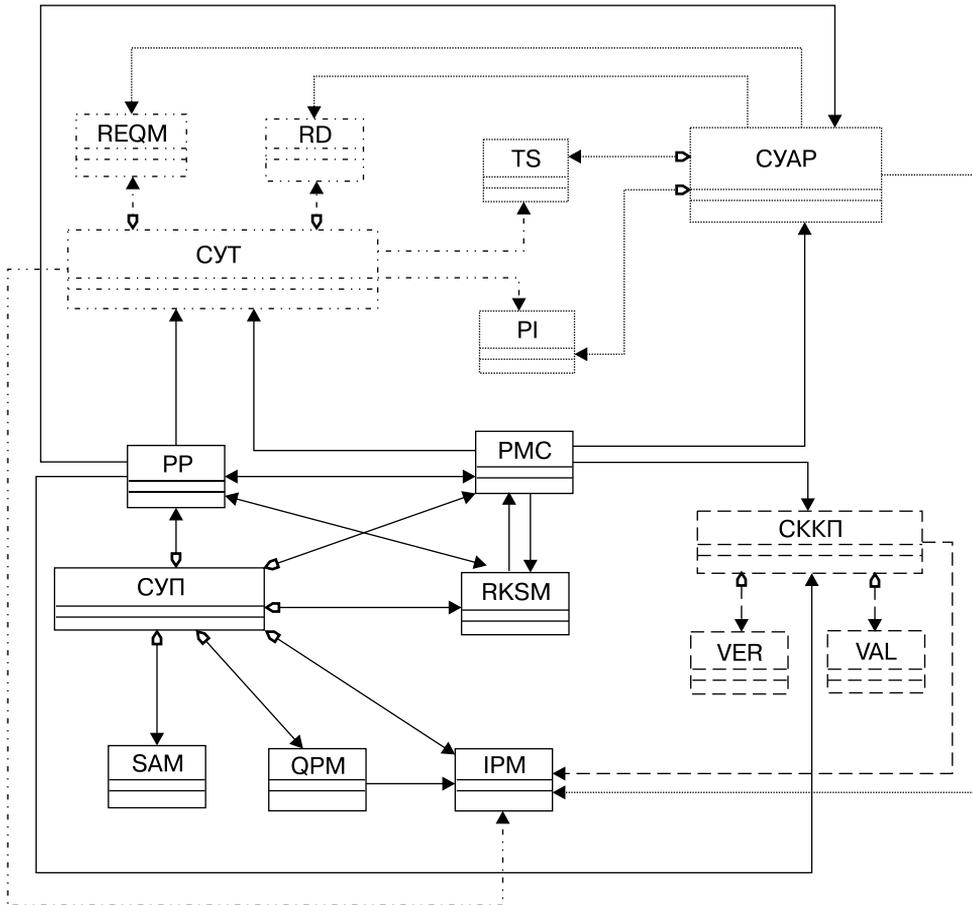
- ◆ Описываем потоки взаимодействия СУ СУПр, получая в качестве классов СУ, в качестве атрибутов таких классов – артефакты СУ, а в качестве методов – хорошие практики и рекомендации (рис. 5.22).



**Рис. 5.22.** Паттерн СУПр. Логическая модель

При этом потоки взаимодействия будут, по сути, высокоуровневым описанием производственных процедур.

Рассмотрим теперь подробнее концептуальную модель СУПр. В качестве компонентов СУ в данной модели мы выбрали соответствующие практики процессных областей СММІ (рис. 5.23).



**Рис. 5.23.** Концептуальная модель СУПр:

СУТ — система управления требованиями (REQM — Requirements Management, RD — Requirements Development); СУАР — система управления архитектурой и разработкой (TS — Technical Solution, PI — Product Integration); СУПр — система управления проектом (PP — Project Planning, RSKM — Risk Management, PMC — Project Monitoring and Control, IPM — Integrated Project Management, QPM — Quantitative Project Management, SAM — Supplier Agreement Management); СМККП — система контроля качества продукта (VER — Verification, VAL — Validation)

Как видно из концептуальной модели, СУПр — это интегрирующая система управления в рамках всей СУПр. Хорошим примером является процедура управления изменениями, которая определена в СУПр, но на которую ссылаются и которая частично реализуется, в том числе и через процедуры других СУ.

СМК является хранилищем всей регламентной базы предприятия и не показана на диаграмме взаимодействия СУ. С СМК взаимодействуют все СУ, так как все процедуры, рабочие инструкции, методологии всех СУ располагаются в СМК и после модификации в рамках совершенствования процессов в любой СУ обновляются в СМК.

Система управления изменениями конфигурации (СУИК) также используется всеми СУ для управления их объектами конфигурации и информационными сущностями. СУИК реализует унифицированный подход к управлению версиями и конфигурациями.

Процедуры СУПр строятся по стандарту РМВОК, который реализует все практики модели СММІ, учитывая, таким образом, и все практики названных процессных областей.



---

Далее приведено концептуальное описание основных потоков деятельности СУ в рамках СУПр.

---

СУПр выполняет стратегическое, тактическое и операционное планирование, формирует портфель проектов и инициирует их выполнение. В стратегическом и тактическом планировании активно участвуют СУАР и СУТ.

СУТ анализирует высокоуровневые требования на всем этапе планирования, оценивает трудоемкость работ по созданию требований. СУАР и СККП обеспечивают оценку трудоемкости и участвуют в управлении изменениями в СУПр.

СУПр принимает управленческие решения и достигает соглашения с заказчиком по показателям рентабельности, срокам выпуска продукта и объему ожиданий заказчика, которые он хочет видеть реализованными к этому сроку. СУТ активно участвует в управлении ожиданиями заказчика, СУАР активно участвует в управлении архитектурно значимыми запросами заинтересованных лиц и выборе технологий реализации продуктов.

СУПр определяет тип проекта и жизненный цикл разработки ПО. СУПр инициирует проект и запускает его выполнение. СУТ в соответствии с жизненным циклом создает требования, отталкиваясь от запросов заинтересованных лиц, требований бизнеса и технических требований, определенных в СУПр на этапе стратегического, тактического и операционного планирования. СУАР в соответствии с жизненным циклом разрабатывает архитектуру и активно участвует в выявлении архитектурных требований и управлении ими.

Разработанные требования и архитектура проходят Peer Review (одноранговые ревью) в проектной команде. После этого согласования требования и архитектура верифицируются в СККП.

СККП создает полный набор тест-планов и тест-кейсов, которые проходят ревью в СУТ и СУПр. СУАР создает продукт, который подается на вход СККП для тестирования. СККП в соответствии с разработанными тест-планами и кейсами тестирует продукт.

СУП обеспечивает интеграцию процессов всех технологических СУ, отвечает за планирование и авторизацию работ и управление ими и за выпуск конечного продукта или его очередной версии, которая поступает в отдел маркетинга и продаж.

СМК выполняет аудит качества процессов всех СУ. Записи о качестве и аудитах хранятся в СМК.

СУП закрывает проект и подводит итог – формулирует выученные уроки.

СМК совершенствует процедуры и артефакты всех СУ. СМК представляет сводный анализ использования СУ в Steering Committee (управляющий комитет) и группу SERG (группу совершенствования процессов инженерии программного обеспечения). СУП поставляют в эти же органы результаты выученных уроков и обновляет базу активов компании.

## 5.8. Резюме главы

В этой главе мы рассмотрели общие и специальные паттерны разработки ПО, которые предназначены для формирования проектной среды и управления ею.



---

Наиболее точно концепцию среды проекта отражает пример с садом в паттерне «Управление средой проекта». Если вы по каким-то причинам пропустили этот раздел, обязательно прочитайте его внимательнее.

---

Хочу также напомнить, что все паттерны, рассмотренные в разделе «Паттерны разработки ПО», могут применяться не только в проектах разработки ПО. Принципы и подходы, заложенные в этих паттернах, универсальны. Так, например, паттерн «Обеспечение качества и анализ рисков» может применяться даже для разработки проекта строительства дома, только для формирования правильного реестра рисков и работ, снимающих эти риски, нужно будет привлечь профессионалов в этой области.

В начале главы есть раздел «Сводная таблица паттернов формирования проектной среды», где можно быстро найти нужный паттерн для того, чтобы начать использовать его прямо сейчас.

# 6. Паттерны управления

---

---

## 6.1. Сводная таблица паттернов управления

В этом разделе приводится сводная таблица паттернов управления (табл. 6.1–6.3), которые рассматриваются в главе. Вы можете использовать этот раздел для быстрого поиска нужного паттерна, чтобы начать использовать его прямо сейчас.

### 6.1.1. Паттерны управления проектом

Таблица 6.1. Паттерны управления проектом

Паттерн	Кому нужен? Зачем нужен?	Организационный аспект	Поведенче- ский аспект	Профиль мене- джера	Стра- ница
1	2	3	4	5	6
«Экстремальная разработка»	Управленцу, команде. Надо, чтобы итерация состоялась как можно раньше. Нет времени даже на создание расширенного устава проекта. Надо, чтобы разработчики как можно раньше начали работать над прототипом	Шаблон постановки задачи заполняется и в кратчайшие сроки отдается на прототипирование. Прототип создается и обсуждается с заинтересованными лицами. Уточняется постановка, при необходимости создаются более «тяжелые» артефакты – спецификации, сценарии, макеты и дизайны интерфейсов	Scrum Team	Релиз-менеджер, менеджер проекта	
«Чек-листинг»	Команде. При выполнении какого бы то ни	Чек-листы создаются в наиболее критических или сложных процедурах	Сотрудники понимают и принимают	Администратор проекта,	

Продолжение ➔

Таблица 6.1 (продолжение)

1	2	3	4	5	6
	было процесса сотрудники заполняют чек-листы, или листы самопроверки, где перечислены основные выученные уроки, которые легли в основу хороших практик	производственных процессов. Сотрудники заполняют чек-листы. Периодически требуется выборочный контроль заполнения чек-листов	важность практик ХП, ВУ и чек-листинга	релиз-менеджер, менеджер проекта	
«Смартинг»	Команде, управленцу. Четкие, измеримые, конкретные, ограниченные во времени задачи	При постановке задач уделяется время тому, чтобы они были четкими, измеримыми, конкретными, ограниченными во времени	Сотрудники не боятся задавать вопросы, если задача не smart	Администратор проекта, релиз-менеджер, менеджер проекта	
«Расширенный устав проекта»	Управленцу, команде проекта. Если проект реализуется в жестких условиях ограничения по срокам и стоимости управленческих активностей, то предыдущий паттерн вам, скорее всего, не подойдет. Для проектов, управляемость которых должна быть высокой, а затраты на создание управленческих артефактов минимальными, я рекомендую пользоваться этим паттерном	Устав проекта, расширенный разделом «Управление проектом». Управление проектом берется за основу как шаблон. В итоговом документе оставляют только те практики и информацию, которые будут реально использоваться в проекте	Управленец должен владеть РМВОК или четко представлять себе, что такое устав проекта, ПУП и основные процессы управления проектом	Релиз-менеджер, менеджер проекта, менеджер программы проектов	
«Ромашка Ананьина – Минцберга»	Управленцу, бизнесу. Этот паттерн хорош для анализа инноваций и управления ими – для этого надо применить его ко всем тактическим инициативам, которые возникают по каждому направлению.	Формируется таблица с пятью графами: <ul style="list-style-type: none"> <li>• «Цель»;</li> <li>• «Роли и ответственность»;</li> <li>• «Приемы и навыки»;</li> <li>• «Стандарты»;</li> <li>• «Планы».</li> </ul> После заполнения таблицы выполняется согласование этого высокоуровневого видения	Данный паттерн отлично дополняет паттерн «Управление средой проекта». От управленца требуется понимание связи проекта	Менеджер проекта, менеджер программы проектов, менеджер продукта	

1	2	3	4	5	6
	В случае взаимодействия бизнеса и IT этот паттерн особенно хорош для быстрого начала этого диалога. Он, собственно, и позволяет начать диалог	заинтересованными лицами	со средней проектов в компании		
«Управление неопределенностью проекта»	Управленцу, команде. Идея паттерна состоит в том, что по мере узнавания среды проекта и самого проекта и управления ими вы постоянно снимаете неопределенности и смотрите, какие еще неопределенности надо снять	На основании шаблона управления неопределенностью проекта последовательно снимается неопределенность	Вся команда вовлечена в управление неопределенностью. Заинтересованные лица являются частью команды и как минимум принимают участие в части целей, ролей, ответственности и стандартов	Менеджер проекта, менеджер программы проектов, менеджер продукта	
«СУП на РМВОК»	Управленцу. Построить архитектуру (процедуры, их взаимодействие, действия и проектные артефакты)	Паттерн в части таблиц процедур и артефактов берется в качестве шаблона для проектирования архитектуры СУП вашей компании (проекта/программы проектов) и принятия решения о том, какие практики и артефакты вам нужны, а какие — нет	Управленец должен самостоятельно изучить РМВОК и как минимум еще какую-то методологию управления проектами (СОВНЕТ, «экстремальное управление проектами» и т. п.)	Менеджер проекта, менеджер программы проектов	
«СУП на воде, или Критические принципы управления»	Управленцу, команде. Универсальные и только наиболее критические принципы управления проектами	Принять принципы в работу, в ходе выполнения проекта находя ответы на вопрос, как конкретно это делать	Scrum Team. Желание и умение находить ответы на сложные вопросы.	Администратор проекта, релиз-менеджер, менеджер проекта	

Продолжение ⇨

Таблица 6.1 (продолжение)

1	2	3	4	5	6
	и проектными коммуникациями, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев		Стремление развиваться и изучать методологии управления проектами		
«Зажарка в СУП, или Универсальные принципы управления проектом»	Управленцу, команде. Универсальные принципы управления проектами, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев	Принять принципы в работу, в ходе выполнения проекта находя ответы на вопросы, как конкретно это делать	Scrum Team. Желание и умение находить ответы на сложные вопросы. Стремление развиваться и изучать методологии управления проектами	Релиз-менеджер, менеджер проекта	
«Стабилизация ПО»	Управленцу, команде. В ситуации, когда вы не можете позволить себе полноценное управление проектом, можете применить паттерн «Экстремальная разработка» совместно с данным паттерном для более или менее эффективного управления проектом «на коленке». Своевременное обнаружение и мониторинг картины стабилизации ПО с помощью этого паттерна позволит своевременно перенаправить ресурсы разработчика на решение проблем,	Таблица багов заполняется после поставки каждой версии ПО. В таблице паттерна видно, какой процент багов повторялся из версии в версию, чему может быть две причины: или это крайне сложный баг, на который требуются значительное время и силы разработчиков и они не успевают его исправить в рамках итерации выпуска версии (напомню, это всего неделя), или причина бага неясна и разработчик не знает, что именно исправлять. И в том и в другом случае это зоны риска продукта. И вы, как менеджер, должны такие риски выявлять и управлять ими	Команда не просто заполняет таблицу, она анализирует то, что получилось, пытается понять причины именно такой картины стабилизации, выявить риски и, возможно, пересмотреть приоритеты. Любой участник команды может выступить с предложениями	Администратор проекта, релиз-менеджер, менеджер проекта	

1	2	3	4	5	6
	<p>наиболее критических с точки зрения начала внедрения продукта. Этот же паттерн используется для регрессионного тестирования – тестируются баги критические или те, которые крайне долго устранялись чаще, чем все прочие</p>				
<p>«С ног на голову, или ПМП ПСР (программа и методика проведения приемосдаточных работ)»</p>	<p>Управленцу со стороны заказчика. Может применяться как фактор, «вытягивающий» представление о системе в случае, когда подрядчик не настроен на конструктив, проект в критическом пике и подрядчик не хочет ничего дополнительно вам рассказывать и создавать какие-то дополнительные спецификации или диаграммы. Может применяться не только в ситуации «с ног на голову», но и при нормальной, прямой и последовательной разработке. Я бы рекомендовал в качестве приложения к договору оформлять примерно такой ПМП ПСР</p>	<p>Создается рабочий документ ПМП ПСР как инструмент управления проектом.</p> <p>Все правки в документе между версиями <math>i.0</math> (где <math>i = 1, 2, \dots</math>) вносятся в режиме управления изменениями до согласования заказчиком очередной версии документа. Исправления в документ вносятся и исполнителем, и заказчиком. Изменения в документе принимаются только заказчиком.</p> <p>По мере работы над документом снимаются риски, открытые вопросы, корректируется раздел «Программа и методика проведения...».</p> <p>После согласования документа на его основе создается календарный план-график проведения приемочных испытаний, в соответствии с которым проводятся подготовка к приемке и приемка разработанного решения</p>	<p>Раздел «Контроль качества выполнения проекта» был крайне полезен для того, чтобы начать разговор о том, что не видно достаточных свидетельств того, что в проекте должное внимание уделялось качеству продукта. Не терять терпения и стараться как можно спокойнее объяснять, зачем нужны такие тесты и именно такие проверки, что это дает и какие риски заказчика это снимает</p>	<p>Менеджер проекта, менеджер программы проекта, менеджер продукта</p>	

## 6.1.2. Паттерны управления функциональным отделом

Таблица 6.2. Паттерны управления функциональным отделом

Паттерн	Кому нужен? Зачем нужен?	Организационный аспект	Поведенческий аспект	Профиль менеджера	Страница
1	2	3	4	5	6
«Фундамент отдела»	<p>Руководителю функционального отдела.</p> <p>Относится к паттернам формирования среды.</p> <p>От того, насколько хорошо применен и применен ли вообще этот паттерн в отделах, зависят общая корпоративная культура и то, насколько в принципе жизнеспособны любые производственные процессы организации</p>	<p>Последовательно-параллельно выполняются работы и паттерны:</p> <ul style="list-style-type: none"> <li>• разрабатываются базовые правила работы;</li> <li>• создается стратегия/концепция развития отдела;</li> <li>• формируется база знаний (БЗ) отдела (ХП и ВУ);</li> <li>• WAS;</li> <li>• разрабатывается план персонального развития сотрудника;</li> <li>• разрабатывается план курсов и тренингов;</li> <li>• создается система менеджмента качества (QMS) отдела.</li> </ul> <p>Сделав это, вы получите актуальный на дату выполнения фундамент вашего отдела. Дальше вы постоянно, ежедневно поддерживаете этот фундамент в актуальном для достижения целей отдела состоянии</p>	<p>Важно довести до сотрудников, что данные WAS нужны вам не для назначения, а для получения прозрачной картины и повышения точности планирования работы отдела в целом.</p> <p>Командный дух.</p> <p>Вы — лидер, наставник, учитель.</p> <p>Идеально, когда каждый ваш подчиненный будет считать своей (!) заслугой то, как он самостоятельно (!) вырос профессионально</p>	<p>Руководитель функционального отдела</p>	
«Удаленное управление»	<p>Руководителю функционального отдела.</p> <p>Необходимо управление распределенной или полностью удаленной командой</p>	<p>WAS.</p> <p>Чек-листинг.</p> <p>Смартинг.</p> <p>Специфика проведения виртуальных встреч.</p> <p>Адаптированные базовые правила работы</p>	<p>Открытость и коммуникабельность участников команды;</p> <p>доступность;</p> <p>желание достигать взаимопонимания</p>	<p>Руководитель функционального отдела, менеджер проекта</p>	

1	2	3	4	5	6
«Маслоу по Фрейду»	<p>Руководителю функционального отдела.</p> <p>Сформировать у сотрудников нужную модель поведения.</p> <p>Корректно довести до каждого существующие правила.</p> <p>Научить работать в коллективе</p>	<p>Паттерн показывает эффективное влияние на мотивацию ИД посредством воздействия на эго и суперэго применительно к пирамиде Маслоу и информационному полю руководителя функционального отдела.</p> <p>Вы постоянно занимаетесь формальным доведением и закреплением правильных норм и морали.</p> <p>Как учитель и наставник неформально, не в форме готовых выводов, а на примерах, через разбор ситуаций добиваетесь правильных «мантр» в головах ваших подчиненных</p>	<p>Честность и прозрачность отношений.</p> <p>Перед правилами равны все.</p> <p>Быть примером</p>	<p>Руководитель функционального отдела</p>	
«Лидерство»	<p>Руководителю функционального отдела.</p> <p>Сборник некоторых хороших практик с описанием определенного лидерского подхода к формированию зрелой самоуправляемой группы</p>	<p>Анализ уровня зрелости группы/отдела.</p> <p>Четыре стили лидерства применительно к четырем уровням зрелости группы.</p> <p>Вовлечение, мотивация.</p> <p>Премирование, депремирование.</p> <p>Качество трудовой жизни</p>	<p>Не скупиться на похвалу.</p> <p>Понять, что именно помешало выполнить задачу.</p> <p>ВУ как «разбор полетов».</p> <p>Это можно сделать еще лучше.</p> <p>Прозрачная денежная мотивация рычагов воздействия на ИД (паттерн «Маслоу по Фрейду»)</p>	<p>Руководитель функционального отдела, руководитель разработки ПО</p>	
«Саморазвитие и личный бренд»	<p>Руководителю.</p> <p>Самосовершенствование и развитие</p>	<p>Пирамида развития навыков менеджера.</p> <p>Личностный бренд</p>	<p>Предельная честность перед самим собой</p>	<p>Руководитель функционального отдела, руководитель разработки ПО</p>	

### 6.1.3. Паттерны управления подразделением разработки ПО

Таблица 6.3. Паттерны управления подразделениями разработки ПО

Паттерн	Кому нужен? Зачем нужен?	Организационный аспект	Поведенче- ский аспект	Профиль мене- джера	Стра- ница
1	2	3	4	5	6
«Мини-мальный офис управления проектами»	<p>Руководителю подразделения разработки ПО.</p> <p>Быстро с чего-то начать с точки зрения превращения хаоса в процессах и головах.</p> <p>Втиснуть максимум теории практической направленности за минимальное время.</p> <p>Дать фреймворк, который позволит сеньор-менеджерам сразу начать работать.</p>	<p>Упрощенный фреймворк по типу PDCA.</p> <p>Акцент с Do смещен в Check.</p> <p>PreCAT и CAT.</p> <p>Пересекается с паттерном «Экстремальная разработка»</p>	<p>Сотрудники должны быть активны и открыты к новому, стремиться к самостоятельному получению знаний.</p> <p>Любые более «правильные» подходы, чем в этом фреймворке, приветствуются.</p> <p>Вы сами строите свой фреймворк. Улучшайте то, что вам дано</p>	Руководитель разработки ПО	
«Лидерство на уровне департамента»	<p>Руководителю подразделения разработки ПО.</p> <p>Сборник хороших практик</p>	<p>Клуб IT-менеджеров.</p> <p>Архитектурные совещания.</p> <p>Доска для всех.</p> <p>Приходите с решениями</p>	<p>Сотрудники должны быть активны и открыты к новому, стремиться к самостоятельному получению знаний.</p> <p>Любые более «правильные» подходы приветствуются.</p> <p>Сеньор-менеджер – это самостоятельная «боевая» единица, имеющая свою зону полномочий и ответственности</p>	Руководитель разработки ПО	

1	2	3	4	5	6
«Мотивационная схема»	Руководителю подразделения разработки ПО. В качестве примера возможного подхода и инструмента	Определить KGI. Определить KPI. Определить правила их применения и трактовки спорных ситуаций	Схема мотивации направлена на то, чтобы стимулировать выполнение правильной работы по нужным для компании направлениям. Невыплаченный бонусный фонд уходил в фонд поощрения лучших сотрудников по итогам года. Для популяризации активно использовать паттерн «Информационный бюллетень». Все должно быть максимально четко, ясно и прозрачно	Руководитель разработки ПО	
«Стратегия и планирование в СУПр»	Руководителю подразделения разработки ПО. Подход к стратегическому и тактическому планированию. Увязка стратегии с тактикой и операционной мощностью подразделения	Трехуровневое планирование. Постепенное движение от стратегических целей в областях «назначение», «рынок», «архитектура» и «технология продукта» через тактическое планирование и формирование бизнес-видения, технологического видения и неделимых подмножеств коммерчески выгодных функциональностей продуктов (фич). Формирование портфеля проектов. Создание операционного плана, включающего	Доступ ко всей нужной информации. Вовлеченность бизнес-дизвиона. Рабочие руки аналитика. Экспертиза, выполняемая ведущими архитекторами	Руководитель разработки ПО	

Таблица 6.3 (продолжение)

1	2	3	4	5	6
		в себя усиление возможностей компании для достижения требуемых конкурентных преимуществ			
«Управление портфелем проектов»	Руководителю подразделения разработки ПО. Как формировать портфели проектов. Почему и когда нужно, чтобы портфелей проектов было несколько? Увязка стратегии с вопросами управления портфелями проектов	Состав портфеля. Управление портфелем «привлекательность – управляемость». Этапы управления портфелем проектов. Специфика управления проектами в зависимости от портфелей	Требуется глубокое знание методологий разработки вопросов ЖЦ продукта. ЖЦ проекта. Требуются знание и понимание паттернов «СУП на РМВОК», «СУП на воде» и «Зажарка в СУП»	Руководитель разработки ПО	

## 6.2. Еще раз о проекте

Прежде чем мы перейдем к паттернам, давайте снова немного поговорим об известных, скорее всего, всем теоретических вещах. Я буду предельно краток и постараюсь пользоваться самыми простыми формулировками.



Проект — это работы, планы, мероприятия и другие задачи, направленные на создание нового продукта (устройства, работы, услуги).

Выполнение проекта составляет *проектную деятельность*, которая включает:

- ◆ проведение управленческих мероприятий (проектное управление). Достигается на основе использования, в частности, принципов и методов *управления проектом*, являющегося частью системы менеджмента предприятия, универсальной для решения разных производственных задач;
- ◆ решение специализированной задачи;
  - ◆ разработку продукции для заказчика. Продуктами проекта могут быть:
    - результаты маркетинговых исследований (маркетинг);
    - проектно-конструкторская документация (управление проектированием). Комплект такой документации называется *проектом*. Он предназначен для создания разработанного устройства, его эксплуатации, ремонта и ликвидации, а также проверки или воспроизведения промежуточных и конечных решений, на основе которых он был разрабо-

- тан (обратите внимание на то, что значение слова «проект» в управленческой и научно-технической деятельности различаются);
- технологическая документация (управление производством);
- программное обеспечение (управление проектами) и т. д.;
- ◆ решение внутренних производственных задач:
  - повышение качества продукции (управление качеством);
  - повышение эффективности организации труда (управление персоналом);
  - оптимизация финансовых потоков (финансовый менеджмент) и др.

## 6.3. Паттерны управления проектом

В этой главе мы поговорим об управлении проектом и о паттернах управления, которые могут успешно применяться для управления проектами.

Как ни странно, но целевой аудиторией паттернов являются не только менеджеры проекта. Если вы работаете над каким-либо проектом, то вы — участник проектной команды, так что и знание названных паттернов, и умение применять их положительно скажется на результате проекта в целом, а значит, и на результате вашей работы в частности.

Вы можете применять паттерны все или по отдельности — в зависимости от того, что это за проект, в какой организации он выполняется, каков уровень зрелости проектной команды и заказчика, вы можете комбинировать и видоизменять эти паттерны. Моя цель — подсказать и показать определенный путь, который я прошел сам, не говоря, что только он является правильным, но утверждая его эффективность, подтвержденную практикой.

Если вы прочитали главу 1 книги, то какая-то информация, возможно, покажется вам знакомой. Это так и есть. Поскольку я стараюсь описывать паттерны как условно независимые подходы, то в некоторых случаях действительно показываю суть вещей с разных сторон. Сделано это во многом потому, что читатель вполне может начать изучение книги с конкретного паттерна, так что я стремлюсь к тому, чтобы информация о паттерне была достаточной для того, чтобы он мог начать его применять.

### 6.3.1. Паттерн «Экстремальная разработка»



В качестве вступления я расскажу о реальном проекте из своей практики.

Вводные: стартап, инновационная идея успешно продана инвестору и получены инвестиции под реализацию идеи. Бюджет на разработку ПО — то, что называется «впритык». Сроки — «вчера». Концептуальная архитектура решения предполагает наличие трех специализированных АРМ, один из которых представляет собой довольно сложный инструмент разметки географических карт и привязки к разметке разнородных объектов в связанной логике предметной области.



В этом случае важно, чтобы первая итерация состоялась как можно раньше. Нет времени даже на создание расширенного устава проекта. Нужно, чтобы разработчики как можно раньше начали трудиться над прототипом. А уже потом можно вернуться к паттерну «Расширенный устав проекта», что и было сделано в проекте, о котором я говорю.

Или другой пример. Вы управляете функциональным отделом — у вас нет проектной разработки или проекты настолько малы и уникальны, что, по сути, это выглядит как операционная деятельность с ежедневным решением новых и новых задач.

В этом случае я рекомендую использовать такой шаблон постановки задачи.

### **Постановка задания**

**«[наименование задачи, используемое в переписке]»**

**для подразделения: [наименование структурного подразделения]**

#### **1. Бизнес-обоснование.**

Информация этого раздела должна подробно разъяснять руководителю исполняющего подразделения цели, которые будут достигнуты при его реализации, и то, зачем это нужно. При этом здесь не должны обозначаться конкретные этапы решения и способы достижения результатов.

#### **2. Оценка и обоснование оценки экономической эффективности.**

Это конкретная сумма, а также обоснование происхождения этой суммы (включая формулу, алгоритм ее получения или логику рассуждений).

#### **3. Бизнес-контекст использования решения.**

Это ответ на вопрос: «Кто, каким образом и с какой периодичностью будет использовать полученное решение?»

#### **4. Бизнес-ограничения и допущения.**

Это ответ на вопросы: «Какие из выявленных на начальном этапе факторов должны быть учтены при принятии задачи к исполнению? Какие из них могут быть обязательными, а какие — нет?»

#### **5. Ожидаемая заказчиком функциональность.**

Ответ на вопрос: «Что заказчик увидит/услышит/почувствует в итоге, после того, как его задача будет выполнена?» Здесь также могут быть представлены конкретный перечень функций и их краткое описание. Необязателен для заполнения.

#### **6. Предложения заказчика по реализации решения.**

Ответ на вопрос: «Как клиент сам понимает принцип решения задачи?» Необязателен для заполнения. Здесь же могут быть сценарии, картинки с интерфейсом, компонентные и архитектурные диаграммы. Для экономии времени эти артефакты можно нарисовать в Paint или карандашом на бумаге и встроить как фотографии.

## 7. Форма отчетности.

Здесь указывают формат, периодичность и адресатов, получающих уведомления о ходе выполнения (завершения) задачи. Прямо в текст ПЗ можно включать примеры отчетов, выполненные в Word, Excel и т. д.

Как видите, это очень простая форма, но тем не менее она дает ответы на многие вопросы, которые мы обсуждали в других паттернах.

Этот паттерн работает лучше, если вы используете совместно с ним принцип «Управляем пулей».

Название принципа мне подсказал Дуг ДеКарло в своей книге «Экстремальное управление проектами». Суть подхода в том, что если в классическом понимании управления проектами мы «выбираем цель, прицеливаемся и стреляем», то в экстремальном проекте сначала «стреляем», а затем «управляем полетом пули», ведем ее в цель.

Этот принцип отлично работает в стартапах и срочных, напряженных, с непонятным скоупом задачах, где нет времени и возможности ни уточнить вопросы, ни спланировать. Например, задача по срочному восстановлению работоспособности ПО в автотранспортном парке, которое команда, которой это поручили, ранее в глаза не видела, опыта установки и развертывания не имеет. Документации нет. Есть один специалист в сторонней компании, который год назад работал с этой системой. Есть один разработчик ПО в другой сторонней компании, который дорабатывал серверную часть системы, но мало что знает про АРМ.

Суть принципа заключается в том, что мы «стреляем», то есть намечаем первые конкретные шаги в виде простого списка с ответственными и сроками, а затем «управляем пулей».



---

Далее приведен пример применения такого принципа в моих письмах — постановках задачи. Обычно я использую цветовую гамму: серым цветом фона обозначаю выполненные шаги, зеленым — дальнейшие шаги, в книге же серый цвет будет заменен *курсивом*, а зеленый — подчеркиванием.

---

## Письмо 1

### Обсуждали/выяснили:

- ♦ в парке Z все развернуто и работает, со слов IT-специалиста в парке Иванова (информация получена от Петрова через Сидорова);
- ♦ образа диска для разворачивания серверов в парках у Company 1 нет.

### Дальнейшие шаги

#### 1. Встреча и подготовка к демонстрации.

А. Бойков созванивается и договаривается о встрече в 10:00 в парке.

- Встреча завтра, в 10:00, в парке Z. Всем сообщить свои паспортные данные Бойкову.
- Бойкову сообщить всем адрес и то, как добираться в парк Z.

- Б. Желязнов актуализирует все действующие внешние IP-адреса серверов «боевой» среды для возможных настроек паркового ПО.
2. Сценарий демонстрации.
- А. Писарев заканчивает и высылает документ с описанием сценария Иванову.
- Б. Иванов и Писарев анализируют возможность и невозможность демонстрации сценариев по каждому шагу сценария.
- В. Писарев готовит финальную версию сценариев демонстрации.
3. Информация по развертыванию.
- А. Иванов поднимает всю имеющуюся информацию: мануалы, логины и пароли к самим серверам, логины и пароли суперпользователей в АРМ, контакты по проекту, IP-адреса серверов в парках и т. п. и присылает адресатам письма.
- Б. То, что критично для возможного развертывания, Иванов как можно раньше присылает отдельным списком адресатам этого письма.
4. Снятие образа.
- А. *Перерве запросить образ серверов в Страну 2.*
- Б. *Желязнову совместно с Юниксутовым подготовиться к снятию образа сервера для дальнейших настроек:*
- *внешний CD с ПО Acronis;*
  - *инструкция по снятию образа.*
- В. Желязнов снимает образ завтра после развертывания и проверки того, что АРМ работают.

*С уважением, Перерва А. Д.*

Так был произведен «выстрел».




---

С этого момента начинается «управление пулей». РП (руководитель проектов) ментально разделяется на несколько частей, одна отвечает за текущую оперативку, вторая контролирует движение хотя бы в сторону цели, третья пытается придать текущему хаосу управляемость и предсказуемость или хотя бы прозрачность. Также РП «отращивает» несколько ментальных «щупалец», каждое из которых лежит на пульсе у соответствующего человека проектной среды.

---

РП к «управлению полетом пули» готов.

## **Письмо 2**

### Сделано

1. Включили парковый сервер (сервер был зашатдаунен).
2. Иванов проверил и восстановил работоспособность всех АРМ паркового ПО (см. ниже).

3. Иванов провел консультации по настройкам Желязнова.
4. Желязнов запустил утилиту снятия образов с сервера.

#### Достигнутый результат

1. АРМ 1 работает в штатном режиме.
2. АРМ 2 работает в штатном режиме.
3. АРМ 3 запустился. Но нет возможности выбора организации в выпадающем списке организаций. Так что АРМ НЕРАБОТОСПОСОБЕН.
  - А. Проверили настройки в АРМ администратора в соответствии с рекомендацией Писарева. Пользователь Nag имеет права на парк Z. Не помогло.
  - Б. Проверили работу при подключении «прямого» Интернета к этому АРМ (в штатном режиме не должно быть). Не помогло.
  - В. Перерва задал вопрос в Comranу 2 о возможных причинах неработоспособности.
4. АРМ 4 запустился. Проверить работоспособность выгрузки архивных данных невозможно, так как нет ни одного наряда (не работает АРМ 3).

#### Дальнейшие шаги

1. Желязнов 20.11 забирает в парке записанный образ сервера.
2. В случае неудачи при выполнении п. 1. Желязнов 20.11 привозит диски с сервера. Совместно с Юниксутовым снимают образ в компании. Желязнов отвозит диски обратно.
3. Желязнов по горячим следам оформляет записи в виде рабочих инструкций (РИ) по настройке парковой среды и согласует с Ивановым.
4. Перерва связывается с Головатым и пытается получить консультацию и выяснить возможные причины того, почему не работает.
5. Бойков уточняет возможный сценарий демонстрации, исходя из технологической последовательности, изложенной Писаревым.
6. Иванов присылает Перерве материалы, о которых говорили.

**Иван Иванов**, спасибо Вам за помощь и консультации.

*С уважением, Перерва А. Д.*

### **Письмо 3**

#### Статус на 20.11.2013 и уточненные дальнейшие шаги

1. Желязнов 20.11 забирает в парке записанный образ сервера – развернуть в НИСе среду не удалось.
2. Создание образов паркового сервера.
  - А. Желязнов договаривается с администратором Z о том, чтобы работать с сервером через TeamViewer, – доступ есть.

- Б. Желязнов совместно с Юниксутовым изучают настройки сервера для возможного создания нужного для развертывания образа самостоятельно.
3. Эксплуатация паркового ПО.
- А. Желязнов по горячим следам оформляет записи в виде РИ по настройке парковой среды и согласует с Ивановым.
- Б. *Иванов присылает Перерве материалы, о которых говорили, – схему развертывания в Z и «презентацию» демонстрации.*
- В. Бойков уточняет возможный сценарий демонстрации, исходя из поправок Иванова и технологической последовательности, изложенной Ивановым.
4. Работоспособность паркового ПО.
- А. *Перерва связывается с Ивановым, пытается получить консультацию и выяснить возможные причины того, почему не работает. Причина выявлена, АРМ 3 заработал.*
- Б. Писарев завтра, 21.11, проверяет технологическую цепочку в парке.
- В. Писарев завтра, 21.11, проводит обучение сотрудников парка в объеме, достаточном для выполнения проверок их силами.

*С уважением, Перерва А. Д.*

## **Письмо 4**

### **Статус на сегодня и уточненные дальнейшие шаги**

1. *Желязнов 20.11 забирает в парке записанный образ сервера – развернуть среду не удалось.*
2. Создание образов паркового сервера.
- А. *Желязнов договаривается с администратором Z о том, чтобы работать с сервером через TeamViewer. Доступ есть.*
- Б. Желязнов совместно с Юниксутовым изучают настройки сервера для возможного создания нужного для развертывания образа самостоятельно.
- В. Перерва пытается получить образ сервера от Головатого и Comranу 1.
3. Эксплуатация паркового ПО.
- А. *Желязнов по горячим следам оформляет записи в виде РИ по настройке парковой среды и согласует с Ивановым.*
- *РИ на согласовании у Иванова, Писарева, Перервы.*
- *РИ согласована.*
- Б. *Иванов присылает Перерве и Писареву материалы, о которых говорили.*
- В. *Писарев уточняет возможный сценарий демонстрации, исходя из поправок Иванова и технологической последовательности, изложенной Ивановым.*
- *Сценарий уточнен и согласован Ивановым.*

## 4. Работоспособность паркового ПО.

А. Перерва пытается получить консультацию и выяснить возможные причины того, почему не работает. Причина выявлена, АРМ 3 заработал.

Б. Писарев проверяет технологическую цепочку в парке:

– 21.11 проверена технологическая цепочка: 1) нажатие тревожной кнопки на ТС; 2) отработка тревожного сообщения на АРМ 1 формированием заявки; 3) формирование наряда на АРМ 3 на основании заявки, печать наряда; 4) замена накопителя, закрытие наряда, печать акта о замене накопителя; 5) снятие данных на АРМ 4 на CD; 6) проверка данных на АРМ 5 в объеме 1–2–3–4. 5-й шаг не работает. Причина не определена. Сбой ПО привел к невозможности повторения шага 5;

– 26.11 Писарев проводит чистый эксперимент на выделенной среде с целью проверки работоспособности технологической цепочки в парке при соблюдении процесса. Безуспешно. Проблемы со средой;

– 27.11 Писарев и Желязнов продолжают работу по чистому эксперименту.

В. По факту успешного прохождения п. 4, б, Писарев проводит обучение сотрудников парка в объеме, достаточном для выполнения проверок их силами.

*С уважением, Перерва А. Д.*

И так далее... Я думаю, что приведенных практических писем-примеров достаточно для иллюстрации идеи.

Почему рекомендуется держать весь текст текущего «полета пули» перед глазами на одном листе? Чтобы видеть всю «траекторию движения», которую совершила наша «пуля» после выстрела до настоящего момента, видеть ее текущее движение (next steps), возможные пути отклонения и новую траекторию «пули» (если и когда это возможно).

Постепенно из картины «полета пули» появится более структурная картинка из ближайших шагов (краткосрочные планы) и рисков/возможностей (то, что не отмечено ни как выполненные работы, ни как дальнейшие шаги).

Также совместно с паттерном «Экстремальная разработка» рекомендую использовать принцип Agile-управления проектом.



Организовывать проектную команду с учетом принципов построения команды в Agile-методологии (Scrum Team). В команду входят представители отдела продаж и менеджер продукта, которые выступают как клиент для остальных участников проектной команды.

Контролировать границы проекта через бэклог итерации и получать подтверждение понимания сути работ и обязательства о выполнении этой работы от каждой задействованной проектной роли.

Рекомендуется создать инструмент/матрицу, помогающую формировать WBS проекта (перечень обязательных/рекомендуемых деятельности)

в зависимости от параметров проекта (характеристик проекта, требований/ограничений по треугольнику качества, прочих рисков проекта (реестр типовых рисков), CSF, KPI и KGI компании).

Проводить ежедневные 15–30-минутки по Scrum agenda в проектах на темы:

- ◆ «Я узнал новое о проекте/продукте»;
- ◆ «Я сегодня собираюсь сделать...»;
- ◆ «Я жду информации от...»;
- ◆ «Мне мешает выполнить работу...».

Проводят эти совещания, стоя возле флипчарта с бэклогом итерации.

Для сложных проектов организовывать итерации продолжительностью максимум 1–2 недели, для простых — 3 дня.

Конечно, в этом разделе указаны только основные моменты управления проектами по Scrum, так сказать, скелет этого подхода. Один из идеологов Agile в России дал хорошее определение: «Agile — это итеративность, инкрементальность и самоорганизующиеся команды». С моей точки зрения, дать более четкое и короткое определение невозможно. Если Agile вас интересует, я советую обратиться к паттернам «Самоорганизующиеся команды», «Итеративная разработка ПО» и «Лидерство». Паттерн «Итеративная разработка ПО» даст вам информацию о правильных принципах построения бэклога, а «Самоорганизующиеся команды» и «Лидерство» помогут в построении эффективной команды.

## 6.3.2. Паттерн «Чек-листинг»

Это самый простой для понимания паттерн. И, наверное, самый короткий по описанию в этой книге.

Суть паттерна заключается в том, что при выполнении какого бы то ни было процесса сотрудники заполняют чек-листы, или листы самопроверки, где перечислены основные выученные уроки, которые легли в основу хороших практик.



То есть паттерны «Выученные уроки» и «Хорошие практики» тесно связаны с этим паттерном.

Далее приведен пример такого чек-листа из одного из моих реальных проектов (табл. 6.4).

**Таблица 6.4.** Чек-лист по подготовке устройств CitySurf

Проверочный пункт	Отметка о прохождении
<b>1</b>	<b>2</b>
ИД устройств (-а) (допускается перечень ИД через запятую)	
Устройство прошито прошивкой	
Установленная версия ПО	

1	2
Флажок отладки (USB Debugging) снят	
Параметр VM Heap в Syslogonmod Options установлен в значение «Не ниже 48 Мбайт» (максимальное для устройств P500)	
Установлено правильное время в 24-часовом формате	
При прошивке устройств использовалась своя админка для каждого города (либо вручную удалялись теги карты предыдущего города)	
Файл Update.apk скопирован в корень карты памяти, и при тестировании его запустили, то есть установили автообновление программы	
Точка эпицентра установлена и проверена	
Для тестирования в гостиницах или еще где-нибудь выдано 2 (два) устройства	
Перед выдачей устройства проверено рабочее состояние контента, выполнена проверка на критические ошибки и «вылет» из программы	
На устройстве включен GPS-модуль	
На устройства залит максимальный текущий контент для данного города	
На устройстве нет тестовых прогулок. Исключение — конкретная просьба	
Устройство для выдачи клиенту или для демонстрации потенциальным клиентам и заинтересованным лицам брендировано	
Вместе с устройством выданы зарядное устройство и наушники (устройство может тестироваться несколько дней)	
Если устройства готовятся для гостиницы, лицензия погашена	
После прошивки устройств провели синхронизацию с сервером, чтобы лицензии стали активны	

Устройство подготовил (ФИО): \_\_\_\_\_

Как видно по тексту проверочных строк, большинство из них появлялись по мере выполнения данной процедуры — обновления ПО на мобильном устройстве.

### 6.3.3. Паттерн «Смартинг»



Тоже очень простой, но крайне эффективный паттерн. Он особенно критичен при постановке задач менеджером своей команде. Любая задача должна преследовать достижение цели.

Цели должны быть:

- ◆ конкретными (Specific). Цель должна быть четко сформулирована. Иначе может быть достигнут результат, отличающийся от запланированного;
- ◆ измеримыми (Measurable). Если у цели нет каких-либо измеримых параметров, то будет невозможно определить, достигнут ли результат;
- ◆ достижимыми (Achievable). Нужно ставить сложные цели, предполагающие приложение значительных усилий, но при этом они должны быть достижимыми. Не нужно ставить такие цели, которые приводили бы к увеличению количества стрессов в вашей жизни или в жизни ваших подчиненных.

В идеале SMART-цели используются в качестве стимула для успешного развития в целом, через цепочку успехов в достижении конкретных целей;

- ◆ ориентированными на результат (Result oriented). Цели должны характеризоваться на основе результата, а не проделываемой работы. Таким образом достигается эффективность. Можно поставить себе цель работать на два часа больше, но если не определен конкретный результат такой работы, то эффективность данной цели вызывает большие сомнения;
- ◆ имеющими конкретный срок достижения (Timed). Любая цель должна быть достижима в определенное время.

Крайне важно, чтобы ваши отношения как «вниз» — с командой, так и «вверх» — с топ-менеджерами строились на основе SMART-целей.

У находящегося на позиции менеджера неизбежны производственные конфликты. Порекомендую вам одно из высказываний К. Фиорины [11]: «Не стоит переживать из-за чьей-то узколюбости или заблуждений. Этот мир несправедлив. Эту реальность следует принять и не позволять себе переживать из-за нее. Ваши сердце и нервы принадлежат только вам и никому больше. Берегите их».

Кстати, К. Фиорина — потрясающая личность и один из лучших менеджеров в мире. Она возглавляла HP. Я рекомендую вам найти в Интернете ее книги и почитать их.

### 6.3.4. Паттерн «Расширенный устав проекта»

Если проект разрабатывается в жестких условиях ограничения по срокам и стоимости управленческих активностей, то предыдущий паттерн вам, скорее всего, не подойдет. Для проектов, управляемых которых должна быть высокой, а затраты на создание управленческих артефактов — минимальными, я рекомендую пользоваться этим паттерном.



Паттерн основывается на создании одного документа, в котором объединены несколько управленческих артефактов. При таком подходе вы не перегружаете проект процедурами управления, но не теряете управляемость проекта. К сожалению, такой подход все чаще применяется даже там, где надо бы более тщательно выстраивать процессы управления. Таковы реалии нашего времени — мало кто готов платить за качество, тем более если это часто непонятное качество управления проектом, и мало кто способен объяснить внятно, за что же, собственно, платит, а те, кто способны, настолько часто делали это в своей практике, что иногда им проще и эффективнее выбрать подход, похожий на этот паттерн.



На сайте книги, доступном по ссылке [http://bit.ly/pmway\\_projectcharterex](http://bit.ly/pmway_projectcharterex), вы найдете реальный пример документа «Устав проекта», который разработан в соответствии с этим паттерном. Документ был создан мной в программе проектов по созданию продукта «Аудиогид», в которой было несколько про-

ектов. Центральный проект, собственно «Аудиогид», выполнялся ближе к «СУП на РМВОК» по методологии SADD (смотрите сайт книги <http://br4you.ru> и первую книгу). Второй проект — по разработке части решения, которая называлась «Редактор контента», — выполнялся с применением данного паттерна в управлении по методологии Iconix с уклоном в Agile. Третий и четвертый проекты — по созданию АРМ управления — выполнялись с использованием паттерна «Управление через реестры» по методологии Agile.

---

Здесь я хочу привести только оглавление такого расширенного устава.

Общие сведения о проекте

Наименование проекта

Бизнес-цель инициации проекта

Цель проекта

Заказчик проекта

Исполнитель проекта

Основание для выполнения проекта

Портрет проекта

    Основные заинтересованные лица проекта

    Бизнес-ожидания от проекта

    Контрольные точки и вехи проекта

Управление проектом

    План управления проектом

        Методики, применяемые на проекте

        Этапы проекта

        План-расписание проекта

        Периодическая отчетность по проекту

    План управления границами и рисками проекта

        Управление границами проекта

            Реестр «Границы релизов»

            Реестр «Архитектурные риски»

            Реестр «Проблемы и открытые вопросы»

        Управление изменениями проекта

    План управления качеством

        Критерии оценки проекта

            Критерии оценки результатов проекта

            Критерии оценки работ исполнителя

- Требования к документированию по фазам проекта
- Релизное тестирование
- Приемочное тестирование
- План управления коммуникациями
  - Персональный состав команды проекта
  - Функции членов команды проекта
  - Взаимодействие сторон
  - Виды взаимодействия в проекте
  - Уточнение технического задания
- План управления документами
  - Стандарт хранения документов
  - Структура корневого каталога
    - Справочные папки
    - Доступ к проектной библиотеке
  - Стандарт идентификации документов
    - Имя файла документа в электронной библиотеке
    - Стандарт подготовки документов
- Процедура управления качеством проектной документации
  - Назначение
    - Роли процедуры утверждения
    - Рецензирование документа (только для технической документации)
    - Результаты процедуры
    - Утверждение итогового документа
    - Результаты процедуры

Как видите, это небольшой, но информативный документ, который выполнил, по сути, роль устава проекта, ПУП и процессного руководства.

### 6.3.5. Паттерн «СУП на РМВОКе»

Название этого паттерна получилось забавным, но уверен, оно запомнится быстро и просто. Суть паттерна – в формировании архитектуры СУП (системы управления проектами) на базе методологии РМВОК. На сайте книги, доступном по ссылке [http://bit.ly/pmway\\_rmsrmbok](http://bit.ly/pmway_rmsrmbok), вы найдете презентацию, которая в визуальной форме показывает эти же процедуры.

В данном разделе представлены результаты проекта по построению СУП в компании при следующих вводных.

- ◆ Разработать концепцию системы управления проектами для компании — производителя программного обеспечения.
- ◆ Основное направление деятельности компании — разработка коробочных продуктов и адаптируемых решений клиентов из коробочных продуктов (90 % операционной деятельности компании). Компания также разрабатывает проекты под ключ (10 % операционной деятельности).
- ◆ В концепции системы управления проектами (СУП) должны быть показаны место СУП в системе управления производством (СУПр) и зависимость формирования портфеля проектов СУП от стратегии развития продуктов, маркетинговой стратегии компании.
- ◆ Концепция СУП должна быть структурирована по типам проектов в компании. Каждый раздел, описывающий тип проекта, должен содержать краткое описание ЖЦ проекта.
- ◆ В концепции СУП должны быть отражены универсальные принципы управления проектами, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев в большинстве компаний.
- ◆ Должна быть разработана регламентная база СУП — перечень процедур, РИ и методологий по управлению проектом и разработке ПО. При разработке регламентов требуется учитывать рекомендации РМВОК и реальную ситуацию в компании.

Под архитектурой СУП понимается перечень процедур и спецификаций СУП. Архитектура СУП основывается на стандарте РМВОК.

## Процедуры СУП

В табл. 6.5 представлены процедуры СУП с описанием основных видов деятельности, выполняемых в рамках процедуры, и перечнем наиболее важных артефактов, изменяемых или создаваемых в ходе выполнения процедуры.



---

Комментарий «драфт» рядом с артефактом означает, что выполнение процедуры не финиширует создание этого артефакта, а значит, не требуется проводить процедуру формального согласования и утверждения артефактов с таким комментарием.

Комментарий «релиз» говорит о том, что создается новый релиз артефакта, следовательно, должна быть выполнена процедура «Управление конфигурацией».

---

Как можно видеть, два артефакта проектного управления — ПУП (план управления проектом) и ПУД (план управления документами) — наиболее важные артефакты, являющиеся стержнем всего проекта.



ПУП определяет, как должен выполняться, контролироваться и закрываться проект, регламентирует использование инструментов и методов, используемых в процессах управления проектом.

ПУД содержит описание плана управления документами в проекте, в частности то, какие артефакты, спецификации и документы к концу какого этапа проекта (фазы) в каком состоянии должны быть, кто является согласующим и утверждающим для каждого артефакта.

**Таблица 6.5.** Процедуры СУП

Процедура	Основная деятельность, выполняемая в рамках процедуры	Наиболее важные результирующие артефакты
1	2	3
Создание инфраструктуры проекта	Выбрать менеджера проекта. Определить политики компании и существующие системы. Собрать процессы, процедуры и историческую информацию. Идентифицировать заинтересованных лиц. Создать структуру репозитория проекта	Репозиторий проекта. ПУП (драфт)
Разработка устава проекта	Документировать бизнес-потребности. Определить цели проекта. Определить допущения и ограничения. Разработать устав проекта	Устав проекта. Предварительное описание содержания (драфт)
Первичное планирование проекта	Разбить проект на фазы. Разработать предварительное содержание проекта	Предварительное описание содержания (драфт)
Разработка WBS проекта	Создать описание содержания проекта. Определить команду. Создать WBS и словарь WBS	WBS проекта. Описание содержания проекта. ПУП (драфт). ПУД (драфт)
Планирование выполнения проекта	Создать список активностей. Создать сетевую диаграмму. Оценить требуемые ресурсы (процедура оценки трудоемкости). Оценить время и стоимость. Определить критический путь. Создать план-расписание	ПУП (драфт). ПУД (драфт). Бюджет проекта (оценка трудоемкости). План-расписание проекта (драфт)
Определение рисков проекта	Идентифицировать риски. Провести качественный и количественный анализ рисков. Планировать реагирование и предупреждение рисков	RBS проекта (реестр рисков)
Планирование управления проектом	Определить, как будет производиться планирование. Определить роли и ответственности. Разработать бюджет.	ПУП (релиз). ПУД (релиз)

1	2	3
	<p>Определить стандарты, процедуры и метрики качества.</p> <p>Определить, какие компоненты системы покупать (отдавать на разработку на аутсорсинг).</p> <p>Создать конечную версию плана управления проектом и baseline(s) для измерения производительности.</p> <p>Закончить аспект «как выполнять и контролировать» в плане управления проектом.</p> <p>Создать план улучшения процессов.</p> <p>Подготовить документы по выбору контракторов</p>	
Старт проекта	<p>Получить формальное согласие участников команды.</p> <p>Получить ресурсы в конечную команду проекта.</p> <p>Собрать стартовый митинг</p>	План-расписание проекта (релиз)
Обеспечить возможность выполнения проекта	<p>Убедиться в управлении всеми ожиданиями заинтересованных лиц.</p> <p>Убедиться в одинаковом понимании работы.</p> <p>Установить каналы связи и управлять ими</p>	ПУП (релиз)
Выполнение проекта	<p>Выполнять проект в соответствии с планом управления проектом.</p> <p>Реализовывать утвержденные действия по исправлению ошибок, запросы на изменения, корректирующие действия, превентивные действия.</p> <p>Использовать систему авторизации работ.</p> <p>Следовать определенным процессам.</p> <p>Создавать отчеты о прогрессе проекта.</p> <p>Завершать выполнение работ по пакетам работ WBS.</p> <p>Распространять информацию в соответствии с планом управления коммуникациями</p>	<p>Карточка проекта.</p> <p>План-расписание проекта.</p> <p>WAS (реестр работ и артефактов).</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр выученных уроков.</p> <p>Отчет о ходе выполнения проекта</p>
Управление аутсорсерами	<p>Запрашивать ответы от поставщиков (аутсорсеров).</p> <p>Выбирать поставщиков (аутсорсеров).</p> <p>Администрировать контракты</p>	<p>Документы по выбору контракторов (критерии оценки).</p> <p>Коммерческие предложения (от поставщиков).</p> <p>Контракт.</p> <p>SOW (артефакт СУТ).</p> <p>Устав проекта (от поставщика).</p> <p>Описание содержания проекта (от поставщика).</p> <p>План-расписание проекта (от поставщика).</p> <p>План управления коммуникациями.</p> <p>План управления рисками</p>

Таблица 6.5 (продолжение)

1	2	3
Управление запросами на изменения	<p>Управлять конфигурацией (процедура).</p> <p>Рекомендовать запросы на изменения и корректирующие действия.</p> <p>Управлять Deliverables и сроками поставки.</p> <p>Измерять параметры проекта по отношению к бейзлайну производительности в соответствии с планом измерений.</p> <p>Определять отклонения. При значительных отклонениях инициировать запросы на изменения или коррективы.</p> <p>Запрашивать изменения, рекомендованные корректирующие, превентивные действия и действия по исправлению ошибок.</p> <p>Анализировать информацию о выполнении работ.</p> <p>Рекомендовать действия по исправлению ошибок, запросы на изменения, корректирующие действия, превентивные действия.</p> <p>Утверждать изменения на ССВ проекта и продукта.</p> <p>Утверждать действия по исправлению ошибок, запросы на изменения, корректирующие действия, превентивные действия.</p> <p>Убеждаться, что только утвержденные изменения выполняются</p>	<p>Реестр запросов на изменение.</p> <p>ПУД (релиз).</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр рисков</p>
Обеспечение качества выполнения проекта	<p>Выполнять действия по контролю качества с точки зрения следования определенным процессам.</p> <p>Исполнять все запланированные в системе качества систематические действия для обеспечения уверенности в том, что проект будет соответствовать необходимым стандартам качества.</p> <p>Проводить аудит проекта</p>	<p>Руководство по качеству компании.</p> <p>Чек-лист аудита проекта.</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр рисков</p>
Построение команды	<p>Заниматься формированием команды.</p> <p>Оценивать производительность и вклад сотрудников в успех проекта.</p> <p>Давать награды и поощрения</p>	<p>Реестр выученных уроков.</p> <p>Реестр проблем и решений</p>
Прохождение вехи проекта	<p>Контролировать завершение работ по пакетам работ WBS.</p> <p>Завершать все работы в границах проекта</p>	<p>Чек-лист прохождения вехи проекта.</p> <p>Протокол совещания о прохождении вехи проекта.</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр рисков</p>
Управление рисками и проблемами	<p>Контролировать расписание, стоимость и качество по отношению к их бейзлайнам, контролировать импакт на них.</p> <p>Управлять временем и резервами.</p>	<p>Реестр рисков.</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр выученных уроков</p>

1	2	3
	<p>Получать дополнительное финансирование, когда требуется.</p> <p>Использовать журнал проблем и открытых вопросов.</p> <p>Идентифицировать корневую причину проблемы.</p> <p>Разрешать конфликты.</p> <p>Проводить аудит рисков.</p> <p>Создавать прогнозы</p>	
Отчетность по проекту	<p>Отчитываться о производительности.</p> <p>Измерять производительность членов команды</p>	<p>Отчет о ходе выполнения проекта.</p> <p>План-расписание проекта.</p> <p>ПУД (релиз)</p>
Подготовка к завершению проекта	<p>Разработать процедуру закрытия проекта.</p> <p>Подтвердить, что все требования проекта были выполнены.</p> <p>Произвести конечные платежи.</p> <p>Завершить закрытие контракта.</p> <p>Определить и задокументировать, что проект достиг критериев завершения или закрытия</p>	<p>Чек-лист прохождения вехи проекта.</p> <p>План приемочных испытаний (артефакт СККП)</p>
Сдача проекта	<p>Получить документ о формальной приемке продукта.</p> <p>Создать конечный отчет о производительности.</p> <p>Измерить степень удовлетворенности заказчика</p>	<p>Чек-лист прохождения вехи проекта.</p> <p>Акт сдачи-приемки продукта</p>
Завершение проекта	<p>Собрать выученные уроки и обновить базу знаний.</p> <p>Провести выученные уроки (процедура).</p> <p>Обновить записи проектов.</p> <p>Сдать законченный проект.</p> <p>Освободить ресурсы.</p> <p>Отпраздновать</p>	<p>Реестр рисков (релиз).</p> <p>Реестр проблем и решений.</p> <p>Реестр открытых вопросов.</p> <p>Реестр выученных уроков</p>
Обучение сотрудников в проекте	<p>Определить потребность в обучении.</p> <p>Создать план управления обучением.</p> <p>Создать план-расписание обучения</p>	<p>ПУП (релиз).</p> <p>План-расписание обучения</p>
Оценка трудоемкости	<p>Оценить влияние на утвержденные требования/архитектуру/артефакты.</p> <p>Оценить требуемые ресурсы</p>	<p>Карточка проекта.</p> <p>Бюджет проекта (оценка трудоемкости)</p>
Формальное согласование и утверждение артефакта	<p>Определение аудитории согласования.</p> <p>Инициация согласования.</p> <p>Устранение замечаний</p>	<p>ПУД (релиз)</p>
Проведение выученных уроков проекта	<p>Обсудить реестр выученных уроков в проектной команде.</p> <p>Доложить о хороших практиках и рекомендациях на РМО.</p> <p>Инициировать изменения процессов через запрос на изменение процессов в SEPC</p>	<p>Реестр выученных уроков.</p> <p>Запрос на изменение процессов.</p> <p>Чек-лист аудита процессов проекта</p>

Продолжение ⇨

Таблица 6.5 (продолжение)

1	2	3
Управление конфигурацией	Создание бейзлайна требований проекта. Создание бейзлайна тестовых спецификаций проекта. Создание бейзлайна архитектурных спецификаций проекта. Создания бейзлайна требований продукта. Создание бейзлайна тестовых спецификаций продукта. Создание бейзлайна архитектурных спецификаций продукта	ПУП. Описание объектов конфигурации (через WAS)

## Артефакты СУП



В табл. 6.6 представлены артефакты проектного управления, список проверяющих, согласующих и утверждающих лиц.

Артефакты других СУ, такие, например, как стратегия тестирования (СККП), концепция системы (СУАР) или спецификация варианта использования (СУТ), в таблице не представлены и выходят за рамки концепции СУП.

Мы с вами рассмотрели паттерн «СУП на РМВОКе», который вы можете брать за основу при проектировании архитектуры СУП в вашей компании и принятия решения о том, какие практики и артефакты вам нужны, а какие — нет.

### 6.3.6. Паттерн «СУП на воде, или Критические принципы проектного управления»



Далее представлены универсальные принципы управления проектами и проектными коммуникациями, подходящие без серьезной адаптации для любой компании и любых проектов и рекомендуемые к применению в большинстве случаев.

В этом разделе приведена только часть принципов — наиболее критические для успеха управления проектами и те, которые должны быть внедрены в первую очередь.

- ◆ **Определенность проекта.** В проекте всегда должен создаваться устав как минимум со следующими разделами: бизнес-потребность в выполнении проекта (идеи, BVISION и TVISION), ограничения и допущения по проекту, проектная команда, фазы проекта, критические вехи проекта, дедлайны по срокам выполнения проекта (если есть).

Таблица 6.6. Артефакты СУП

Тип артефакта/ документа	Описание артефакта/ документа	Ответственный	Список проверяющих (reviewers)	Список согласователей	Список утверждающих лиц	
1	Project Management Condict (Кодекс менеджера проекта)	2	3	4	5	6
	Данный артефакт описывает кодекс поведения, этику и профессиональные нормы, которыми обязаны руководствоваться менеджеры проекта компании в профессиональной деятельности	Директор РМО	РМО	Software Engineering Process Group (SEPG)	Исполнительный комитет	
	Заявка на создание продукта/ услуги. Документ является основным источником информации для принятия решения по инициации проекта	Сотрудник компании	Председатель продуктового комитета	Бизнес-аналитик, основные заинтересованные лица проекта	Продуктовый комитет	
	Устав проекта. Данный документ является документом высокого уровня, описывающим потребность в создании продукта/ услуги, выявленные требования к продукту/ услуге, и дает полномочия назначенному менеджеру проекта привлекать ресурсы компании для выполнения проектных работ	Менеджер проекта	Бизнес-аналитик, заинтересованные лица проекта	Основные заинтересованные лица проекта	Заказчик проекта, спонсор проекта, менеджер проекта	
Work Breakdown Structure (иерархическая структура работ)	Документ, описывающий согласованную с результатами поставки иерархическую декомпозицию работ, которую команда проекта должна выполнить для достижения целей проекта и создания оговоренных результатов поставки. С его помощью структурируется и определяется все содержание проекта. WBS может быть представлено в форме плана-расписания без указания трудоемкости	Менеджер проекта	Бизнес-аналитик, системный архитектор, тест-менеджер	Бизнес-аналитик, системный архитектор, тест-менеджер	ССВ проекта	

Продолжение ⇨

Таблица 6.6 (продолжение)

1	2	3	4	5	6
Preliminary Project Score Statement (предварительное описание содержания проекта)	Предварительное описание проекта дает общее представление о его содержании всем участникам проекта и заказчику, содержит основные цели проекта. Служит для окончательного уточнения содержания проекта	Менеджер проекта	Бизнес-аналитик, заинтересованные лица проекта	Менеджер проекта, основные заинтересованные лица проекта	Заказчик проекта, спонсор проекта
Project Scope Statement (описание содержания проекта)	Описание проекта дает общее представление о его содержании всем участникам проекта и содержит основные цели проекта. Документ подробно описывает поставку проекта и работы, необходимые для создания этих поставок	Менеджер проекта	Бизнес-аналитик, заинтересованные лица проекта	Менеджер проекта, основные заинтересованные лица проекта	Заказчик проекта, спонсор проекта
Project Schedule Baseline (базовый план расписания проекта)	Принимается и утверждается командой управления проектом в качестве базового плана-расписания. В нем указаны базовый старт и базовый финиш	Менеджер проекта	Бизнес-аналитик, системный архитектор, тест-менеджер	Бизнес-аналитик, системный архитектор, тест-менеджер	Команда проекта
Project Management Plan (ПМП)	Документ определяет, как должен выполняться проект, регламентирует инструменты и методы, используемые в процессах управления проектом. Включает в себя: <ul style="list-style-type: none"> <li>• Project Scope Management Plan;</li> <li>• Staffing Management Plan;</li> <li>• Communication Management Plan;</li> <li>• правила проектной переписки;</li> <li>• правила формирования адресатов переписки;</li> <li>• каналы коммуникации;</li> </ul>	Менеджер проекта	Бизнес-аналитик, заинтересованные лица проекта	Менеджер проекта, основные заинтересованные лица проекта	Заказчик проекта, спонсор проекта, директор ГМО, директор по качеству

1	2	3	4	5	6
	<ul style="list-style-type: none"> <li>• правила использования каналов коммуникации;</li> <li>• правила разрешения конфликтов и эскалации проблем;</li> <li>• Quality Management Plan;</li> <li>• Risk Management Plan;</li> <li>• Schedule Management Plan;</li> <li>• Requirements Management Plan;</li> <li>• Configuration Management Plan;</li> <li>• Procurement Management Plan;</li> <li>• Contract Management Plan</li> </ul>				
Document Management Plan (ПУД)	Документ содержит описание плана управления документами в проекте, в частности, какие артефакты, спецификации и документы к концу какого этапа проекта в каком должны быть состоянии	Менеджер проекта	Бизнес-аналитик, системный архитектор, тест-менеджер	Бизнес-аналитик, системный архитектор, тест-менеджер	ССВ проекта
Реестр вопросов и ответов	Реестр вопросов и ответов, возникающих при выполнении работ в проекте	Менеджер проекта	Проектная команда	Не требуется согласование	Не требуется утверждение
Реестр рисков	Реестр для управления рисками проекта	Менеджер проекта	Менеджер проекта	Проектная команда	ССВ проекта
Реестр проблем и решений	Реестр для управления проблемами и фиксированием решений в проекте	Менеджер проекта	Менеджер проекта	Проектная команда	Не требуется утверждение
Реестр выученных уроков	Реестр для фиксирования выученных уроков проекта	Проектная команда	Менеджер проекта	Не требуется согласование	Не требуется утверждение
Оценка трудоемкости выполнения запроса на изменения	Служит для оценки трудоемкости	Менеджер проекта	Проектная команда	Не требуется согласование	Менеджер по продукту
Отчет о ходе выполнения проекта	Служит для информирования заинтересованных лиц проекта и заказчика проекта о ходе его выполнения	Менеджер проекта	Проектная команда	Не требуется согласование	Не требуется утверждение

Продолжение ⇨

Таблица 6.6 (продолжение)

1	2	3	4	5	6
Чек-лист прохождения ревью вехи проекта	Служит для контроля качества, является основой для формального подтверждения прохождения вехи проекта проектной командой и заказчиком	Менеджер проекта	Не требуется проверка	ССВ проекта	Заказчик, ССВ проекта
Запись о прохождении ревью процессов	Служит для контроля качества, выполняется в форме аудита качества процессов	Аудитор проекта	Менеджер проекта	Менеджер проекта	SEPG
Матрица допустимых совмещений ролей в проекте	Матрица допустимых совмещений ролей в проекте в зависимости от каждого выявленного типа проекта	Директор РМО	РМО	SEPG	Исполнительный комитет
Шаблон протоколов проектных совещаний	Шаблон протоколов проектных совещаний	Проектная команда	Менеджер проекта	Участники совещания	Не требуется утверждение
Чек-лист аудита проекта	Служит для контроля качества, является основой для формального подтверждения прохождения вехи проекта проектной командой и заказчиком	Аудитор проекта	Менеджер проекта	Не требуется согласование	SEPG

- ◆ **Управляемые проблемные совещания.** Все проблемные совещания должны иметь повестку и назначаться заранее. К ним следует готовить материалы для ознакомления, которые должны рассылаться всем участникам встречи вместе с приглашением на встречу. На совещании нужно выбирать секретаря и модератора. Последними вопросами совещания должны быть обсуждение дальнейших шагов и решение, кто должен получить протокол этого совещания. Секретарь совещания по окончании совещания в течение 3–5 часов создает протокол и выкладывает его в проектное хранилище, уведомляя участников совещания и определенных на встрече заинтересованных лиц.
- ◆ **Прозрачность проектных коммуникаций.** Если в проекте не создаются планы управления коммуникациями, требованиями, рисками, качеством и границами проекта, требуется провести отдельные проблемные совещания со всей проектной командой и ССВ проекта по обсуждению этих вопросов. Протоколы совещаний заменят собой упомянутые планы управления проектом.
- ◆ **Статус-митинги как средство повышения предсказуемости и прозрачности в проекте.** Согласно этому принципу все участники проектной команды обязаны своевременно и по существу информировать менеджера проектов и проектную команду о статусах выполнения задач и артефактов, за которые они отвечают, через систему авторизации работ (WAS). Статус-митинги служат для обсуждения реестра рисков, реестра проблем и решений, реестра вопросов и ответов и реестра выученных уроков. На статус-митингах производится командное обсуждение и выявление новых рисков, проблем, открытых вопросов и выученных уроков, анализируются уже существующие риски и другие реестры, актуальность информации в них. Статус-митинги не предназначены для выработки каких бы то ни было решений. Решения и детальные обсуждения предпринимаются на проблемных совещаниях в соответствии с принципом «управляемые проблемные совещания по существу».
- ◆ **Прозрачное делегирование конфликтов в проектах.** Реестр проблем и открытых вопросов проекта ведется всей проектной командой с момента старта проекта. Любой сотрудник имеет право внести в него запись. Реестр проблем и открытых вопросов обсуждается на статус-митингах. Согласно этому принципу любой сотрудник компании, участвующий в проекте, может делегировать конфликт директору РМО, если считает, что конфликт существует и не разрешен менеджером проекта. При подобном делегировании конфликта обязательно создается письмо с описанием конфликта с точки зрения инициатора, копия отсылается менеджеру проекта. Менеджер проекта может дополнить это письмо своими соображениями, ответив только директору РМО. Если директор РМО выступает в роли менеджера проектов, такое письмо адресуется непосредственно директору центра разработки ПО.
- ◆ **Возможность «выбросить» show stopper.** Show stopper — это «флаг», «выбрасыванием» которого сотрудник проектной команды хочет привлечь внимание РМО к вопиющим проблемам в процессах, методике разработки ПО и/или методике управления проектами, так как считает сложившийся ход

проекта и распоряжения менеджера проекта неправильными и/или вредными для проекта. До разрешения конфликта сотрудник имеет право не выполнять распоряжения менеджера проекта. Разрешается проблема группой в составе директора РМО, сотрудника, инициировавшего show stopper, и SEPC. Любой show stopper после разрешения должен быть рассмотрен на РМО.

- ◆ **Прозрачные выученные уроки.** Реестр выученных уроков ведется всей проектной командой с момента старта проекта. Любой сотрудник имеет право внести свои выученные уроки и плохие практики в реестр выученных уроков. Менеджер проекта не имеет права править видение выученных уроков, представленное членом проектной команды. Если после обсуждения с менеджером проекта инициатор не признает выученный урок разрешенным, то обсуждение таких уроков происходит в рамках всей проектной команды. Утвержденные командой хорошие практики, плохие практики и рекомендации по устранению плохих практик менеджер проекта доносит до РМО.

- ◆ **Управление рисками**, в том числе технологическими, с помощью выявления степени влияния, вероятности наступления риска и буферизации времени.

Требуется разработать реестр типовых рисков в соответствии с ЖЦ, указать, кто за риск отвечает, и в качестве плана предупреждения рисков указать, через какие артефакты и деятельности какие роли снимают эти риски. Например, «риск того, что искажены ожидания клиента», отвечает РМ, план предупреждения — утверждение «описания содержания проекта» архитектором проекта. Или «риск нереализуемости концепции системы в оцененные сроки», отвечает архитектор проекта, план предупреждения — согласование концепции с разработчиками и системным аналитиком.

Если какой-то риск наступил по вине ответственного, это должно учитываться в КРІ (эфффективное управление рисками) и влиять на его зарплату.

Рекомендуется применять прототипирование критических моментов, связанных с производительностью системы, особенно в ее взаимодействии с окружением (другими системами) и при значительных нагрузках системы.

Рекомендуется разрабатывать функциональные работающие прототипы разрабатываемой системы с ограниченной функциональностью, чтобы снять риски неправильно разработанного решения как можно раньше.

Управлять риском превышения времени обработки комментариев клиентом через добавление времени к оценке трудоемкости. Добавочное время рассчитывается на основе предыдущего опыта как среднее время, затрачиваемое на согласование комментариев всем клиентам.

- ◆ **Управление расписанием проекта.** WBS проекта должна адаптироваться из типовой WBS компании и учитывать требования к качеству продукта/характеристики конкретного проекта. Если какая-то работа из WBS исключается, следует обязательно проверить RBS проекта на наличие соответствующего риска.

План-расписание проекта создавать на уровне задач, необходимых для производства объекта поставки или значимого проектного артефакта, все более низкоуровневые и узкоспециализированные задачи учитывать через систему авторизации работ (WAS). Планирование должно содержать майлстоуны (вехи проекта).

Следует заранее определять крупные вехи проекта, по наступлению которых РМ отчитывается перед заказчиком о ходе выполнения проекта. Все крупные вехи должны иметь чек-листы для формального контроля.

Недостаточно называть заказчику срок проекта, рассчитанный только по трудоемкости проекта. Требуется проверять возможность его обеспечения ресурсами при всех уже запущенных проектах и их рисках, а также возможность переключения сотрудников с задач, которыми он уже занят, на другие задачи или на возможность параллельного их выполнения. Для этого лучше строить сетевые диаграммы проекта (зависимости проектов и задач).

Нужно создать таблицу, которая показывает чистую трудоемкость (если заниматься только этой задачей) стандартных задач (инструментов рекламной кампании), автоматически отслеживать загрузку ресурсных мощностей компании и отражать в ней рассчитанный срок выполнения задачи в зависимости от реальной загрузки мощностей компании уже начатыми задачами.

Оценивать трудоемкость должны непосредственные исполнители работы. Если дается оценка по аналогии, то будущие исполнители должны провести как минимум ревю этой оценки и добавить риски/увеличить оценки или согласиться с ней.

При изменениях в ограничениях или требованиях проекта должна выполняться переоценка трудоемкости исполнителями. Следует также пересчитывать рентабельность проекта на основании вновь вычисленной себестоимости проекта.

Если вы не знаете, как начать управлять своим проектом, попробуйте применять этот паттерн. Со временем все войдет в норму.

### **6.3.7. Паттерн «Зажарка в СУП, или Универсальные принципы управления проектами»**

#### **Принципы организации и проведения проектных совещаний**

Совещания рекомендуется проводить в отдельной комнате, оборудованной флипчартом, цветными фломастерами, телефоном для телеконференций и ПК со всем возможным ПО для рассмотрения проектных материалов, подключенным к локальной сети.

Совещания должны планироваться, иметь четко сформулированную повестку, назначенных секретаря, модератора, докладчиков, обязательных и необязательных участников. Приглашение на совещание инициатор должен рассылать за 1–2 дня до его проведения. Материалы для совещания нужно прикреплять к приглашению и рассылать как можно раньше — желательно за один день до совещания.

Секретарь совещания должен вести протокол совещания. Протокол является формальным артефактом проекта, он должен храниться вместе с остальными проектными артефактами, его следует рассылать всем участникам совещания не позднее чем через три часа после его проведения.

Мозговые штурмы должны проводиться по одной схеме: нет — критике, нет — оценке, нет — обсуждениям идей! Мозговой штурм — это идеегенерирующее событие. Все идеи, несмотря на их кажущуюся абсурдность, секретарь должен фиксировать. Обсуждать идеи нужно на обычных совещаниях по текущим проблемам.

В проектах, где не используются практики Scrum Team, стоит проводить еженедельные статус-митинги. Основные цели статус-митингов — обзор и переоценка существующих и выявление новых рисков.

## Принципы закрытия проекта

По окончании проекта рекомендуется всей командой подводить итоги в виде выученных уроков, определять, что получилось хорошо, что плохо и как можно избежать повторения плохих моментов и добиться повторяемости хороших.

PM должен составить отчет о проведенном проекте. В отчет обязательно включать следующие разделы: анализ рисков проекта и планов по их предупреждению и реагированию, анализ WBS и истории ее изменения, хорошие практики проекта, плохие практики проекта и рекомендации, выявленные сложности в отношениях с поставщиками и подрядчиками, дополнения в базу знаний, сделанные в ходе проекта.

## Принципы управления артефактами

В каждый момент времени должен назначаться единственный ответственный за любые артефакт или задачу с четким сроком плановой даты окончания работы и согласующим/принимающим артефакт/задачу. Следует обеспечить наличие и контроль статусов артефактов, а также планируемой даты готовности артефакта и определить, какая роль работает над ним в конкретный момент.

В проекте нужно как можно раньше создавать план управления документами и получать коммитмент о том, что назначение артефакта и суть работы с артефактом понятны, от каждой проектной роли, которая над ним трудится.

Должна быть введена практика внутреннего контроля качества разработанных артефактов со стороны PM с помощью заранее разработанных формальных чек-листов по мере окончания работы над ними. Следует практиковать многоуровневое согласование артефактов. Например, концепция системы:

- ◆ сначала согласуется системным аналитиком и менеджером проекта;
- ◆ затем согласуется архитектором проекта и системным аналитиком;
- ◆ затем проверяется на реализуемость архитектором проекта и разработчиками проекта;
- ◆ и в итоге проверяется на тестируемость менеджером по тестированию проекта.

Нужно ввести практику ревью исполнителем входных артефактов перед началом его работы и после формальной проверки, выполненной РМ, и срочного устранения недоработок. Обязателен формальный контроль качества утверждаемых клиентом артефактов не только РМ, но и всеми ролями, чьи обязательства этот артефакт закрепляет. При изменении утвержденного артефакта сотрудники, работающие с ним, должны быть уведомлены о произведенных изменениях.

Рекомендуется использовать в проектах систему хранилища версий. Главные плюсы использования этой системы:

- ◆ простота поддержки проектов — всегда ясно, где что расположено, где последняя версия артефактов/исходников, кто его выложил и почему в него вносились исправления;
- ◆ возможность снимать метрики в автоматическом режиме (сколько раз менялся/утверждался артефакт, каков его объем, каков измененный объем и т. д.).

Рекомендуется ввести практику внесения изменений в артефакты в режиме Change Tracking Mode. Информация о каждом типе артефакта должна иметь утвержденный в компании формат (оформляться по шаблону).

Вместо участия экспертов по продуктам/предметной области в согласованиях будет лучше, если эксперты станут особое внимание уделять ограничениям на решения и утверждать их, а РМ и системный аналитик проекта — строго контролировать их соблюдение в проекте. Ограничения в ЖЦ проекта должны появляться как можно раньше.

## Принципы управления задачами

Должна быть разработана система авторизации задач, которая позволяет управлять задачами и работами, знать, каковы фактические трудозатраты по задачам (для этого сотрудники указывают в системе свое рабочее время). При постановке задачи исполнителю тот, кто будет пользоваться результатом его работы, должен иметь возможность уточнить постановку задачи/участвовать в постановке.

Любой участник команды может дать поручение/задачу по любому артефакту любому другому члену проектной команды. Сотрудник не приступает к выполнению проектной задачи, пока она не будет авторизована РМ.

В случае если проект очень короткий и нет времени на создание полноценных ТЗ, рекомендуется управлять требованиями через Story Board — инструмент для описания поведения системы. Часто он выполняется в виде раскадровки графического интерфейса.

## Принципы управления ошибками

Для продукта следует разработать единые bug reports, с которыми должны работать все проектные роли.

Для заказчика целесообразно ведение собственного журнала ошибок, в котором фиксируются все выявленные ошибки. В дальнейшем список ошибок используется на встречах для обсуждения ошибок, а также отслеживания их статуса.

## Принципы влияния на систему менеджмента качества и корпоративную культуру

Рекомендуется проводить аудит проектов на предмет правильного выполнения процессов. Он должен заканчиваться уроками и, возможно, инициировать изменения в процессах, которые нужно рассматривать центру компетенций по процессам компании. Рекомендуется создать концепцию коммуникаций компании, описывающую основные общие правила коммуникации, правила проведения встреч/совещаний, делегирования и разрешения конфликтов.

Следует создать описание процессов компании, в котором будут использованы различные представления: по ролям, артефактам, общему ЖЦ производства, дисциплинам, отделам, KPI, KGI. Создать портфель нового сотрудника компании.

Нужно описать процедуру работы с хранилищем проектов — кто, как, куда и когда должен выкладывать свою информацию и кого об этом информировать. Рекомендуется ввести стандарт именования проектной документации, а также как можно раньше начать создание словаря терминов и понятий.

Рекомендуется создать Job Description для каждой проектной роли, где указать деятельность, которую выполняет роль, артефакты роли, планирование и отчетность роли, этические нормы роли. Прописать e-mail групп рассылок, адреса сотрудников и контакты.

PM, или менеджер по продукту, или менеджер по развитию клиента должны создавать Contact Reports после каждой встречи с клиентом и информировать (при необходимости обсуждать) участников проектной команды о Contact Reports.

При длительном отсутствии сотрудника на месте следует назначить его заместителя и письменно сообщить об этом всем сотрудникам. Отсутствующий сотрудник должен включить функцию «отсутствие в офисе» и указать, с какого по какое число его не будет и кто его замещает.

Рекомендуется ввести практику выпуска периодической корпоративной газеты в виде информационных бюллетеней, в которых рассказывать о жизни компании, ее миссии, стратегических, тактических и операционных целях, проектах, успехах и достижениях.

## Принципы управления знаниями

Должна формироваться и регулярно актуализироваться БЗ о подрядчиках, чтобы при возникновении потребности в быстром поиске подрядчика использова-

лись кандидаты из этой БЗ. Рекомендуется оценивать качество работ аутсорсеров и комфортность работы с ними команды проекта.

Должна формироваться и актуализироваться БЗ о выученных уроках проектов, содержащая хорошие и плохие практики и рекомендации по устранению последних. БЗ должна влиять на типовую WBS проектов компании и типовой реестр рисков компании при периодически выполняемом аудите БЗ выученных уроков. За каждый элемент БЗ должен быть назначен один ответственный. Должен существовать план аудита БЗ, которого следует придерживаться.

Рекомендуется иметь БЗ типовых проблем проектов и клиентов, где приводить описание ситуаций, которые вызвали сложности в коммуникациях с клиентом или в принятии решений. Ею нужно делиться с коллегами, чтобы при возникновении похожих ситуаций в следующих проектах менеджеры проектов могли управлять подобными рисками.

Результаты выученных уроков и информацию, которую по окончании проекта перенесли в корпоративную БЗ, нужно доносить до офиса управления проектами (РМО). Выученные уроки проекта фиксируются с самого начала работы над ним и доступны любому члену проектной команды. Никакие записи из уроков не удаляются. Следует подводить промежуточные результаты фиксации выученных уроков (по наступлении определенных вех проекта) и докладывать о них на РМО.

### 6.3.8. Паттерн «Стабилизация ПО»

Этот паттерн крайне полезен на стадии стабилизации ПО. В его основе лежит идея итерации баглога из методологии Agile.



На сайте книги, перейдя по ссылке [http://bit.ly/pmway\\_buglist](http://bit.ly/pmway_buglist), вы найдете реальный пример этого паттерна. В ситуации, когда вы не можете позволить себе полноценное управление проектом, примените паттерн «Экстремальная разработка» совместно с этим паттерном для более или менее эффективного управления проектом «на коленке». Но в целом практика отслеживания истории исправлений багов в рассматриваемом виде крайне полезна при любой методологии разработки, не только когда есть ограничения.

Цвета в табл. 6.8 читаются в соответствии с легендой (табл. 6.7).

**Таблица 6.7.** Градация легенды баглога

	Исправлено
	Исправлено, но не протестировано заказчиком
	Исправлено, но есть замечания
	Повторяющийся баг
	Критический баг
	Новый баг
	Не баг — баг был снят тестировщиком после пояснений

В основе паттерна лежит таблица вида, приведенного далее (табл. 6.8).

Таблица 6.8. Баг-лист и беклог. Инструмент управления

№	Краткое описание бага	Статус бага	Сценарий воспроизведения	Комментарий тестировщика	Номер билда, в котором исправлен	Номер билда, в котром появилась снова	Номер билда, в котором повторно исправлен	1,89	1,93	1,95	1,98	1,992	1,995	1,999
11	Не работают кнопки ли-стания ИБ на карточке ДП	Исправлено			1.98									
12	Программа при запуске принудительно завершает свою работу (№ 817, 853)	Исправлено			1.98									
13	При построении собственного маршрута, цифра 1 вылезает за пределы значка	Исправлено			1.98									
14	Переход к экрану ввода SMS при долгом зажатии кнопки Поиск	Исправлено		Исправлено на уровне прошивки от 17.07.2012, однако короткое нажатие теперь закрывает алерт	1.999	2.09	2.15							



Основная польза этого паттерна в том, что вы видите динамику исправления багов и визуально понимаете, каков общий статус релиза. Также вы понимаете, насколько высоки риски повторения багов и насколько они были серьезными и сложными, — по длине неисправленных отметок в графах с номерами версий (билдов).

Сейчас придется сделать краткое отступление, чтобы уточнить, что под релизом здесь понимается некоторая запланированная поставка ПО в рамках, скажем, этапа проекта или большой итерации. А версией называется выпуск версии (билда, сборки) для проверки в рамках этой итерации. В проекте, из которого взят пример, релизов было порядка семи, а версии поставлялись каждую неделю, начиная с третьего релиза. Это было сделано, чтобы более эффективно управлять стабилизацией ПО. И данная таблица очень помогла команде в выполнении этой задачи.

Посмотрите, как выглядит эта таблица на последней итерации проекта (рис. 6.1) (показан фрагмент — исправленные баги отрезаны), — неважно, что здесь не видно, что именно написано, пусть вас это не смущает. Важно то, что видно, какой процент багов повторялся из версии в версию. Причин этому могло быть две: или это крайне сложный баг, на который требуются значительные время и силы разработчиков и они не успевают его исправить в рамках итерации выпуска версии (напомню, это всего неделя), или причина бага неясна и разработчик не знает, что именно исправлять. И в том и в другом случае это зоны риска продукта. И вы, как менеджер, должны такие риски выявлять и управлять ими.

Вариант с загрузкой разработчиков работой над другой функциональностью не проходит, так как это именно фаза стабилизации продукта. Они не разрабатывают новую функциональность.

На основании анализа этой таблицы можно сделать вывод, что имеется два повторяющихся бага, один из которых (предпоследний) вдруг повторился в последних версиях. И второй, причину появления которого не удается установить уже 18 недель. Также обнаружены три новых бага, которые до сих пор не исправлены. И остается один критический баг, который также не исправлен в течение уже пяти недель.

Своевременное обнаружение и мониторинг картины стабилизации ПО с помощью этого паттерна позволили своевременно перенаправить ресурсы разработчика на решение проблем, наиболее критических с точки зрения начала внедрения продукта. В этом основная польза данного паттерна.



Раскрашивание бэклога — цветовое кодирование его элементов может дать настолько хорошее представление о состоянии проекта, что имеет смысл не ограничиваться цветовой индикацией дефектов, а распространить эту практику на весь бэклог. Основные идеи этой процедуры изложены в докладе одного из создателей RUP, Филиппа Кратчена, *What color is your backlog* (<http://www.youtube.com/watch?v=XWYBkmvNGrk>).

№	И	В	С	Д	У	Р	В	Н	Т	Г	К	Л	М	О	П	Q	R	S	T	U	V	W	X	Y	Z	aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak	al	am	an	ao	ap	aq	ar	as	at	au	av	aw	ax	ay	az	ba	bb	bc	bd	be	bf	bg	bh	bi	bj	bk	bl	bm	bn	bo	bp	bq	br	bs	bt	bu	bv	bw	bx	by	bz	ca	cb	cc	cd	ce	cf	cg	ch	ci	cj	ck	cl	cm	cn	co	cp	cq	cr	cs	ct	cu	cv	cw	cx	cy	cz	da	db	dc	dd	de	df	dg	dh	di	dj	dk	dl	dm	dn	do	dp	dq	dr	ds	dt	du	dv	dw	dx	dy	dz	ea	eb	ec	ed	ee	ef	eg	eh	ei	ej	ek	el	em	en	eo	ep	eq	er	es	et	eu	ev	ew	ex	ey	ez	fa	fb	fc	fd	fe	ff	fg	fh	fi	fj	fk	fl	fm	fn	fo	fp	fq	fr	fs	ft	fu	fv	fw	fx	fy	fz	ga	gb	gc	gd	ge	gf	gg	gh	gi	gj	gk	gl	gm	gn	go	gp	gq	gr	gs	gt	gu	gv	gw	gx	gy	gz	ha	hb	hc	hd	he	hf	hg	hh	hi	hj	hk	hl	hm	hn	ho	hp	hq	hr	hs	ht	hu	hv	hw	hx	hy	hz	ia	ib	ic	id	ie	if	ig	ih	ii	ij	ik	il	im	in	io	ip	iq	ir	is	it	iu	iv	iw	ix	iy	iz	ja	jb	jc	jd	je	jf	fg	jh	ji	jj	jk	jl	jm	jn	jo	jp	jq	jr	js	jt	ju	kv	kw	kx	ky	kz	la	lb	lc	ld	le	lf	lg	lh	li	lj	lk	ll	lm	ln	lo	lp	lq	lr	ls	lt	lu	lv	lw	lx	ly	lz	ma	mb	mc	md	me	mf	mg	mh	mi	mj	mk	ml	mm	mn	mo	mp	mq	mr	ms	mt	mu	mv	mw	mx	my	mz	na	nb	nc	nd	ne	nf	ng	nh	ni	nj	nk	nl	nm	nn	no	np	nq	nr	ns	nt	nu	nv	nw	nx	ny	nz	oa	ob	oc	od	oe	of	og	oh	oi	oj	ok	ol	om	on	oo	op	oq	or	os	ot	ou	ov	ow	ox	oy	oz	pa	pb	pc	pd	pe	pf	pg	ph	pi	pj	pk	pl	pm	pn	po	pp	pq	pr	ps	pt	pu	pv	pw	px	py	pz	qa	qb	qc	qd	qe	qf	qg	qh	qi	qj	qk	ql	qm	qn	qo	qp	qq	qr	qs	qt	qu	qv	qw	qx	qy	qz	ra	rb	rc	rd	re	rf	rg	rh	ri	rj	rk	rl	rm	rn	ro	rp	rq	rr	rs	rt	ru	rv	rw	rx	ry	rz	sa	sb	sc	sd	se	sf	sg	sh	si	sj	sk	sl	sm	sn	so	sp	sq	sr	ss	st	su	sv	sw	sx	sy	sz	ta	tb	tc	td	te	tf	tg	th	ti	tj	tk	tl	tm	tn	to	tp	tq	tr	ts	tt	tu	tv	tw	tx	ty	tz	ua	ub	uc	ud	ue	uf	ug	uh	ui	uj	uk	ul	um	un	uo	up	uq	ur	us	ut	uu	uv	uw	ux	uy	uz	va	vb	vc	vd	ve	vf	vg	vh	vi	vj	vk	vl	vm	vn	vo	vp	vq	vr	vs	vt	vu	vv	vw	vx	vy	vz	wa	wb	wc	wd	we	wf	wg	wh	wi	wj	wk	wl	wm	wn	wo	wp	wq	wr	ws	wt	wu	wv	ww	wx	wy	wz	xa	xb	xc	xd	xe	xf	fg	xh	xi	xj	xk	xl	xm	xn	xo	xp	xq	xr	xs	xt	xu	xv	xw	xx	xy	xz	ya	yb	yc	yd	ye	yf	yg	yh	yi	yj	yk	yl	ym	yn	yo	yp	yq	yr	ys	yt	yu	yv	yw	yx	yy	yz	za	zb	zc	zd	ze	zf	zg	zh	zi	zj	zk	zl	zm	zn	zo	zp	zq	zr	zs	zt	zu	zv	zw	zx	zy	zz
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																

Рис. 6.1. Инструмент стабилизации релиза — бэклог

В соответствии с его рекомендациями предлагается разбить все элементы бэк-лога проекта (фичи, User Story — неважно) на четыре основные категории:

- ◆ функциональные возможности разрабатываемой системы;
- ◆ реализация нефункциональных требований (базовая архитектура, внутренние фреймворки проекта);
- ◆ дефекты (баги, ошибки);
- ◆ технический долг.

Далее каждой категории присваивается свой цвет:

- ◆ зеленый — функциональные возможности системы (в виде User Stories, Use Case и т. д.);
- ◆ желтый — нефункциональные требования;
- ◆ красный — дефекты;
- ◆ черный — технический долг.

Может возникнуть вопрос, чем желтые элементы бэклога отличаются от черных? Мне очень нравится такая формулировка: черные элементы — это просто не сделанные вовремя желтые. Если в проекте небрежно реализуются функциональные требования, если не выделяется время на проработку архитектуры решения, бэклог почернеет, и надолго.

При формировании цветного бэклога можно руководствоваться следующими принципами.

- ◆ Деление элементов бэклога на видимые и невидимые. То есть на то, что можно явно продемонстрировать заказчику, и то, что скрыто внутри продукта и, если все хорошо, заказчика не волнует. В большинстве случаев заказчик не интересуется, насколько грамотно спроектирована архитектура. Если реализация новых требований легко вписывается в текущее решение, если исправление одного дефекта не порождает десятков новых, о слове «архитектура» заказчик, скорее всего, не вспомнит.
- ◆ Деление элементов бэклога на позитивные и негативные. Здесь все просто. «Позитив» — то, что просил заказчик: функционал, производительность и т. д. «Негатив» — то, что вызывает зубную боль у заказчика или команды: дефекты, код, требующий рефакторинга, громоздкий процесс сборки, из-за которого тестирование каждого билда превращается в проблему.

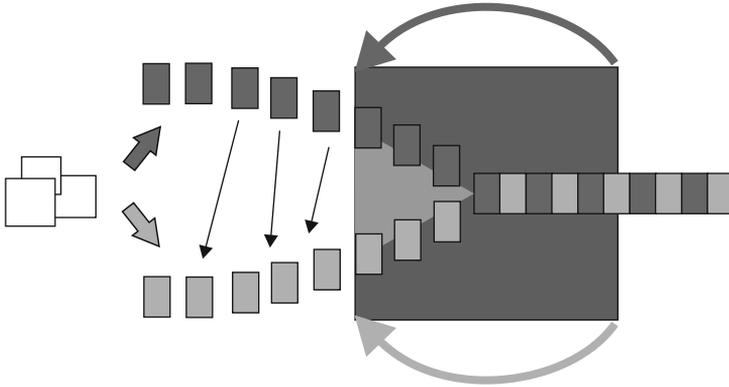
Это деление может быть представлено в виде матрицы  $2 \times 2$  (рис. 6.2).

	Видимые	Невидимые
Позитив	Видимые заказчику возможности	Архитектурные задачи (скрытые от заказчика)
Негатив	Дефекты	Технический долг

**Рис. 6.2.** Категории элементов бэклога в виде матрицы  $2 \times 2$

После того как элементы бэклога раскрашены — отнесены каждый к своей категории, можно определять последовательность реализации требований.

На начальном этапе проекта основными строительными кирпичиками бэклогов итераций являются зеленые элементы — как правило, из них строятся первые версии продуктов. Но пренебрежение желтыми задачами — верный путь к появлению технических проблем. С самого начала нужно закладывать в проект правильные архитектурные и инфраструктурные решения. Филипп Кратчен для этого использует метафору застежки-молнии (рис. 6.3).



**Рис. 6.3.** Застежка-молния бэклога

Рекомендуется чередовать функциональные и нефункциональные задачи, причем лучше всего, если между ними есть зависимости. Например, на функционал управления учетными записями пользователей системы сразу накладывается соответствующий механизм аутентификации.



В процессе разработки в системе неизбежно появятся «негативные» элементы — ошибки, необходимость рефакторинга, их также надо вписывать в план работ.

Цветной бэклог в описанной концепции использования цветов очень хорош тем, что позволяет поверхностным взглядом сразу оценить, нет ли перекоса в одну из сторон.

- ◆ В бэклоге сплошные зеленые элементы — высока вероятность того, что команда под внешним давлением штампует новый функционал, не задумываясь о растущих проблемах в архитектуре.
- ◆ Много желтых элементов подряд — взяли власть программисты и с увлечением разрабатывают фреймворки и автоматизируют сборку, а функционала нет и востребованность фреймворков неясна.
- ◆ Высокая плотность черных и красных элементов — индикатор того, что пришлось остановить реализацию функционала и заняться «лечением» проекта. Стоит хотя бы рассмотреть выученные уроки, выявляющие причины этой ситуации.

Принцип деления на цвета, изложенный Кратченом, не единственный, к тому же не обязательно использовать четыре цвета. Может, например, хватить трех:

- ◆ зеленого;
- ◆ красного;
- ◆ черный и желтый объединить в одну категорию — технические работы.

Другой вариант деления — сгруппировать элементы бэклога на основе анализа по методу Канона:

- ◆ обязательные (threshold);
- ◆ ожидаемые (performance);
- ◆ приводящие к восхищению (excitement).

Какой бы критерий раскрашивания вы ни выбрали, использование цветного бэклога даст всем заинтересованным лицам проекта отличное визуальное представление о том, над чем команда работает сейчас, какие проблемы могут возникнуть или в чем причины ряда имеющихся проблем.

### 6.3.9. Паттерн «С ног на голову, или ПМП ПСР»



---

Паттерн «С ног на голову, или ПМП ПСР (программа и методика проведения приемо-сдаточных работ)» может применяться или как фактор, «вытягивающий» представление о системе в случае, когда подрядчик не настроен на конструктив, проект в критическом пике и подрядчик не хочет ничего подробнее вам рассказывать и создавать дополнительные спецификации или диаграммы.

---

В моей практике был проект, в котором вроде и система работала, и документация была оформлена по ГОСТам 19-й и 34-й серии, но я не был уверен в том, что смогу достичь цели дальнейшего развития этой системы своими силами. Да и в том, что основные целевые алгоритмы системы были действительно работоспособными, а не «делали вид», я не был уверен. Тогда я применил данный паттерн и паттерн «Открытые вопросы». Плановмерно и тактично я начал объяснять подрядчику, что его задача — сдать мне систему, поэтому, несмотря на то что в договоре не сказано, что будут создаваться программа и методика испытаний, есть ряд критических моментов, поэтому я, как ответственный человек со стороны заказчика, должен получить ответы на свои вопросы, а также подтверждение работоспособности.

А так как в проектной документации на тот момент уже было целых три документа под названием «программа и методика испытаний» (ПМИ): для серверной части, мобильного приложения и веб-портала, то я придумал название рабочего документа «Программа и методика проведения приемо-сдаточных работ», в котором упомянул и про эти ПМИ.



---

На сайте книги, перейдя по ссылке [http://bit.ly/pmway\\_acceptancetesting](http://bit.ly/pmway_acceptancetesting), вы можете скачать документ «Программа и методика проведения приемо-сдаточных работ». Здесь же я приведу только те выдержки, которые показывают смысл документа.

---

**Целевая аудитория документа** — сотрудники заказчика и исполнителя, отвечающие за приемку работ по договору №11-11-111.

### Правила работы с документом

- ◆ Документ является рабочим инструментом управления проектом.
- ◆ Все правки в документ между версиями  $i.0$  (где  $i = 1, 2, \dots$ ) вносятся в режиме управления изменениями до согласования заказчиком очередной версии документа. Исправления в документ вносят и исполнитель, и заказчик. Изменения в документе принимает только заказчик.
- ◆ По мере работы над документом снимаются риски, открытые вопросы, корректируется раздел «Программа и методика проведения...».
- ◆ После согласования документа на его основе создается календарный план-график проведения приемочных испытаний, в соответствии с которым проводятся подготовка к приемке и приемка разработанного решения в рамках выполнения работ по договору № 11-11-111.

### Процедура приемки решения

- ◆ Создается приемная комиссия в составе трех представителей заказчика и двух представителей исполнителя.
- ◆ В ходе приемки решения фиксируются все результаты проведения приемки по настоящей программе.
- ◆ Все недостатки, выявленные в процессе приемочных испытаний, фиксируются в протоколе приемочных испытаний либо протоколе доработок — по решению приемной комиссии.
- ◆ Замечания, внесенные в протокол приемочных испытаний, подлежат устранению исполнителем в сроки, оговоренные в данном протоколе, до подписания акта приемки.
- ◆ Недостатки системы, не противоречащие техническому заданию и прочей технической документации, подлежат внесению в протокол доработок и после приемки системы будут устраняться исполнителем по дополнительным соглашениям к договору.
- ◆ По окончании приемки на финальном совете приемная комиссия принимает решение о том, насколько выявленные недостатки критичны с точки зрения финальной приемки на основании данных протокола приемочных испытаний.
- ◆ В случае положительного решения оформляется акт приемки системы в эксплуатацию.

### Краткое описание программы проведения прямо-сдаточных работ

Состав работ программы испытаний состоит из шести основных блоков: «тестирование (приемочных испытаний) решения», «приемка кода», «приемка пользовательской документации», «развертывание», «контроль качества выполнения проекта», «правовая экспертиза».

Цели блока «тестирование решения»:

- ◆ убедиться в соответствии решения требованиям (тестирование на соответствие ТЗ);

- ◆ убедиться в применимости решения для целей его создания (бизнес-тестирование);
- ◆ убедиться в работоспособности решения (тестирование алгоритма расчета прогноза);
- ◆ убедиться в выполнении требований к производительности и интеграции.

Методологически цели блока достигаются за счет определенных действий:

- ◆ цель «убедиться в соответствии решения требованиям (тестирование на соответствие ТЗ)» — за счет:
  - ◆ проверки покрытия (трассировки) всех требований тест-кейсами (испытаниями ПМИ);
  - ◆ итеративного тестирования требований от максимального приоритета («системное») к минимальному («обычное») с понижением порога допустимых ошибок:
    - первая итерация — тестирование требований с приоритетом «системное». Допустимый порог ошибок — 3 % от общего количества требований;
    - вторая итерация — тестирование требований с приоритетом «высокое». Порог — 10 %;
    - третья итерация — выполняется в рамках полного тестирования. Тестируются требования с приоритетом «обычное». Порог — 20 %;
- ◆ цель «убедиться в применимости решения для целей его создания (бизнес-тестирование)» — за счет:
  - ◆ покрытия (трассировки) шагов бизнес-кейса (варианта использования решения) бизнес-потребителем (business actor) тест-кейсами (испытаниями GVB);
  - ◆ отсутствия критических багов в сценариях;
- ◆ цель «убедиться в работоспособности решения (тестирование алгоритма расчета прогноза) и убедиться в выполнении требований к производительности и интеграции» — за счет:
  - ◆ логирования требуемой информации в ходе работы алгоритма;
  - ◆ поставки сценария с необходимыми скриптами для получения отчета силами заказчика.

Цели блока «приемка кода»:

- ◆ убедиться в покрытии (трассировке) всех основных функций/требований кодом;
- ◆ убедиться в реализации требований по масштабируемости и адаптации к вводу внешних данных (критическое бизнес-требование) в архитектуре и системном дизайне;
- ◆ принять код со всеми необходимыми для его дальнейшего самостоятельного развития документированием, пояснениями и комментариями.

Методологически цели блока достигаются за счет:

- ◆ проведения ревью кода;
- ◆ запроса и анализа спецификаций уровня системного дизайна у разработчиков.

Цели блока «развертывание»:

- ◆ убедиться в принципиальном разворачивании решения;
- ◆ научиться самостоятельно разворачивать и настраивать для работы решение с использованием полученных дистрибутивов без участия исполнителя;
- ◆ провести интенсивную опытную эксплуатацию системы на предмет отработки бизнес-кейсов ее использования и попытки вывести решение из работы.

Методологически цели блока достигаются за счет:

- ◆ обучения;
- ◆ проведения испытаний развернутого решения.

Цель блока «контроль качества выполнения проекта» — убедиться в положительной динамике стабилизации решения по отчетам исполнителя.

Методологически цель блока достигается за счет представления отчетов о тестировании и сопроводительных отчетов к билдам с положительной динамикой стабилизации и корреляции.

Цель блока «правовая экспертиза» — убедиться в исключительности прав исполнителя на передаваемое заказчику решение.

Методологически цель блока достигается за счет проведения экспертизы представленного решения на чистоту авторских прав.

Все цели перечисленных блоков должны быть достигнуты при приемке.

Методики достижения целей обсуждаемы и могут быть изменены по согласованию сторон.

### Открытые вопросы

Заданные заказчиком и исполнителем открытые вопросы по проведению приемо-сдаточных работ вносятся в таблицу (см. далее) любой из сторон по мере подготовки и согласования программы и методики проведения приемо-сдаточных работ.

Решение готово к приемке по факту установки заказчиком статуса «закрыт» для всех вопросов.

Дата	Вопрос	Ответ	Статус
20.03.2013	Для понимания архитектуры системы и ее системного дизайна прошу предоставить sequence diagram (диаграмму последовательности) запуска серверной части системы. Или указать, где это задокументировано		Открыт

Впоследствии открытые вопросы были выделены в отдельный проектный артефакт. Сама программа и методика проведения приемо-сдаточных работ разрабатывались и согласовывались непросто, потому что подрядчик не был заинтересован в оперативных ответах и выполнении хоть каких-то дополнительных действий. Он был уверен, что все, что нужно, он уже сделал.

Раздел «Контроль качества выполнения проекта» был крайне полезен для начала разговора о том, что я не вижу никаких свидетельств того, что в проекте должное внимание уделялось качеству продукта. Я не вижу ни отчетов о тестировании, ни сопроводительных записок к билдам/поставкам версий, ни инструментов типа паттерна «Стабилизация ПО», описанного чуть раньше. Как я могу быть уверен в том, что вы, как проектная команда, сделали все для получения качественного продукта и что продукт действительно качественный?

Дальше шаг за шагом мы продвигались вперед. Главное здесь — не терять терпения и стараться как можно спокойнее объяснять, зачем нужны такие тесты и именно такие проверки, что это дает и какие мои риски как заказчика это снимает.

А в некоторых случаях, конечно, надо идти на уступки. Собственно, я часто стараюсь выдвинуть 110 % требований, 10 % из которых не критичны. Ими я охотно пожертвую, получив возможность говорить: я иду вам навстречу — давайте делать это вместе. Хотя, как вы понимаете, нижняя планка здесь где-то 80 %. Но важен не количественный показатель процента уступок, а то, что есть проверки, которые ни при каких обстоятельствах не могут быть не проведены.

Следует заметить, что этот паттерн может применяться не только в ситуации «с ног на голову», но и при нормальной, прямой и последовательной, разработке. Я бы рекомендовал в качестве приложения к договору оформлять примерно такой ПМП ПСР. Это значительно облегчит жизнь и заказчику и подрядчику, так как обеспечит ясные и четкие правила игры и однозначно трактуемое и одинаковое понимание конечной цели.

### 6.3.10. Паттерн «План управления документами»

План управления документами (ПУД) регламентирует управление поставками и степень готовности документов по завершении этапов производства ПО. ПУД содержит в себе реестр артефактов и документов, которые должны быть в состоянии Draft или Release к определенным моментам жизненного цикла.



ПУД является стержнем коммуникаций в проекте и должен согласовываться всей проектной командой.

Несмотря на то что паттерн называется «План управления документами», на самом деле речь идет о всевозможных проектных артефактах: документах, спецификациях, рабочих инструкциях, моделях, диаграммах, отчетах и т. п.

Как инструмент менеджера, ПУД позволяет определить все объекты поставки в проекте, их назначение и время жизни, момент необходимости, ответственных за создание/качество/согласование и утверждение объекта поставки. Если в любом проекте воспользоваться этим паттерном и создать ПУД, то, как правило, риски выпустить не то, что надо, или то, что надо, но не в те сроки, стремятся к нулю.



Структура этого документа состоит из трех блоков.

### Идентифицирующий блок информации

№ артефакта	Тип артефакта	Категория	Вид артефакта/документа	Расположение документа	Описание артефакта/документа
-------------	---------------	-----------	-------------------------	------------------------	------------------------------

Здесь:

#### ◆ тип артефакта:

- ◆ документ;
- ◆ модель;
- ◆ проект требования;
- ◆ прототип;
- ◆ спецификация;

#### ◆ категория:

- ◆ проектное управление;
- ◆ аналитика;
- ◆ архитектура;
- ◆ GUI-дизайн;
- ◆ тестирование;

#### ◆ вид артефакта/документа:

- ◆ заявка на создание продукта/услуги;
- ◆ устав проекта;
- ◆ бизнес-требования и бизнес-правила;
- ◆ концептуальная модель системы (Conceptual Model) и т. д.;

- ◆ **описание артефакта/документа** — краткое описание назначения и состава артефакта, например, для спецификации бизнес-требования и бизнес-правил таково: «Документ содержит описание всех выявленных в процессе сбора и анализа бизнес-требований и бизнес-правил. В документе описывается, каким образом разрабатываемая система связана с достижением бизнес-целей и что она должна для этого делать».

**Блок, представляющий фазы (этапы) жизненного цикла разработки**

PA	BA	CD	UCM	UCA	SA	FD	TD	D	DEV
----	----	----	-----	-----	----	----	----	---	-----

Здесь PA – Preliminary Analysis, BA – Business Analysis, CD – Conception Design, UCM – Use Case Modelling, UCA – Use Case Analysis, SA – System Analysis, FD – Functional Design, TD – Test Design, D – System Design, DEV – Development. В ваших проектах будут какие-то иные фазы, возможно, их будет меньше или больше, это неважно. Важно то, что с помощью этого паттерна вы сможете эффективно организовать артефакты проекта и управлять ими, в любой момент получая срезы, информирующие о том, какие артефакты в каком состоянии должны быть к моменту окончания конкретных фаз.

**Блок, непосредственно регламентирующий работу со спецификациями и проектными документами**

Ответственный	Состояние	Плановая дата завершения	Список проверяющих (reviewers)	Список согласователей	Список утверждающих лиц
---------------	-----------	--------------------------	--------------------------------	-----------------------	-------------------------

Внешний вид ПУД приведен на рис. 6.4.



ПУД может создаваться в MS Excel. Стандартная функция MS Excel Автофильтр позволяет получать различные представления информации ПУД. Наиболее полезное представление — срез проектных артефактов и их состояний к окончанию каждой фазы проекта. Также очень полезные представления можно сформировать по состоянию артефактов, ответственному, списку согласующих и утверждающих лиц.

Например, применив фильтр Показывать только значения RELEASE к столбцу CD, мы увидим, какие проектные артефакты должны быть закончены и утверждены к окончанию фазы Conceptual Design (рис. 6.5).



В заключение скажу, что хороший план управления коммуникациями, который является частью плана управления проектом, разрабатывается менеджером проекта после того, как разработан ПУД. Очень важно определить правильных проверяющих, согласующих и утверждающих лиц для каждого типа проектных документов. Ошибки в определении правильных лиц в этих списках влияют на качество конечного продукта.

ПУП и ПУД – два главных управленческих артефакта в проекте. Именно они полностью обеспечивают стратегию и тактику управления всеми аспектами проекта. При этом ПУД может выглядеть и гораздо проще, но даже фиксация объектов поставки простым списком, как в табл. 6.9, позволяет навести порядок в едином понимании обязательств по составу и срокам поставки объектов поставки.



A	C	U1	U2	S	F	TC	D	DE	Описание	Состояние	Планируемая длительность	Список проверяемых элементов (reviewers)	Список соответствий	Список утверждающих лиц
									Создание элементов			Присылать уведомление	Бизнес-департмент, Особый департаментальный гвард проект	Присылать уведомление
									Минишар проекта			Бизнес-департмент, департаментальный гвард проекта	Особый департаментальный гвард проекта	Департамент проекта, Системный проект, Минишар проекта
	RELEASE								Системный департамент			Минишар проекта	Минишар проекта, Особый департаментальный гвард проекта	Минишар проекта
	RELEASE								Системный департамент			Бизнес-департмент, минишар проекта	Бизнес-департмент, минишар проекта, Особый департаментальный гвард проекта	Департамент проекта, Системный проект
Draft	RELEASE								Минишар проекта			Бизнес-департмент, системный архитектор, тест-инженер	Бизнес-департмент, системный департамент, системный архитектор, тест-инженер	СШ проекта
Draft	RELEASE								Системный департамент			Системный департамент	Минишар проекта, Системный архитектор, Бизнес-департмент	не требуется утверждение
Draft	RELEASE								Системный департамент			Минишар проекта, системный архитектор, системный департамент	Минишар проекта, системный архитектор, Бизнес-департмент	СШ проекта
Draft	RELEASE								Минишар проекта			Бизнес-департмент, системный архитектор, тест-инженер	Бизнес-департмент, системный департамент, системный архитектор, тест-инженер	СШ проекта
Draft	RELEASE								Архитектор			Системный департамент, минишар проекта	Бизнес-департмент, минишар проекта, системный департамент, тест-инженер	Системный архитектор, Бизнес-департмент, минишар проекта
Draft	RELEASE								Бизнес-департмент			Минишар проекта, системный архитектор, системный департамент	Минишар проекта, системный департамент, системный архитектор	СШ проекта (минишар проекта, Бизнес-департмент, минишар проекта, системный архитектор, системный департамент, тест-инженер, системный проект)
	RELEASE								Минишар проекта			Бизнес-департмент, департаментальный гвард проекта	Минишар проекта, Особый департаментальный гвард проекта	Департамент проекта, Системный проект
		Draft	RELEASE						Проверка показателя-стагнация			Системный департамент, минишар проекта	Бизнес-департмент, минишар проекта, системный департамент, тест-инженер	СШ проекта
Draft	Draft	RELEASE							Системный департамент			Системный департамент	Системный департамент	Системный архитектор, минишар проекта
			RELEASE						Архитектор			Системный департамент	Минишар проекта	Системный департамент, Системный архитектор, Бизнес-департмент
		Draft	Draft	RELEASE					Системный департамент			Системный департамент, проверка показателя-стагнация	Минишар проекта, проверка показателя-стагнация, тест-инженер	Бизнес-департмент, проверка показателя-стагнация, тест-инженер
Draft	Draft	Draft	Draft	RELEASE					Системный департамент			Системный департамент, проверка показателя-стагнация, системный архитектор	Минишар проекта, системный архитектор	Бизнес-департмент

вид ПУД

## План управления документами

№ артефакта	Тип артефакта	Категория	Тип артефакта / документа	PA	BA	CD	UCM	UCA	S
5	Документ	Проектное управление	Work Breakdown Structure. Иерархическая структура работ.	Draft	Draft	RELEASE			
6	Модель	Аналитика	Модель типов требований	Draft	Draft	RELEASE			
7	Документ	Проектное управление	Requirement Management Plan. План управления требованиями.	Draft	Draft	RELEASE			
8	Документ	Проектное управление	Analytical Work Breakdown Structure. Иерархическая структура аналитических работ.	Draft	Draft	RELEASE			
9	Спецификация	Архитектура и реализация	Концепция системы	Draft	Draft	RELEASE			
10	Спецификация	Аналитика	Концепция создания и развития продукта		Draft	RELEASE			
11	Документ	Проектное управление	Project Scope Statement. Описание содержания проекта.			RELEASE			

Рис. 6.5. ПУД. Представление срезов

Таблица 6.9. ПУД как реестр документов

№	ИД документа	Наименование документа	Тип документа	Статус документа	Дата изменения
1	605.23767.4.25590.001.ЭД.01.01	Ведомость документов	Спецификация	Утвержден	21.09.12
2	605.23767.4.25590.001.ИЗ.01.01	Руководство пользователя	Руководство	Утвержден	26.12.12
3	605.23767.4.25590.001.ПГ.01.01	Описание технологического процесса обработки данных	Описание	Утвержден	21.09.12

Тем не менее важной изюминкой этого паттерна я считаю именно привязку к следующим вехам проекта.

PA	BA	CD	UCM	UCA	SA	FD	TD	D	DEV
----	----	----	-----	-----	----	----	----	---	-----

А	FD	TD	D	DEV	Ответственный	Состояние	Список проверяющих (reviewers)	Список согласователей	Список утверждающих лиц
					Менеджер проекта		Бизнес-аналитик, системный аналитик, системный архитектор, тест-менеджер	Бизнес-аналитик, системный аналитик, системный архитектор, тест-менеджер	ССВ проекта
					Системный аналитик		Системный аналитик	Менеджер проекта, Системный	не требуется утверждение
					Системный аналитик		Менеджер проекта, системный архитектор, системный аналитик	Менеджер проекта, Системный архитектор,	ССВ проекта
					Менеджер проекта		Бизнес-аналитик, системный аналитик, системный архитектор, тест-	Бизнес-аналитик, системный аналитик, системный	ССВ проекта
					Архитектор		Системный аналитик, менеджер проекта	Бизнес-аналитик, менеджер проекта, системный	Системный архитектор, Бизнес-аналитик, менеджер
					Бизнес-аналитик		Менеджер проекта, системный архитектор, системный аналитик	Менеджер проекта, системный архитектор, системный аналитик	ССВ проекта (менеджер проекта, Бизнес-аналитик, менеджер продукта, системный архитектор,
					Менеджер проекта		Бизнес-аналитик, заинтересованные лица проекта	Менеджер проекта, Основные заинтересованные	Заказчик проекта, Спонсор проекта

по этапам/вехам проекта

Если в вашем ПУД есть эти столбцы, то команда вынуждена активно решать вопрос о возможности выпуска версии или релиза объекта поставки к нужной вехе, обсуждать целесообразность и нужность данного объекта в проекте к этому моменту, выявлять риски, связанные с возможностью выполнить эти обязательства.

### 6.3.11. Плохие паттерны

В этом разделе я бы хотел привести очень короткие примеры плохих паттернов, которые я встречал в своей практике.

**Антипаттерн «Сам себе режиссер».** Менеджер проекта «закрылся» в проекте, не сообщая о результатах ни заинтересованным лицам, ни проектному офису. В итоге он «изобрел» уникальный подход к разработке ПО и управлению проектом, противоречащий всем принятым в компании методикам. Возможно, это бы и сработало, но квалификация персонала не соответствовала придуманным

процессам. Так что сроки сорваны. Авторитет у команды потерян. Проект в кризисе.

**Антипаттерн «Я лучше знаю, как надо!».** Один из спонсоров проекта, пользуясь своим положением, постоянно продавливал принятие нужных ему решений проектной командой. Попытки провести фокус-группы приводили к тому, что было сказано: «Они все дураки, я все равно знаю, что им (конечно-му пользователю) надо». В силу занятости описать, что же все-таки надо, этот специалист так и не смог. В итоге сроки сорваны. Авторитет утрачен. Проект в кризисе.

**Антипаттерн «Мы все здесь команда, или У семи нянек...».** Менеджер постоянно мотивирует всю команду так: «Давайте вместе погенерируем идеи, вместе поработаем над документом. Все пишете то, что думаете, а потом присылайте мне. Я скомпилирую». В итоге нет никакого ресурсного управления. Нет понимания границ полномочий. Нет ответственных ни за что. Проекты непрозрачны, выполняются аврально. Команды нет. Лидера нет. Задачи «виснут», артефакты «проваливаются по качеству».

Думаю, что вы без труда вспомните антипаттерны сами, я привел те, которые, на мой взгляд, являются самыми критическими с точки зрения управления проектами. Если вы видите такие антипаттерны в своем или другом проекте, идите за советом к руководителю проектного офиса или руководителю разработки ПО. Ищите выход из такого положения.

Мы с вами подошли к концу раздела «Паттерны управления проектом». Подводя итоги, я хотел бы задать риторический вопрос...

### 6.3.12. Так что же такое управление проектом?

Думаю, что по прочтении этой главы у читателя мог возникнуть вопрос: паттерны — это, конечно, здорово, но что же такое управление проектом? Неужели нет ничего объединяющего эти паттерны?

Есть, конечно. Есть ясные и четкие методологии, которые увязывают воедино то, о чем мы с вами говорим, но в некоторых проектах их применение просто невозможно из-за среды проекта, а в некоторых это просто нецелесообразно или дорого. С течением времени, участвуя в проектах, вы приобретете собственное понимание того, что такое управление проектом, и оно у вас будет постоянно меняться...



---

Моя формулировка на сегодняшний день такова: «Управление проектом — это искусство смешивать нужный “коктейль” из науки управления, методик управления, своих и чужих навыков, ХП и ВУ, оптимально и эффективно удовлетворяющий заказчика в конкретной и уникальной среде проекта за счет приложения и использования собственной внутренней силы, жизненной зрелости, личностных, менеджерских и лидерских способностей и качеств».

---

## 6.4. Паттерны управления функциональным отделом

В данном разделе давайте поговорим об управлении функциональным отделом и о паттернах управления, которые могут успешно применяться для управления функциональным подразделением.

В отличие от управления проектом, направленного на достижение конкретных целей, как правило связанных с созданием нового продукта или развитием уже существующего продукта, когда вы, как управленец, руководите разнородной командой, включающей в себя разные роли в ЖЦ разработки ПО (аналитиков, архитекторов, разработчиков, тест-менеджеров, тестировщиков), при управлении функциональным отделом вы управляете ресурсным пулом какой-то одной роли из перечисленных.

В случае если вы работаете в проектоориентированной организации, ваше подразделение является ресурсным пулом для проектов и, соответственно, ваша задача — эффективное удовлетворение проектов в ресурсах с заданными качеством работы и уровнем квалификации. Однако в любом случае вы управляете проектной средой. Собственно, в данном случае вы во многом формируете эту среду. Непосредственно личным участием в ревю/согласованиях/утверждениях документов, разрешением конфликтов, совершенствованием процессов и процедур в области деятельности вашего подразделения или опосредованно — через образование и повышение квалификации ваших подчиненных.

Очень часто участие в проектной деятельности через образование и повышение квалификации подчиненных является основной задачей руководителя функционального подразделения.

Далее перечислю уже упомянутые в этой книге паттерны, которые, на мой взгляд, должен знать и уметь применять руководитель функционального отдела.

- ◆ Паттерн исследования и формирования бизнес- (проектной) среды: «Управление средой проекта».
- ◆ Паттерны операционной деятельности:
  - ◆ «Базовые правила работы»;
  - ◆ «План управления документами»;
  - ◆ «Система авторизации работ»;
  - ◆ «Итеративная разработка ПО»;
  - ◆ «Обеспечение качества и анализ рисков»;
  - ◆ «Зажарка в СУП, или Универсальные принципы управления проектами».

- ◆ Паттерны процессного совершенствования:
  - ◆ «Выученные уроки»;
  - ◆ «Хорошие практики»;
  - ◆ IDEAL (применительно к совершенствованиям в отделе);
  - ◆ «Система управления производством ПО (СУПр)» (в части функциональной СУ СУПр, относящегося к компетенциям его отдела).
- ◆ Паттерны управления персоналом/командообразующие паттерны:
  - ◆ «Информационный бюллетень»;
  - ◆ «Смартинг»;
  - ◆ «Чек-листинг».

### 6.4.1. Паттерн «Фундамент отдела»

Этот паттерн относится к паттернам формирования среды, но я сознательно вынес его в данный раздел книги, посвященный паттернам руководителя функционального отдела. Потому что именно от того, насколько хорошо применен и применен ли вообще этот паттерн в отделах, зависят общая корпоративная культура и то, насколько жизнеспособны любые производственные процессы в организации.

Без этого фундамента никакая процессная модель в компании не будет работать в принципе. И руководитель более высокого звена не может построить этот фундамент. С одной стороны, он слишком «далек от народа» и не может держать руку на пульсе сотрудников в силу занятости задачами стратегического и тактического уровня. С другой стороны, количество сотрудников, которых можно эффективно контролировать, согласно Г. Минцбергу, равно  $7 \pm 2$ . То есть максимум девять человек могут эффективно работать, когда применяется модель прямого управления.

К тому же у руководителя разработки ПО много задач стратегического уровня: стратегия развития департамента, стратегия развития продукта, управление портфелем проектов, управление бюджетом, управление процессами и т. д., — которые не способствуют погружению в задачи тактического и операционного уровня без ущерба качеству стратегического уровня. Вы, наверное, знаете, что происходит, когда руководитель производственного центра разрывается между стратегическим и тактико-операционным уровнем. То, что видел я, никогда не приносило пользу компании. В отсутствие эффективных и сбалансированных стратегий современный бизнес похож на лодку без руля и без капитана, которая движется рывками в зависимости от перевеса той или иной части команды в спорах. И не всегда эта лодка плывет туда, куда прокладывает маршруты капитан бизнес-эскадры (генеральный директор).

С учетом сказанного резюмирую: именно задача формирования фундамента отдела, через который формируется вся корпоративная культура, — основная задача руководителя функционального отдела.



---

Для формирования фундамента отдела, как и для формирования благоприятной среды для выполнения проекта, самое главное — определить зоны и ответственность участников и заинтересованных лиц, что поможет им эффективно выполнять свою работу и принимать решения, основанные на прозрачных и оптимальных принципах и правилах коммуникации.

---

Для того чтобы применить этот паттерн, крайне важно разобраться в том, что представляет собой организация, а для этого нужно в первую очередь применить паттерн «Базовые правила работы» и обратиться к книге Г. Минцберга «Структура в кулаке: создание эффективной организации» [12]. При этом в базовых правилах работы вы, как руководитель, чья основная задача — создание единого фундамента вашей функциональной единицы, обязательно должны прописать принципы, относящиеся к стандарту профессиональной деятельности. Эти стандарты должны быть понятны всем сотрудникам отдела. Там должны быть указаны уровень квалификации, требования к квалификации в зависимости от уровня, зачем и кому это надо, как можно развивать свои компетенции и т. д.



---

Вторым (на самом деле первым, но не всегда его надо создавать именно первым — часто важнее применить сначала базовые правила, чтобы наладить хоть отчасти прозрачную работу, а потом уже заниматься стратегией) важнейшим артефактом этого паттерна является стратегия/концепция развития отдела. Вы не сможете разработать этот документ, если у вас с руководителями других функциональных отделов и руководителем разработки ПО не будет единого понимания того, какова стратегия развития всего подразделения по разработке ПО в целом и какие цели стоят перед вашим отделом. Поэтому в ходе работы над этим документом вы будете вынуждены применить такие паттерны, как «Управление проектной средой» и «Ромашка Ананьиной — Минцберга». В результате вы сформируете и утвердите у руководителя разработки ПО стратегию/концепцию развития вашего отдела. Важнейшие аспекты этого документа — стратегия развития профессиональных компетенций, развитие базы знаний и системы менеджмента качества (QMS) отдела (инструкций, ХП, ВУ, процедур, шаблонов артефактов).

---

Следующая составляющая паттерна, за которую вы отвечаете и которая является наиболее значительной практической составляющей для ваших сотрудников, — это БЗ отдела. Именно вы отвечаете за то, чтобы сотрудник знал, где найти принятые ХП, куда записать ВУ, как с этим работать, как организовано хранилище знаний.



---

Следующим вы выполняете паттерн WAS. Для этого создаете систему авторизации работ и — если вы работаете в проектной организации — интегрируете ее с WAS PMO. WAS крайне важна для обеспечения прозрачности работы сотрудников вашего отдела, для того, чтобы вы понимали, чем они заняты, каково качество их работы, для постоянного совершенствования точности оценок и, как следствие, постоянного повышения точности планирования работ,

выполняемых отделом. Для этого WAS должен включать в себя два модуля: модуль оценки трудозатрат по задачам и модуль timesheets — отрезок времени, реально затраченного на выполнение задач. От вас зависит, как облегчить жизнь подчиненным и понизить риски недостаточного качества выполнения работ, — для этого достаточно ввести понятие «типовая задача» и определить WBS ее выполнения, которая и оценивается на стадии оценки трудоемкости. В дальнейшем по этим же оценкам оцениваются фактические трудозатраты. Крайне важно довести до ваших сотрудников, что эти данные нужны вам не для наказания, а в первую очередь для получения прозрачной картины работы отдела в целом и повышения точности планирования.

Затем вы формируете для каждого сотрудника план персонального развития. Вы должны быть психологом, наставником и лидером, к которому идут за советом. Вы должны быть учителем и старшим товарищем, для того чтобы постоянно растить нужные компетенции у подчиненных.

Исходя из построенных персональных планов развития, вы формируете план курсов и тренингов. Часть из них проведете вы сами, часть представите для утверждения руководителю разработки, чтобы он включил их в бюджет.



Разработка шаблонов и процедур — тоже ваша прерогатива. Именно вы отвечаете за создание системы менеджмента качества (QMS) отдела, за то, насколько легко вашим подчиненным будет включаться в проекты, как быстро они смогут найти ответы на вопросы наподобие: «Где взять шаблон “концепция системы” для проекта типа “экстремальный”?» На сайте книги, перейдя по ссылке [http://bit.ly/pmway\\_qms](http://bit.ly/pmway_qms), вы найдете презентацию о том, что такое QMS в целом, и сможете более четко сформулировать для себя какие-то принципы.

Выполнив все это, вы получите актуальный на дату выполнения фундамент своего отдела. Дальше постоянно, ежедневно поддерживаете этот фундамент в актуальном для достижения целей состоянии. А на основании утвержденной стратегии/концепции развития отдела начинаете несколько проектов по совершенствованию процессов, созданию инфраструктуры отдела и т. п.

Как видите, руководитель функционального отдела — это голова и руки руководителя подразделения по разработке ПО какого-то функционального направления. Именно на вас, руководителях среднего уровня, базируется главный фактор успеха любого современного проекта — качество персонала.

Это основной паттерн руководителя. Данный раздел не очень большой, и, если вы освоите и примете к сведению хотя бы только этот паттерн, я буду считать цель книги достигнутой.

Я желаю вам стать профессионалами в этом сложном процессе и достичь вершины лидерства, когда каждый ваш подчиненный будет считать своей заслугой то, как он самостоятельно (!) профессионально вырос в компании за последний год и каким специалистом стал.

## 6.4.2. Паттерн «Удаленное управление»

В современных компаниях часто работает распределенная команда. Развиты такие формы трудовых отношений, как аутсорсинг, аутстаффинг, удаленная команда, распределенная команда. В данном разделе я разберу простой паттерн работы с распределенной командой, который неоднократно применял сам.

Говорят, что если команда не собирается очно, то эффективность ее работы в разы меньше по сравнению с эффективностью работы той, которая очно собирается. Это и верно и неверно. На самом деле очень многое зависит от того, насколько хороши и информативны коммуникации в команде.

Я сторонник позиции, согласно которой распределенная команда может работать так же хорошо, как и команда, находящаяся «под крылом». Я считаю, что очные встречи исключительно полезны только в трех случаях: чтобы познакомиться (и конечно, kick-off meeting), провести мозговой штурм ну и... отпраздновать успешное окончание проекта.

Как же добиться эффективной работы распределенной команды?

Мой рецепт таков.

1. Паттерн WAS.
2. Паттерн «Смартинг».
3. Паттерн «Чек-листинг».
4. Специфика виртуальных встреч:
  - ◆ любая встреча с сотрудниками проводится с использованием двух ноутбуков — чтобы на первом ноутбуке видели меня самого, а на втором — то, что я рисую. Придется более активно иллюстрировать свои идеи — вместо флипчарта и доски вполне можно использовать обычный Paint. Да, это потребует некоторого навыка работы с мышью, но его приобретение — это вопрос двух встреч;
  - ◆ при проведении видеоконференции в идеале желательно видеть видео всех участников совещания и иметь возможность передать флипчарт любому из них;
  - ◆ протокол пишется всегда;
  - ◆ протокол всегда пишет та сторона:
    - которой ставили задачу;
    - где больше удаленных сотрудников;
  - ◆ протокол пишется всегда! Даже если это четыре строки;
  - ◆ протокол пишется максимально коротко, но информативно;
  - ◆ рассылается участникам совещания он в течение двух часов после его проведения;
  - ◆ все обязаны поправить протокол или явно одобрить его.

### 5. Адаптированные базовые правила:

- ◆ тим-лидеры доступны по почте всегда. Отвечают в течение часа;
- ◆ любой член команды доступен по «Скайпу» в рабочее время;
- ◆ три итерации дискуссии проходят в переписке, после чего дискуссия продолжается по «Скайпу»;
- ◆ если почта и «Скайп» не работают, любой член команды доступен по мобильному телефону;
- ◆ ответственность за качество и правильность понимания переданной информации лежит и на принимающей стороне («Я правильно понял, что...», «Под этим и тем имеется в виду то-то и то-то...» и т. п.), и на передающей («Подтвердите, что принцип передачи данных понятен...», «Ответьте, какие шаги вы собираетесь предпринять для снятия рисков...»).

Естественно, необходимым условием для того, чтобы этот паттерн работал, является открытость и коммуникабельность участников команды. Также он лучше работает, если команда организована «кустом», то есть у вас есть группы удаленных сотрудников и руководители этих групп.

Почему я поместил этот паттерн в раздел «Паттерны управления функциональным отделом»? Во-первых, потому, что если компания работает с аутсорсинговыми командами на постоянной основе, то вы с большой вероятностью и будете руководителем удаленной группы в формате «куста» для менеджера проекта или руководителя разработки ПО. Ну а во-вторых, потому, что, если даже ваш отдел виртуальный и не все сотрудники физически находятся рядом, вы все равно должны строить единый фундамент отдела. Ну а чтобы это было возможно, и нужен этот паттерн.

## 6.4.3. Паттерн «Маслоу по Фрейду»

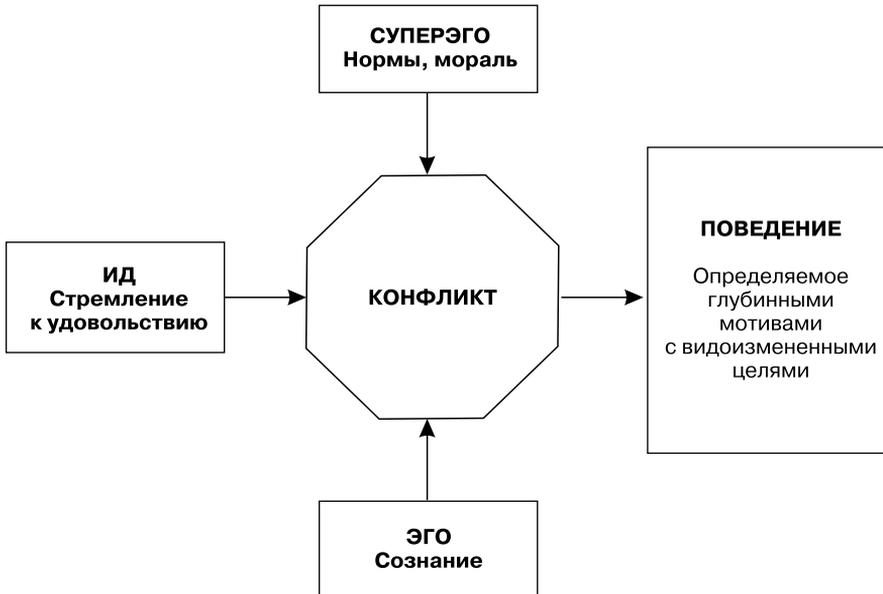
Название этого паттерна опять-таки шутовское, но я думаю, что благодаря этому вы хорошо запомните его суть. А она проста — помнить о пирамиде потребностей Маслоу и применять к ней понимание личностных мотиваций Фрейда.



Звучит запутанно, но на самом деле все крайне просто. Попробую объяснить.

Психоаналитическая теория Фрейда говорит о том, что *«поведение человека есть результат взаимодействия глубинных стремлений человека (ИД) и воспринятых индивидом социальных воздействий извне (социальных норм, морали и т. д.) — СУПЕРЭГО. При этом оба существенных компонента личности оперируют в ее подсознании. ЭГО управляет разрешением конфликта требований ИД и моралистических запретов СУПЕРЭГО»*.

В нотации, чем-то похожей на BPWIN, этот постулат отображен, как показано на рис. 6.6.



**Рис. 6.6.** Психоаналитическая теория Фрейда

То есть при входных данных «Стремление к удовольствию» деятельностью «Конфликт» управляет ресурс «Эго» под управляющим воздействием «Суперэго». Возможно, теперь стало немного понятнее.

Если исходить из сказанного, то в рамках управления персоналом для поиска индивидуального стиля развития сотрудника встает задача воспитания «Суперэго», развития аналитических навыков «Эго» и выяснения мотиваций ИД. Это позволит эффективно направлять мотивацию ИД посредством воздействия на «Эго» и «Суперэго». Этот паттерн и показывает эффективное влияние на мотивацию ИД посредством воздействия на «Эго» и «Суперэго» применительно к пирамиде Маслоу и информационному полю руководителя функционального отдела (табл. 6.10).

**Таблица 6.10.** Мотивация ИД посредством «Эго» и «Суперэго»

Мотивация ИД (по Маслоу)	Влияние на аналитическое сознание («Эго»)	Воспитание норм и морали («Суперэго»)
<b>1</b>	<b>2</b>	<b>3</b>
Физиологические (органические потребности): <i>голод, жажда, половое влечение и т. д.</i>	Обеды = меня уважают. Зарплата = результат моего труда	Оплачиваемые обеды = лояльность компании. Зарплата = польза компании
Потребность в безопасности: <i>стремление чувствовать себя защищенным, избавиться от страха и жизненных неудач</i>	Честные правила игры. Прозрачность. Коэффициенты трудового участия. Аттестация	КЗОТ

Продолжение ⇨

Таблица 6.10 (продолжение)

1	2	3
Потребность в принадлежности и любви: стремление <i>принадлежать общности, находиться рядом с людьми, быть признанным</i> ими и принятым им	Я могу влиять на компанию. ВУ, ХП. Стандарты кодирования. Процедуры и РИ	Корпоративная культура
Потребность в уважении (почитании): стремление к <i>компетентности, достижению успехов, одобрению, признанию, авторитету</i>	Мои знания востребованы. Наставничество приветствуется. Меня уважают и выделяют: успехи отмечены, премии за «подвиг», за «пользу»	Занимаемая роль в компании. Квалификационная сетка
Познавательные потребности: стремление <i>много узнать, уметь, понимать, исследовать</i>	Меня учат нужному мне. Созданы все условия для того, чтобы я получал знания, нужные для эффективного выполнения своей работы	База знаний. Клубы по передаче знаний (Sharing Meeting). Презентации от более опытных коллег
Эстетические потребности: стремление к <i>гармонии, симметрии, порядку, красоте</i>	В компании у нас все хорошо организовано. Обдуманые действия поощряются, и хорошо	Процедуры, регламенты и инструменты, порядок
Потребность в самоактуализации: стремление к <i>реализации своих способностей, развитию собственной личности</i>	Я должен изучать все нормы и положения, которые существуют в компании, и формировать свое отношение к рабочим ситуациям, разбираясь в нормах и поговорив с окружением о том, что происходит в реальности	Направлять мотивацию ИД посредством воздействия на «Эго» и «Суперэго»

Применение этого паттерна вполне очевидно: вы постоянно занимаетесь воспитанием норм и морали и влияете на сознание подчиненных в тех вопросах, где это уместно, приводя аргументы из данной таблицы. Графа «Воспитание норм и морали» — то, что вы обязаны довести до сотрудников формально. Графа «Влияние на аналитическое сознание» — то, что вы должны доводить до подчиненных неформально, не в виде готовых выводов, а действуя как учитель и наставник, на примерах, в разборе ситуаций, так, чтобы эти формулировки возникали уже как собственные мысли в головах подчиненных.

Самое главное — вы:

- ◆ должны быть убедительным;
- ◆ не должны говорить того, во что сами не верите;
- ◆ должны быть личным примером;
- ◆ не должны делать разницы между сотрудниками разных рангов — правила едины для всех.

### 6.4.4. Паттерн «Лидерство»

Этот паттерн, строго говоря, не является паттерном в чистом виде, скорее, это сборник некоторых хороших практик с описанием определенного подхода к формированию зрелости группы.

Основной принцип формирования зрелости группы: по мере достижения подчиненными более высоких уровней поведение лидера должно характеризоваться не только **снижением контроля над группой**, но и **уменьшением воздействий лидера на систему межличностных отношений**.

То есть вы, как лидер, должны всячески поощрять получение подчиненными более высоких результатов и явно (но неформально, как наставник и учитель) доводить до них то, что в производственных отношениях действует причинно-следственная связь «я даю результат — меня меньше контролируют — мне доверяют — у меня больше степеней свободы — со временем у меня появляются преференции, ответственность и полномочия для достижения результатов в масштабах этой ответственности».



Часто выделяют четыре стиля лидерства [14] (табл. 6.11):

- ♦ директивный (сильная ориентация на задачу и слабая ориентация на людей). Лидер определяет роли подчиненных и указывает, что, как, когда и где делать;
- ♦ поддерживающий (сильная ориентация на задачу и людей). Лидер реализует как директивное, так и поддерживающее поведение (стиль, близкий к РМ-лидерству Дж. Мисуми);
- ♦ участвующий (слабая ориентация на задачу и сильная ориентация на людей). Лидер и подчиненные совместно участвуют в принятии решений, при этом лидер выполняет в основном роли модератора и координатора;
- ♦ делегирующий (слабая ориентация на задачу и людей). Лидер малоактивен, осуществляя минимальное руководство и поддержку подчиненных.

Таблица 6.11. Стиль лидерства

Ориентация на людей	Ориентация на задачу	
	Слабая	Сильная
Слабая	<i>Делегирующий.</i> Минимальное вмешательство	<i>Директивный.</i> Лидер определяет роли подчиненных и указывает, что, как, когда и где делать
Сильная	<i>Участвующий.</i> Преимущественно роли модератора и координатора	<i>Поддерживающий.</i> Роли и правила игры определены, прямых указаний нет, есть мониторинг, контроль, курирование и наставничество со стороны лидера

В среде отдела или проектной команды можно выделить четыре уровня зрелости группы (табл. 6.12).

**Таблица 6.12.** Уровень зрелости группы

Уровень компетентности	Уровень мотивации	
	Низкий	Высокий
Высокий	Члены организации (группы) способны, но не хотят выполнять работу, предложенную (порученную) им лидером. Чаще всего это «усталые ветераны» — высококвалифицированные профессионалы, которым их область деятельности становится неинтересной	Члены организации (группы) и способны, и хотят выполнять порученную работу. Это элитная часть команды проекта, то ядро, тот локомотив, который ведет за собой остальных. Руководитель, основная часть команды которого сосредоточена в этом квадранте, молодец и счастливчик. Причем в первую очередь молодец — подобное редко происходит случайно
Низкий	Члены организации (группы) не способны и не хотят нести ответственность за свою работу. У них отсутствуют и профессионализм, и уверенность в своих силах. Люди по-разному попадают в этот квадрант, это могут быть: <ul style="list-style-type: none"> <li>• деградировавшие как специалисты ветераны, которые не смогли найти для себя нового интересного направления;</li> <li>• разочаровавшиеся новички, которые не выросли как профессионалы;</li> <li>• просто случайные в отрасли люди, которые польстились на высокие зарплаты, не будучи реально заинтересованными в профессии</li> </ul>	Члены организации (группы) не способны, но хотят выполнить порученную работу. У них есть мотивация, но отсутствуют нужные профессиональные навыки. Чаще всего ярким примером этой группы являются горящие энтузиазмом новички — студенты, молодые специалисты и т. п.

В применении к этим уровням зрелости оптимальными являются некоторые хорошо зарекомендовавшие себя подходы (табл. 6.13).

**Таблица 6.13.** Хорошие практики управления группами в зависимости от уровня их зрелости

Уровень зрелости группы	Что делать	Ближайший стиль управления	На задачу	На людей
1	2	3	4	5
1 (отсутствуют и профессионализм, и мотивация)	В рамках оперативного управления данная категория подчиненных требует ясных и конкретных указаний. Стратегически надо стремиться переместить их на уровень 2 — есть мотивация, но нет	Директивный	+	- Развивать мотивацию и вовлеченность

1	2	3	4	5
	<p>умения работать. Парадоксальным, но эффективным способом повышения мотивации таких сотрудников может стать их обучение — в процессе его у сотрудника могут проснуться интерес и желание работать. Дэвид Шварц в книге «Искусство мыслить масштабно» [15] дает именно такой совет: «Если вам что-либо неинтересно, постарайтесь узнать об этом больше»</p>			
2 (есть мотивация, но отсутствуют профессиональные навыки)	<p>Необходимо демонстрировать высокую ориентацию и на задачу, и на людей. Первая будет компенсировать отсутствие у подчиненных необходимых знаний и навыков, вторая — ликвидировать сопротивление группы давлению лидера</p>	Директивно-поддерживающий	+	<p>± Ориентировать на результат, повышать профессиональную квалификацию, вводить БЗ и ХП, проводить консалтинг и тренинги для новеньких, хвалить за результат, повышать уверенность</p>
3 (способны, но не хотят)	<p>Главной проблемой в групповой деятельности становится мотивация, и ориентированное на людей поведение лидера во многом позволяет ее решить. В этой ситуации имеет смысл поискать сотруднику новые обязанности, например другой проект. Также хорошо работает смещение обязанностей в смежные области: ведущему программисту поручить руководство группой; руководителю группы разработчиков дать отойти от организационных вопросов и погрузиться в архитектуру. Также неплохо работает подход, когда «усталым ветеранам» поручают обучение новичков. Этим двум категориям есть чем поделиться друг с другом: с одной стороны — энтузиазм, с другой — знания и опыт</p>	Поддерживающе-участвующий	±	<p>+ Пришло время вводить KPI и KGI и дистанцироваться от операционных вопросов, держа их под контролем и вмешиваясь на правах модератора</p>

Таблица 6.13 (продолжение)

1	2	3	4	5
4 (способны и хотят)	Нет необходимости в активном воздействии на подчиненных, так как они уже могут работать самостоятельно, обладая как высокой мотивацией, так и высоким профессионализмом	Делегирующий	-	+ Процессное управление. Минимальное вмешательство в исполнение процессов. Максимальный акцент на KGI и KPI. Корректирование процессов в зависимости от показателей качества



Самая простая и самая действенная рекомендация: вовлечь сотрудников в рабочий процесс поможет формирование реестра проблем, в который любой сотрудник может занести информацию о том, что наболело. Дальше с этим реестром работают в соответствии с принципами, изложенными в паттернах «Открытые вопросы», «Выученные уроки» и «Хорошие практики». Механизмы ВУ и ХП сами по себе являются паттернами, вовлекающими сотрудников в процесс и формирующими командный дух.

Можно порекомендовать в качестве постоянной мотивации не скупиться на похвалу. Я всегда исповедую такой подход: «Если есть за что похвалить сотрудника, то надо сделать это как можно раньше и максимально ярко, обязательно показав ему именно его вклад в общее дело. Если есть за что отругать сотрудника, то лучше сделать это с холодной головой, не переходя на личности. Нужно предварительно выяснить, что именно помешало ему выполнить эту задачу, максимально объективно разобраться в ситуации и попросить сотрудника выполнить паттерн “Выученные уроки”, разослав обезличенный (по возможности) новый выученный урок всей команде». Это работает. Через определенное время у вас появляется самомотивированная команда, которая учится на ошибках друг друга с помощью ВУ и строит оптимальные процессы посредством ХП.

В дополнение к сказанному: я часто мотивирую сотрудников, показывая им, что это можно сделать еще лучше. Здесь главное — соблюсти баланс между похвалой за реально выполненную работу, если результат достигнут, и объяснением того, что можно было бы сделать еще, чтобы были достигнуты и определенные вспомогательные результаты. Например, мы говорим об аналитическом отчете по результату исследований и показываем, что если резюме вынести наверх, то как минимум будет сэкономлена масса времени руководства, а если следующим разделом сделать «Рекомендации», то документ приобретет законченный вид, а команда получит толчок для дальнейших шагов по решению проблем, обозначенных в разделе «Резюме». И начнет движение не с нуля. Кро-

ме того, все заметят глубину проработки вопроса конкретно этим сотрудником и будут признательны за его труд.



Ну и конечно же, говоря о мотивации, нельзя забывать о денежном поощрении. Самым простым способом мотивировать сотрудников является метод учета коэффициента трудового участия. На сайте книги, перейдя по ссылке [http://bit.ly/pmway\\_ild](http://bit.ly/pmway_ild), вы найдете табличку с шаблоном учета КТУ. Главное, что предполагает метод КТУ, — открытое обсуждение с сотрудником того, почему у него именно такой КТУ, за что он понижен или повышен. После каждого такого разговора (а точнее, одновременно с ним) должен корректироваться план персонального развития сотрудника.

Помимо этого, я рекомендую начальнику отдела вести реестры премирования (табл. 6.14) и депремирования.

**Таблица 6.14.** Реестр премирования

№	Месяц	Отдел	За что	Кого	Сколько	
Экономический эффект и обоснование	Комментарий	Инициатор	Дата инициации	Статус	Решение	Сумма

Дам небольшие пояснения по поводу реестра премирования. В графе «За что» довольно подробно описывается конкретная ситуация, за которую премируют сотрудника. Например: *«При разработке решения для функционала А при проектировании архитектуры была предусмотрена и реализована возможность вести учет по любому количеству клиентов, если есть возможность собирать информацию о них без предварительного запроса такой функциональности со стороны заказчика в постановке задачи. Буквально через три дня заказчик высказал такое желание, а оно уже было предусмотрено в результате изначально правильного подхода».*

Графа «Сколько» — лучше, если в ней будет указан какой-то процент от некоей базы. Например, 5 % от месячного оклада. При этом процент зависит от типа достижения.

Графа «Экономический эффект и обоснование» нужна вам, как руководителю, чтобы обосновать, почему бизнес должен платить деньги за это достижение.

Графа «Комментарий» — в ней указан тип достижения, от которого зависит процент вознаграждения, например: *«Отмечается правильный архитектурный подход».*



Реестр депремирования выглядит точно так же, как и реестр премирования, только одна из граф будет называться не «Экономический эффект и обоснование», а «Упущенная экономическая выгода». Важно, чтобы здесь были определены четкие и прозрачные правила депремирования. Для начала также важно, чтобы была определена некоторая база, например процент от недельной зарплаты. Далее я составил перечень возможных «залетов» и ввел весовые коэффициенты, которые показывали подчиненным: данная область крайне важна для компании, поэтому за прокол в ней депремирование больше.

Также крайне важно ввести ограничение по максимальной сумме штрафа. Например, сумма штрафа не должна превышать 30 % от месячной зарплаты.

Таблица депремирования и принципы премирования должны открыто обсуждаться сначала с руководителями рабочих групп, а затем со всеми сотрудниками. Право окончательного решения, естественно, останется за вами, но сам факт того, что мнение каждого сотрудника было выслушано и аргументированно принято или отвергнуто, развивает командный дух.

Немного другой взгляд, но дополняющий то, о чем мы говорим, дает теория Маккилленда (табл. 6.15).

**Таблица 6.15.** Базовая потребность — зона лидера — зона команды

Базовая потребность	Описание потребности	Зона лидера	Зона команды
Мотивация достижений	Потребность быть всегда впереди, стремиться преуспевать и брать на себя ответственность за решение проблем	Ответственность за решения. Модерирование	Ответственность за решения. Выявление проблем. Стремление стать лучшими
Потребность в присоединении или принадлежности	Потребность приобретать друзей, становиться членом групп и ассоциироваться с другими	Обеспечение комфортных условий труда и качества трудовой жизни	Желание быть причастным именно к этой команде
Потребность во власти	Потребность влиять на других людей. Направлять их и, возможно, доминировать над ними. Может быть как позитивной, выливаясь в убеждение и вдохновение других при решении проблем, так и негативной, если она не связана с конструктивными решениями общих проблем	Направлять персонал. Методы убеждения и вдохновения на решение сложных задач и проблем	Желание и осознание способностей команды изменить мир к лучшему. Иметь влияние на смежные процессы, проекты и решения в них

Под качеством трудовой жизни сотрудники чаще всего подразумевают (в порядке частоты ответов):

- ◆ интересную работу;
- ◆ справедливое вознаграждение и достойную оплату труда;
- ◆ обустроенное рабочее место;
- ◆ минимальный контроль со стороны администрации;
- ◆ участие в управлении;
- ◆ гарантию работы, дружеские отношения в команде;
- ◆ средства бытового обслуживания.

В качестве рычагов воздействия в ИД (см. паттерн «Маслоу по Фрейду») можно порекомендовать развивать следующие аналитические составляющие в «Эго» сотрудников:

- ◆ желание работать;
- ◆ удовлетворенность зарплатой;
- ◆ удовлетворенность трудом;
- ◆ психологический климат;
- ◆ совместную деятельность;
- ◆ проведение совещаний;
- ◆ общение и взаимодействие.

### 6.4.5. Паттерн «Саморазвитие и личностный бренд»

В этом разделе я хочу привести пирамиду развития навыков менеджера (рис. 6.7).

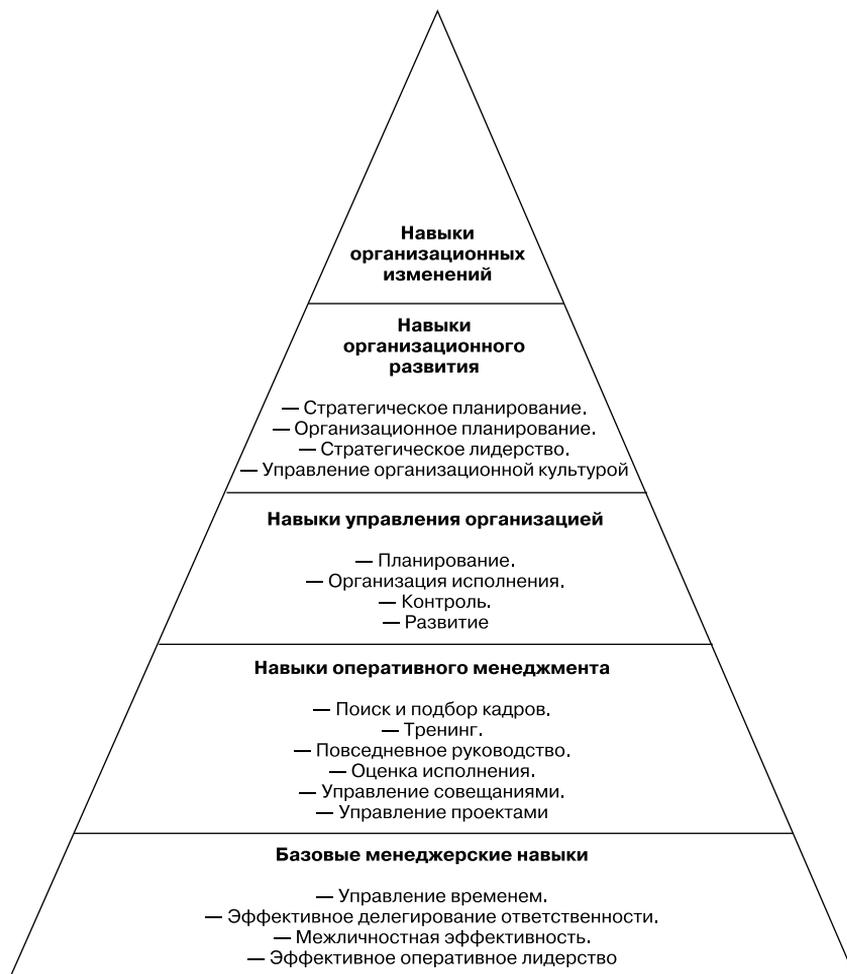
Как начальник отдела, вы должны обладать всеми базовыми менеджерскими навыками, навыками оперативного менеджмента и управления организацией и уметь применять их на практике. В меньшей степени на этой позиции от вас ожидают умения осуществлять стратегическое и организационное планирование, но стратегическое лидерство и управление организационной культурой должны входить в ваши компетенции. Пирамида развития навыков менеджера поможет вам понять, какие навыки и компетенции вам нужны на каждом этапе вашей управленческой карьеры и при планировании саморазвития.

Кроме компетенций и лидерских качеств, я рекомендую вам формировать личностный бренд. Личностный бренд начинается с завоевания авторитета в компании, в которой вы работаете. Это авторитет у сотрудников не только вашего отдела, но и у сотрудников отделов, с которыми вы взаимодействуете в производственной деятельности, и у прочих коллег.

В табл. 6.16 представлена модель четырехмерного личностного бренда.

**Таблица 6.16.** Модель четырехмерного личностного бренда

Социальное измерение	Функциональное измерение	Ментальное измерение	Духовное измерение
Моя способность взаимодействовать с другими людьми	Моя способность приносить пользу другим	Моя способность помогать другим измениться (но прежде изменить самого себя)	Моя способность духовно соединяться с жизнью во всей полноте и брать на себя локальную или глобальную ответственность за экологию, мораль, решение проблем и т. п.



**Рис. 6.7.** Пирамида навыков менеджера

В качестве примера приведу модель Коко Шанель (табл. 6.17).

**Таблица 6.17.** Модель четырехмерного личного бренда Коко Шанель

Социальное измерение	Функциональное измерение	Ментальное измерение	Духовное измерение
Она поддерживала обширные социальные связи с артистами, политиками и бизнесменами	Она была ведущим дизайнером, привнесла в моду маленькое черное платье и костюмы для женщин, а также дешевую бижутерию, чтобы многие женщины могли носить украшения	Ее идеи и влияние до сих пор живы; она была необычной, никогда не выходила замуж, имела любовников	Она помогала женщинам получить свободу, сделать их жизнь проще, менять ценности и добивалась того, чтобы женщины стали равными мужчинам

Хороший прием для составления личного бренда — представить себе, какие хорошие слова скажут о вас на ваших похоронах по этим четырем измерениям. Какой след вы хотели бы оставить в жизни?

## 6.4.6. Паттерн «Самоорганизующиеся команды»

### Определение

Самоорганизующиеся команды в последние годы стали модным трендом в проектах разработки ПО. Без докладов на эту тему не обходится ни одна конференция. Давайте разберемся, что же это такое и в каких случаях может оказаться нужным.

Наиболее нравящееся мне определение: команда — это «...небольшая группа людей с дополняющими навыками, с общей целью, стремящаяся улучшить свою производительность и чувствующая ответственность по отношению друг к другу» [19].

Определение отличное, и требуется совсем немного, для того чтобы прибавить к нему термин «самоорганизация», означающий, что команде не требуется постоянного управляющего воздействия в виде формальных постановок задач, распределения задач и контроля их выполнения. Команда берет на себя часть управленческих функций и реализует их без формального лидера.



---

Возможно, я не прав, но, с моей точки зрения, понятие «самоорганизующаяся команда» — тавтология, например как «масло масляное». Если рабочую группу можно назвать командой, то она уже несет в себе черты самоорганизации, если признаков самоорганизации нет, то, скорее всего, речь идет просто о рабочей группе и называть эту группу словом «команда» — сильно ей льстить.

---

Но это отступление, а хочется поговорить о том, зачем может понадобиться создание такой команды.

#### ◆ Для решения тактических задач.

- ◆ Освобождается время руководителя. Самоорганизующаяся команда снимет с руководителя часть задач по оперативному управлению. Проще говоря, исчезает часть повседневной рутины менеджера проекта в стиле: «Мне нужна помощь Пети, скажи ему, чтобы помог мне», «Я не знаю, как это делать, помогите мне», «От нас пули ушли, проблемы на вашей стороне». Команда решает подобные вопросы самостоятельно, а менеджер имеет возможность сосредоточиться на более важных задачах.
- ◆ Одна голова хорошо, а две — лучше. Даже самый лучший менеджер не гениален, и участие команды в принятии решений позволяет рассмотреть ситуацию с разных сторон и сделать лучший выбор.

- ◆ Решение → ответственность → качество. Участие в принятии решения повышает ответственность членов команды за результат. Это уже не директива, спущенная сверху, это ИХ решение. От него уже не отмахнешься со словами: «Мое дело — не думать, мое дело — делать, как велено». Если получится плохо, это не ошибка «тупого менеджера», это ИХ ошибка. По своему опыту знаю, что команда, привлекаемая к проработке требований к дизайну системы, начинает выдвигать более высокие внутренние требования к качеству, чем заказчики. Изобретаются самые изощренные способы тестирования, в архитектуру закладываются решения для повышения надежности системы.
- ◆ Для решения стратегических задач.
  - ◆ Повышение мотивации. Один из основных аспектов — возможность участвовать в решении. Людям нравится работать на проектах, где они имеют право голоса. Возможность высказывать свое мнение, вносить вклад в создание продукта увеличивает мотивацию сотрудников, которая не может быть заменена деньгами. *Правда, деньги также не могут быть заменены данным способом мотивации, не забывайте об этом.*
  - ◆ Снижение текучки кадров. Здесь все просто: из-за повышенной мотивации люди не торопятся уходить из коллектива. Это ИХ коллектив, ИХ продукт, а СВОЕ так просто не бросают. Из таких коллективов уже не уходят через 1–2 года работы, чтобы просто сменить место работы, получать зарплату больше на 10 % и т. д.

## Как создаются самоорганизующиеся команды

Во-первых, надо понять, что такие коллективы редко создаются сами по себе, «самоорганизующаяся» не означает «образовавшаяся самостоятельно». Во-вторых, формирование таких коллективов нельзя поставить на поток, каждая подобная команда уникальна, как уникален и процесс ее образования.

Тимоти Листер и Том Де Марко в замечательной книге *Peopleware* [13] приводят сравнение с работой садовника: можно взять замечательные семена, посадить их в плодородную почву, выпалывать сорняки, удобрять и поливать землю, но ничего не взойдет. К сожалению, это так, но многое может сделать и руководитель — по крайней мере вырывать сорняки и удобрять почву, то есть создать подходящие условия и устранить помехи.

### Подходящие условия

Необходимый минимум, который вы должны обеспечить для создания эффективной команды, — поставить ясные цели, определить ограничения, дать свободу в принятии решений по достижению целей.

Рассмотрим перечисленное подробнее.

**Ясные цели.** Вам нужно постоянно работать над тем, чтобы весь коллектив понимал краткосрочные и долгосрочные цели проекта. Например, должно существовать четкое понимание того, какого результата ожидают от команды

в конце итерации, а какого — от проекта в целом. И определение целей означает не просто выдачу команде директив в стиле «Нам нужно выпустить продукт к *N*-му году, а в течение ближайших трех недель реализовать функционал управления правами пользователей». Для того чтобы команда понимала, в чем заключаются цели, от них требуется понимание предметной области, в которой они работают, и определенное понимание рынка, для которого создается продукт. Менеджер должен либо сам донести до команды всю эту информацию, либо привлечь людей, которые могут сделать это. Еще одним аспектом работы с целями является то, что менеджер должен работать с личными целями своих сотрудников и находить корреляции между ними и целями организации/проекта/отдела. Например, в беседе с программистом выясняется, что он планирует в перспективе вырасти до архитектора. Отлично! Если его навыки позволяют выполнять дизайн отдельных элементов системы, начните привлекать его к работе архитектора. Он еще недостаточно опытен для данной работы? Определите, что ему нужно сделать для того, чтобы повысить свой уровень. Возможно, кому-то хочется попробовать новые для него технологии, кому-то — поучаствовать в написании требований вместо программирования. Работа с подобными ожиданиями сотрудников, разъяснение им целей проекта, выяснение их целей, поиск способов мотивации, отличных от зарплаты, — задачи именно менеджера.

**Дайте свободу в выборе способа достижения целей.** После того как цели стали ясны, предоставьте людям определенную свободу в их достижении. Начинать лучше всего, конечно, с небольших целей. Например, предложите команде самостоятельно запланировать короткую итерацию или даже меньше — разбить фичу на технические задачи. В качестве первого шага можно дать членам команды самим определить, на каком уровне они готовы принимать решения. Практика показывает, что разработчики вполне самокритичны и не рвутся с первых дней трудиться над стратегическими целями. Лично мне больше всего нравится начинать именно с архитектуры и технических задач. Как правило, разработчики готовы к этому, эти задачи им нравятся и результат выходит отличный при минимальном вмешательстве менеджера (как правило, для модерирования встреч или уточнения целей).

Я сейчас скажу очевидное: дав людям свободу, дайте ее на самом деле. Это значит, дайте им возможность совершать ошибки, даже если вы видите, что это происходит. Очень часто, предоставив свободу на словах, на деле руководитель продолжает постоянно поправлять своих сотрудников, не давая им возможности приобрести опыт. Так ничего не выйдет. Нужно бить себя по рукам, прикусывать язык, иначе они никогда не приобретут собственный опыт и не научатся нести ответственность за СВОИ решения.

**Определить ограничения и правила.** Задайте ограничения и правила, которыми должна руководствоваться команда при принятии решений. Чаще всего такими ограничениями становятся время и деньги. Например: «Предлагаете переделать архитектуру? О'кей! А в сроки сдачи релиза мы вписываемся?» Очень удачно в этом плане можно использовать практики SCRUM.

- ◆ Есть итерация фиксированной длины (ограничение).
- ◆ Есть сессия по планированию в начале каждой итерации (правило).
- ◆ Во встрече по планированию участвуют все члены команды (правило).

Дополнительными положительными приемами формирования команды могут служить следующие.

- ◆ Ограничение размеров команды — все сказанное ранее касается относительно небольших коллективов — 5–9 человек, самоорганизующаяся команда — небольшая спаянная группа единомышленников, а не толпа. Увеличение числа членов команды приведет к тому, что усложнятся коммуникации между ними.
- ◆ Формирование ощущения индивидуальности и отличности от других команд. Здесь хорошо работают такие вещи, как самоназвание команды, майки, кружки и прочее с названием и символикой команды.
- ◆ Возможность профессионального обучения — коллективные тренинги, посещение конференций, семинаров.
- ◆ Собственный рабочий процесс — например, новые практики наподобие автоматизированного тестирования или непрерывной интеграции, внедрение SCRUM. Отказ от каких-либо бюрократических процедур, принятых в компании. На худой конец хотя бы переход на новую версию используемого фреймворка.
- ◆ Неформальные встречи вне работы — походы на концерты, выставки, в тот же пивной ресторан, чего скрывать, шашлыки по выходным. Неформальное общение позволит людям лучше узнать друг друга и повысить степень взаимного доверия.

Правда, при использовании дополнительных мер легко увлечься и перейти на «темную сторону силы».

- ◆ Может начаться огульное внедрение и применение новых практик, невзирая на их эффективность и целесообразность использования в конкретном случае. Примером таких практик могут служить «покерное планирование», выполняемое двумя программистами, или стремление достичь 100%-ного покрытия кода модульными тестами.
- ◆ Неформальные встречи вне работы могут превратить хороший рабочий коллектив в компанию «друганов». В таких коллективах много смеются, отлично проводят время, только проект не идет. По терминологии М. Литвака происходит переход от команды «карьерного» типа к команде «культурно-развлекательного» или «алкогольного» типа (см.: Литвак М. «Командовать или подчиняться»).

### Помехи

Если гарантированного рецепта создания эффективных команд нет, есть только рекомендации, которые могут помочь, то действий с гарантированным противоположным эффектом гораздо больше.

Итак, что может свести на нет возможность самоорганизации.

- ◆ **Физическое разделение команды.** Если у вас люди разбросаны по разным кабинетам по всему зданию, сделать из них спаянный коллектив будет тяжело. У меня был опыт: четыре человека из команды сидели в одной комнате за соседними столами, еще трое были разбросаны по другим комнатам. Через три месяца четверка понимала друг друга с полуслова и представляла собой отличный коллектив, а бывшая на отшибе тройка сильно отставала по уровню вовлеченности в проект и уровню понимания создаваемого продукта.
- ◆ **Дробление рабочего времени между проектами.** Никогда не допускайте этого. Если есть хоть малейшая возможность избежать дробления рабочего времени, всеми силами старайтесь реализовать ее. Во-первых, распыление сил не дает людям прикипеть душой к проекту — у него их несколько. Во-вторых, по результатам исследований, на переключение между проектами тратится неоправданно много времени — если сотрудник задействован на трех проектах, то его эффективность снижается примерно на 60%! В-третьих, недобросовестным сотрудникам это дает возможность отлынивать от работы, особенно если проекты ведут несколько менеджеров: «Жене сказал, что пошел к любовнице, любовнице — что пошел к жене, а сам...»
- ◆ **Снижение требований к качеству.** Людям нравится гордиться тем, что они создают. Если вы будете «продавливать» и выпускать релизы низкого качества, то никогда не сможете добиться у людей горящих глаз. Если на проекте ходят шутки на тему «продукт — отстой» или выражают сочувствие пользователям продуктов, это очень тревожный симптом. Такую ситуацию проще предотвратить, чем выправить.
- ◆ **Бюрократия** — один из наиболее демотивирующих факторов. Хотите снизить мотивацию программистов — заставьте их ежедневно писать отчеты о проделанной работе или в течение двух месяцев возиться с бумагами для закупки ПО на сумму порядка 1000 рублей. Проверено на практике — работает безотказно!
- ◆ **Неудачные люди.** Бывает, неплохие специалисты совершенно не приспособлены для командной работы. Это могут быть люди со сложным характером, с низкой мотивацией, работающие из-под палки. Стоит обратить внимание на описание паттерна «Лидерство», там приведена матрица 2 × 2 (компетентность — мотивация). Участники эффективной команды расположены в правой половине матрицы — это либо горящие энтузиазмом новички, либо высокомотивированные профессионалы. Будьте осторожны, добавляя в команду человека с низкой мотивацией (левая половина матрицы) или просто сотрудника со сложным характером — один такой человек будет огромной помехой в создании эффективной команды, а будучи введенным в уже сформировавшуюся команду, способен ее разрушить.

## Когда и кому это нужно

Всегда ли нужно работать над созданием самоорганизующихся команд? Я бы сказал: «Да», если бы не одно «но». Не всегда менеджер, непосредственно работающий

с командой, определяет политику формирования команд в организации. А если мы говорим о крупных организациях, то менеджеры проектов, как правило, не определяют эту политику, она формируется на более высоком уровне.



Работать над созданием самоорганизованной команды имеет смысл тогда, когда вы планируете сотрудничать с ней довольно продолжительный период — от одного года. Если же в организации принята политика, согласно которой команда создается для краткосрочного проекта (до шести месяцев), а по его завершении распускается и не будет снова собрана вместе, целенаправленное внедрение практик самоорганизации неоправданно.

Это связано со сроками. Путь к достижению самоорганизации занимает определенное время, причем в силу того, что какое-то время будет затрачено на притирку людей друг к другу, совершение ими определенных ошибок, в первое время наблюдается даже падение эффективности. Исходя из эмпирического опыта отмечу, что производительность команды меняется следующим образом.

- ◆ До трех месяцев — производительность падает, люди учатся работать по-новому.
- ◆ От трех до шести месяцев — производительность выравнивается, но явных преимуществ вы не ощутите.
- ◆ После шести месяцев — только сейчас, если все прошло успешно, вы ощутите достоинства самоорганизации.

Поэтому, прежде чем приступить к работе над созданием самоорганизующейся команды, подумайте над следующими вопросами.

- ◆ Сколько времени продлится проект?
- ◆ Будет ли возможность сохранить команду для работы над последующими проектами?
- ◆ Подходит ли состав коллектива для командной работы, нет ли там пассивного балласта и антилидеров?
- ◆ Достаточно ли у вас полномочий, чтобы повлиять на кадровый состав коллектива?

## Вы так как-нибудь самоорганизуетесь...

Знаете, после всего, что здесь написано о пользе самоорганизации, будет выглядеть странным следующее соображение: то, что менеджер-практик много говорит о самоорганизации, кричит, что у него «команда самоуправляемая», — довольно тревожный сигнал. Скорее всего, речь идет о том, что данный менеджер просто не способен управлять людьми и на самом деле команда вертит им как хочет и делает что хочет, а он просто использует термин «самоорганизации» для самооправдания.

Это как в анекдоте про больного энурезом и психоаналитика:

- Почему у тебя унылый вид?
- А-а... стыдно признаться... Энурез — мочусь во сне.

– Сходи к психотерапевту, он тебя вылечит.

Через месяц:

– Ну, у тебя совсем другой вид, держу пари, что психотерапевт тебя вылечил от энуреза.

– Нет, не вылечил, но теперь я этим горжусь!

Так и здесь: руководитель не умеет управлять подчиненными, но у него есть оправдание — «самоорганизация», теперь он не тряпка, не умеющий руководить, а опытный управленец, использующий в работе передовые практики.

Наверное, здесь свой вклад вносит сам термин «самоорганизующаяся команда». Многие трактуют его так, что команда организовалась сама по себе. Это неверно, команду кто-то сделал такой, вырастил, и теперь она обрела самостоятельность. И умение «выращивать» такие команды, с моей точки зрения, — основной признак квалификации в области People Management.

## Заключение

Удачным завершением данного раздела является цитата из доклада «И все-таки программисты — дети!», сделанного Романом Юферевым на AgileDays'2011 в Москве: «Но помните, и вашим детям, и вашей команде нужна мать, а не нянька». В начале данного раздела я сравнивал создание команды с работой садовника, но можно провести явные параллели с воспитанием ребенка: сначала вы максимально его опекаете, потом даете чуть-чуть самостоятельности и выпускаете во взрослую жизнь, когда он повзрослеет. Кстати, с самим докладом тоже советую ознакомиться — будет полезно и весело. Видеозапись доклада выложена по ссылке <http://vimeo.com/21287114>.

### 6.4.7. Плохие паттерны

В этом разделе мы вновь коротко поговорим о плохих паттернах, которые я встречал в своей практике.

**Антипаттерн «Сами разберутся».** Начальник отдела занят чем угодно, кроме наставничества и курирования. Он считает, что сделал все, что нужно, выдав своим сотрудникам ссылки на БЗ отдела и описание процедур. В дальнейшем видит себя в роли контролера. В случае конфликтов не принимает участия в проектах. Авторитет у сотрудников утерян. Команды нет. Качество услуг, предоставляемых таким отделом, зависит от желания и профессионализма каждого конкретного сотрудника.

**Антипаттерн «Я знаю, как лучше!».** Начальник отдела вмешивается в проектную работу, требуя от своего подчиненного выполнения каких-то работ/создания артефактов и не согласовывая эту деятельность с менеджером проекта. В итоге сроки сорваны. Авторитет отдела в РМО утерян. Проект в кризисе.

**Антипаттерн «Пусть менеджеры сами планируют».** Начальник отдела не следит за работами сотрудников своего отдела. WAS не используется. Нет понимания того, какой сотрудник чем занят. Нет понимания того, когда будут готовы те или иные артефакты, когда может понадобиться кураторская поддержка начальника отдела. И проектные и операционные работы отдела непрозрачны. Часто работы выполняются аврально. Некоторые люди перегружены, другие не загружены. Команды нет. Лидера нет. Задачи «виснут», качество артефактов страдает.

Думаю, что вы без труда вспомните и другие антипаттерны. Я привел те, которые, на мой взгляд, являются самыми критическими с точки зрения управления отделом.

Мы с вами подошли к концу раздела, и, подводя итоги, в заключение я хотел бы уже традиционно задать риторический вопрос...

## 6.4.8. Так что же такое управление отделом?

На мой взгляд, лучший начальник отдела — это отец-командир своего «взвода», суровый, но справедливый, умеющий построить программы подготовки личного состава и проводить «учения», подставляющий плечо в сложных ситуациях. Сотрудники могут и «за жизнь» с ним поговорить, и прийти к нему с проблемой, будучи уверенными в том, что им помогут разобраться, после чего будет найдено правильное и изящное решение. С ним легко, за ним они в безопасности в конфликтах с руководством, а если надо на амбразуру, то он будет рядом, плечом к плечу. Иными словами, начальник отдела — это тот самый незаметный лидер, который, находясь в гармонии с тем, что имеет (люди, процессы), создает новый фундамент, обеспечивающий уровень подготовки, достаточный для достижения успеха «боевого проекта», как только его «солдатам» представится такая возможность.

И еще хочу привести два слайда из презентации о лидерстве (рис. 6.8 и 6.9).



Рис. 6.8. Формула успеха по дао

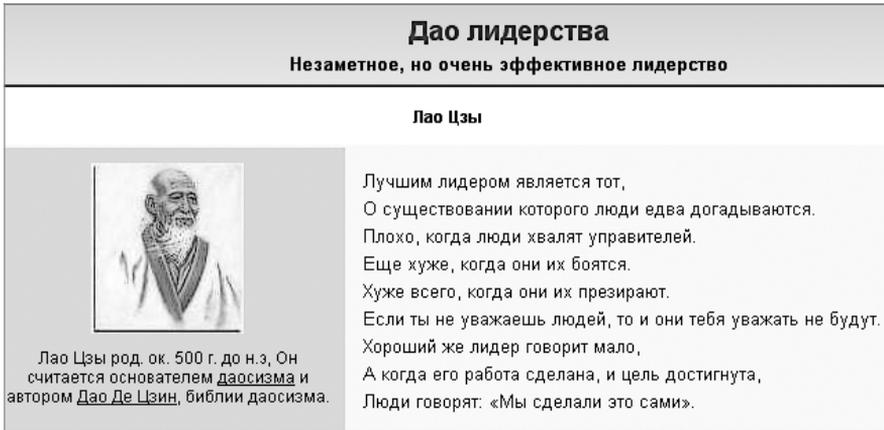


Рис. 6.9. Дао лидерства

Штирлиц знал, что запоминается последняя фраза. Поэтому закончу эту главу цитатой: «Лучший лидер — тот, о существовании которого люди почти не знают. Когда работа сделана, они говорят: мы сделали это сами» (Лао Цзы).

## 6.5. Паттерны управления подразделением разработки ПО

В этом разделе мы поговорим об управлении подразделением разработки ПО и о паттернах управления, которые могут успешно применяться для управления департаментом.



Именно от вас, как от руководителя всего направления разработки ПО, бизнес ожидает выполнения поставленных задач. От вас зависят правильная направленность и сбалансированность портфеля проектов. От вас зависят эффективность и оптимальность процессов разработки и управления, стандарты квалификации персонала и командный дух.

Здесь мы рассмотрим некоторые паттерны, которые я считаю крайне полезными для работы руководителя подразделения разработки ПО. Я практически не делаю акцента в этой книге на стратегическом управлении и балансировании по целям с топ-менеджерами и их бизнес-стратегиями, бюджетировании и управлении бюджетом, а также стратегии развития продуктов. Эти вопросы выходят за ее рамки.

Я постарался сконцентрироваться только на паттернах управления. Ими, как хорошими практиками, я хочу поделиться с руководителями подразделений разработки ПО, потому что, будучи руководителем этого звена, вы, скорее всего, ничего нового для себя здесь не найдете.

## 6.5.1. Наиболее важные паттерны

Далее перечислю уже упомянутые в этой книге паттерны, которые, на мой взгляд, крайне важны и которые обязательно должен знать и уметь применять руководитель подразделения разработки ПО.

- ◆ Паттерны исследования и формирования бизнес- (проектной) среды:
  - ◆ «Управление средой проекта»;
  - ◆ глава «Общие понятия».
- ◆ Паттерн стратегической направленности «Управление неопределенностью проекта» (применительно к стратегическому и информационному полю всей компании);
- ◆ Паттерны операционной деятельности:
  - ◆ «Фундамент отдела» (масштабированный на подразделение разработки);
  - ◆ «План управления документами»;
  - ◆ «Итеративная разработка ПО»;
  - ◆ «Обеспечение качества и анализ рисков»;
  - ◆ «Экстремальная разработка» и «Расширенный устав проекта»;
  - ◆ «СУП на РМВОКе» и «СУП на воде»;
  - ◆ «Зажарка в СУП, или Универсальные принципы управления проектами».
- ◆ Паттерны процессного совершенствования:
  - ◆ СММІ;
  - ◆ IDEAL;
  - ◆ «Система управления производством ПО (СУПр)».
- ◆ Паттерны управления персоналом/командообразующие паттерны:
  - ◆ «Информационный бюллетень»;
  - ◆ «Маслоу по Фрейду»;
  - ◆ «Лидерство»;
  - ◆ «Саморазвитие и личностный бренд».
- ◆ Паттерн «Минимальный офис управления проектами».



---

В качестве примера этого паттерна я приведу созданный мной фреймворк управления проектами в компании с уровнем зрелости между уровнем 1 и 2 по СММІ, то есть между уровнями часто уникальных и каждый раз заново и частично повторяемых процессов.

Проекты в этой компании выполнялись быстро — максимальная длительность — три месяца, были в основном ориентированы на внутреннего заказчика. Часть проектов, по сути, являлись задачами по оперативному изменению существующих продуктов, лившимися бесконечным потоком.

---



Для того чтобы как можно быстрее навести порядок и подтянуть департамент до хоть какого-то понимания принципов проектной деятельности, мной был создан вот такой вот...

## Упрощенный фреймворк управления проектом PDCA (цикл Демминга)

### **P: Plan – планируйте.**

Ищите готовое решение на стороне до того, как начать все остальное.

Ищите тех, кто делал подобное на стороне, — отдавайте на аутсорсинг.

Применяйте SMART.

В ходе коммуникаций следуйте принципу «5К» («кто, кому, в какой форме, когда, как»).

Всегда определяйте роли и ответственности ролей — к кому по какому поводу обращаться и кто за что отвечает, особенно за подготовку тестирования (сами тесты и тестовые данные) и проведение тестирования.

Всегда иницилируйте проект и оценивайте трудоемкость.

Всегда составляйте ПЗ или хотя бы его упрощенные письменные формы (например, письмо, отправляемое по почте).

### **D: Do – выполняйте планы.**

DAR – Decision Analysis and Resolution.

Главное — заранее определить критерии, по которым будет проводиться анализ альтернатив.

Каждый критерий должен заранее получить свой вес, с которым он принимает участие в суммарной оценке, то есть то, насколько именно это важно.

Также заранее следует определить, как будет проводиться анализ альтернатив (как будут трактоваться значения критериев) при их выборе.

Выполнять задуманное, если что-то мешает сделать это, заранее предупреждать ЗЛ.

Говоря «мы это сделаем», всегда указывайте, кто «мы», к какому сроку и какие существуют ограничения и допущения.

Используйте правило Next Steps всегда и во всем, особенно во всех коммуникациях.

Добивайтесь подтверждения.

Пишите письма с открытыми вопросами.

Добивайтесь формального подтверждения там, где нужно.

### **S: Check – проверяйте и управляйте.**

Управляйте рисками.

Риски отслеживайте! Задавайте себе и команде правильные вопросы.

- ◆ Что нового появилось?
- ◆ Что может мне помешать удовлетворить бизнес-потребность заказчика?
- ◆ Бизнес-потребность все еще актуальна?
- ◆ Бизнес готов получить решение за деньги, потраченные на зарплату и оборудование?
- ◆ Я укладываюсь в сроки?
- ◆ У меня есть вся информация для работы?
- ◆ Нужные мне люди доступны?
- ◆ А завтра они будут доступны?

Следите за тем, чтобы потребность, которая зафиксирована в ПЗ, соответствовала текущей бизнес-потребности.

Поймите, что надо сделать, чтобы достичь удовлетворения бизнес-потребности, что уже мешает и что еще может помешать.

Уточняйте у заказчика актуальность реализации потребности ценой таких усилий.

Выявляйте новые риски. Разрабатывайте план предупреждения и план реагирования.

Уточняйте трудоемкость после изменения веса или влияния рисков.

Уточняйте план предупреждения и план реагирования после изменения веса или влияния рисков.

Доносите все уточнения и изменения рисков до заказчика как можно раньше!

Реагируйте на возникшие риски в соответствии с планом!

### **Управляйте проектом.**

Поддерживайте в актуальном состоянии портфель проектов (каждую запись о каждом проекте):

- ◆ ведите поля «Состояние» и «Статус» проекта совместно с командой;
- ◆ вовлекайте заказчика в управление портфелем с помощью полей «Состояние», «Статус» и «Плановая дата окончания»;
- ◆ направляйте еженедельный отчет всем ЗЛ проекта.

Совместно с заказчиком определяйте контрольные точки и формальные тесты в них.

Определяйте критерии оценки решений/вариантов (критерии приемки в контрольных точках) по DAR.

Проводите старт проекта как контрольную точку (как минимум учитывайте критерий «Наличие утвержденного формально ПЗ и оценки трудоемкости»).

Определите и проходите внутренние контрольные точки разработки (например, «Разработка (включая smoke test) проведена», «Функции проверены», «Регресс выполнен» и т. д.).

Формально, но быстро проходите внутренние контрольные точки (как минимум чек-листы).

Проводите preCAT (Customer Acceptance Test) как контрольную точку.

Выполняйте подготовку к PreCAT как личную приемку отчетов и форм с точки зрения того, будет ли достигнута заявленная в ПЗ бизнес-потребность.

Проводите CAT как контрольную точку.

Проводите тестирование IT.

Проводите тестирование совместно с бизнесом.

### Управляйте разработкой.

Специфицируйте с точки зрения достаточности (далее жирным шрифтом обозначено то, что считается основным);

- ◆ ПЗ;
- ◆ ТЗ;
- ◆ шаблон **отчетов** и **экранных форм**;
- ◆ UML:
  - ◆ **концепцию системы** (диаграммы классов);
  - ◆ поведение системы (диаграммы последовательности);
- ◆ архитектурные спецификации:
  - ◆ **слои**;
  - ◆ распределенную модель;
  - ◆ **компонентную модель**;
- ◆ документацию пользователя.

**Документируйте код** и проверяйте документирование кода.

Проверяйте соблюдение стандартов кодирования, особенно наличие простейших ошибок и узких мест (ляпов):

- ◆ отсутствие try catch;
- ◆ проверка наиболее возможной ситуации находится в конце списка проверок;
- ◆ отсутствие проверки, «если ничего из указанного»;
- ◆ отсутствие валидации параметров метода в самом начале метода;
- ◆ отсутствие валидации данных до выполнения операции с данными в БД (update, insert);
- ◆ обращение в БД с запросами вида «select \* from A» в каком-либо цикле;
- ◆ неоправданное и неаргументированное время жизни при использовании кэшей;

- ◆ ошибки с правами доступа (пароли явно где-то лежат);
- ◆ константы в коде классов.

Выполняйте тестирование алгоритмов. Оно всегда производится на реальных данных от заказчика в первую очередь самим разработчиком.

- ◆ тестирование выполняется вне контекста всей задачи, проверяется только правильность работы самого алгоритма;
- ◆ данные, посчитанные вручную, должны сойтись с результатом работы программы;
- ◆ следует приготовить отчет о тестировании (в произвольной форме).

Применяйте **парное программирование**, если есть возможность.

Выполняйте **код ревью** критических участков кода.

Покрывайте **Unit-тестами**.

- ◆ критические с точки зрения архитектуры и логики системы и часто используемые участки кода — на 80 %;
- ◆ некритические участки кода — на 20 %.

**Тестируйте.**

Создавайте **тест-планы** и тестовые сценарии.

Проверяйте код на тривиальные ошибки.

Проверяйте **отчеты на точность**.

Выполняйте Functional Testing.

Выполняйте **Integrational Testing**.

Выполняйте **Regression Testing**.

Выполняйте **личную приемку** (PrePreCAT).

Проводите PreCAT (демонстрацию) и четко **обозначайте, что это демонстрация**.

Согласуйте с заказчиком дату и время демонстрации по доработкам, если необходима проверка на реальных данных, ее сроки и порядок согласуйте с заказчиком дополнительно.

При организации preCAT необходимо соблюдение следующих условий:

- ◆ наличие итогового ПЗ;
- ◆ PreCAT выполняют представители заказчика, исполнитель, руководитель группы;
- ◆ руководитель группы разработчиков проверяет код на соответствие ПЗ, в том числе и в отчетах.

Проводите приемку заказчиком (CAT).

Помните про стабилизацию — во время CAT дефектов быть не должно.

**Закрытие проекта.**

САТ проводится непосредственно заказчиком. Обязательно наличие письма от заказчика с подтверждением приемки. Только после этого проект считается сданным в промышленную эксплуатацию.

**Используйте разработанную демотивационную схему формально.**

Обязательно поощряйте отличившихся.

**Управляйте коммуникациями.**

Управляйте перепиской:

- ◆ вы всегда должны получать копии писем от членов вашей команды ЗЛ;
- ◆ следите за **правилами коммуникации** и переписки;
- ◆ внимательно формируйте списки «Кому» и «Копия».

Добивайтесь подтверждения получения информации и ее понимания именно в том контексте, который вы имели в виду:

- ◆ задавайте вопросы: «Я правильно понимаю, что?..», «Какие мои дальнейшие шаги?», «Прошу подтвердить срок...», «Есть ли замечания/уточнения?» и т. п.;
- ◆ подтверждайте для себя, что вы правильно поняли заказчика, повторяя его фразы своими словами и задавая вопрос, например: «Андрей, я правильно понял, что нужно излагать смысл сказанного заказчиком своими словами и добиваться подтверждения того, что смысл я уловил правильно?»

Действуйте через технического директора, если возникают сложности.

**A: Act – реагируйте на изменения и учитесь на ошибках.**

Change Control Management – управление изменениями:

- ◆ выявляйте и **фиксируйте все изменения формально в ПЗ** под формальное согласование с ЗЛ;
- ◆ оценивайте, **на что повлияет** требуемое изменение;
- ◆ оценивайте **трудоемкость** реализации изменения;
- ◆ согласуйте **нужность** изменения с такой трудоемкостью;
- ◆ не забывайте про тестирование после изменений (особенно регрессионное).

Проводите **ВУ** в ходе выполнения проекта.

Выполняйте **корректирующие действия как можно быстрее**:

- ◆ перед выполнением коррекции **уведомляйте ЗЛ о проблемах**;
- ◆ всегда **указывайте трудоемкость** и сроки таких работ;
- ◆ оценивайте и **исправляйте плановую дату окончания** работ в портфеле.

В ходе выполнения проекта всегда и во всем придерживайтесь практики Next Steps!

По окончании проекта обязательно выявляйте ВУ.

Пополняйте список тривиальных ошибок.

Выученными уроками проекта должно стать следующее: что было не так, что было хорошо, что плохо, какие практики работали, какие нет, чему научились, что можно взять в свой багаж на будущее. Такие ВУ должны появляться на сайте.

Руководитель РГ оценивает реальную выгоду от выполненного проекта и сообщает эти сведения заказчику и техническому директору.

## 6.5.2. Паттерн «Лидерство на уровне департамента»

Это еще один очень короткий паттерн. Просто перечислю некоторые практики, которые считаю полезными и которые я успешно применял, руководя департаментом производства ПО.

Периодически я проводил заседания клуба IT-менеджеров для сеньор-менеджеров департамента. Цели клуба — повышение теоретической подготовленности менеджеров, передача знаний, формирование команды. Важно здесь то, что на заседаниях мы никогда не «играли» в наши реальные кейсы — только в вымышленные. Люди должны приходить на позитиве, стремясь в игровой и приятной форме получить новые знания.

Заседание клуба заканчивалось... Протоколом. В свободной форме. Любой из участников клуба мог поправить или дополнить протокол. Так формировалась база знаний менеджеров IT. Эти знания не возбранялось доводить до рядовых сотрудников компании — на усмотрение старших менеджеров.

**Архитектурные совещания.** В ходе разработки ПО самые большие сложности и проблемы возникают при проектировании и создании архитектуры. Я ввел правило, согласно которому протокол любого совещания, на котором речь шла об архитектуре, отсылался членам архитектурного комитета. При необходимости комитет собирался для обсуждения принятых решений, корректировал их. Шел и обратный процесс — все протоколы архитектурных комитетов рассылались всем аналитикам и разработчикам, занятым в создании продукта. Любой сотрудник мог высказать свои замечания и выявить риски.

**Доска для всех.** В коридоре компании мы повесили доску с цветными магнитами и специальными маркерами. Разделили ее на три части. В левой части отмечались основные вехи главных проектов, текущие вызовы и выявленные проблемы. В средней части любой сотрудник мог оставить свою запись любого содержания по теме. В крайней правой (самой узкой) части любой сотрудник мог оставить любой комментарий по любому поводу. Раз в неделю администратор одного из проектов систематизировал записи с доски с помощью специальных инструментов и писем. Таким образом инициировались изменения, выявлялись риски, понимались проблемы и конфликты.



---

На рис. 6.10 вы видите пример такой доски.

---

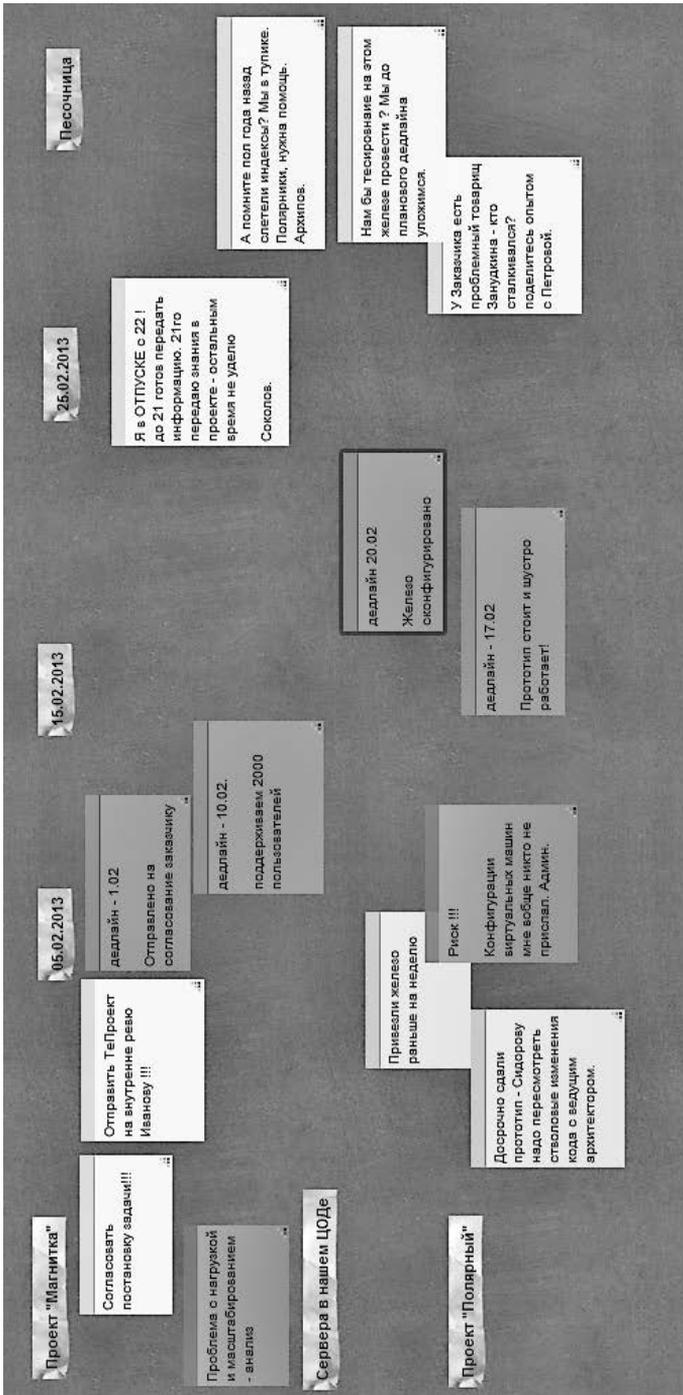


Рис. 6.10. Пример доски для всех как способа организации информационного пространства производства

«Не приходите с проблемами. Приходите с решениями» — такие слова я распечатал на листе формата А4 и повесил над своим креслом. Основной посыл был таков: вы выявили проблему и вы сеньор-менеджер. Первая ваша обязанность — оценить последствия, проработать варианты решений, их последствия, плюсы и минусы и со всем этим прийти ко мне. Изложение проблемы без предварительного анализа в моем кабинете воспринималось как «поговорить за жизнь» и практиковалось крайне редко.

### 6.5.3. Паттерн «Стратегия и планирование в СУПр»

Принцип концентрации на анализе в контрольных точках жизненного цикла продуктов звучит так: *«Основная точка отсчета — это портфель, описывающий все продукты в компании и их рынки, а также соответствующие инвестиции. Каждый продукт должен иметь список возможностей, запланированных на несколько следующих версий. В этом списке указываются назначение, рынок, архитектура и технология продукта».*



Данный принцип лежит в основе трехуровневого планирования, суть которого — в постепенном движении от стратегических целей в областях назначения, рынка, архитектуры и технологии продукта через тактическое планирование и формирование бизнес-видения, технологического видения и неделимых подмножеств коммерчески выгодных функциональных возможностей продуктов к портфелю проектов и операционному плану, включающему в себя усиление возможностей компании для достижения требуемых конкурентных преимуществ.



Компания занимается:

- ◆ развитием коробочных продуктов — 50 %;
- ◆ услугами по сопровождению и развитию адаптированных решений клиентов — 40 %;
- ◆ разработкой уникальных решений под ключ — 10 %.

## Стратегическое планирование

В соответствии со структурой операционной деятельности компании стратегические планы определяют стратегические цели по каждому из направлений.

**Развитие коробочных продуктов — 50 %.** Раз в два месяца менеджер по маркетингу продукта корректирует маркетинговую стратегию компании. Горизонт планирования маркетинговой стратегии — 18 месяцев.

Цель создания маркетинговой стратегии — определить новые ниши рынка, на которые можно выйти с существующими продуктами, или выпустить новые

продукты, а также определить нужные рынку новые направления развития уже существующих продуктов.

Маркетинговая стратегия включает в себя данные о спросе на продукт — факт и прогноз спроса, обоснование прогноза спроса, методику прогнозирования спроса, позиционирование продуктов на рынке. Позиционирование продуктов на рынке включает в себя взаимосвязанные данные о нише рынка, классах продуктов, использующихся в этих нишах, типах потребителей, целевой аудитории продуктов и специфике бизнеса потребителей.

Эту деятельность выполняет менеджер по маркетингу продукта путем выполнения маркетингового анализа потребителей, продуктов и конкурентного анализа. Все маркетинговые исследования производятся в тесном взаимодействии с менеджерами по продукту и Account Managers, которые развивают клиентов или продукты анализируемого сегмента рынка и/или анализируемого типа клиентов.

С точки зрения экономии времени, получения более качественного маркетингового исследования и принимая во внимание, что эта деятельность не относится к ключевым компетенциям компании, первичный маркетинговый анализ рекомендуется отдавать на аутсорсинг профессиональному консалтинговому агентству в области маркетинговых исследований, а в дальнейшем поддерживать силами менеджера по маркетингу продукта.

Собранная информация преобразуется с участием бизнес-аналитика в набор требуемых (желаемых) характеристик продукта (CHR) в заключительном разделе маркетинговой стратегии.

**Услуги по сопровождению и развитию адаптированных решений клиентов — 40 %.** На этапе стратегического планирования Product Group Manager для каждого продукта, за развитие которого он отвечает, создает стратегию развития продуктов. Это делается для того, чтобы определить цели и направления развития продуктов с точки зрения того, что нужно клиенту, что требует закон или изменившийся бизнес клиента.

Стратегия развития продукта включает в себя бизнес-требования, бизнес-правила и запросы на расширение/создание новой функциональности от клиентов компании.

С точки зрения потребностей клиентов или изменившегося бизнеса основа для создания стратегии развития продуктов — история общения с клиентами компании, поддержка продуктов у клиентов, анализ запросов клиентов на изменение продуктов. Эту деятельность выполняют Account Managers и Support Managers в ходе своей операционной активности. В результате получается перечень запросов на расширение функциональности (ER).

Для выявления и формирования бизнес-требований и бизнес-правил бизнес-аналитик совместно с юристом и менеджером по маркетингу продукта проводят анализ сегмента рынка и бизнесов клиентов, а также законов, регламентирующих бизнес клиентов и деятельность собственной компании. Это выполняется при подготовке стратегии развития продукта. В результате получается перечень

запросов на расширение функциональности (Business Rules (BRULE) и Business Requirements (BREQ)).

Партнеры компании могут оценивать некоторых клиентов и некоторые запросы от клиентов (ER) с максимальными весами и таким образом имеют возможность повлиять на бизнес-видение развития продукта (BVISION), что является основным результатом стратегического планирования

Деятельность по обновлению стратегии развития продуктов — это регулярная операционная деятельность отдела маркетинга и продаж, выполняется она с середины до конца каждого месяца и имеет скользящий горизонт планирования на 18 месяцев.

**Разработка уникальных решений под ключ — 10 %.** Проекты по разработке уникального решения под ключ могут быть выполнены по упрощенной схеме — для них не требуется стратегического, тактического и операционного планирования. В этом случае есть возможность сразу инициировать проект, в начальной фазе которого выявляются BR, BREQ и характеристики проекта, после чего он выполняется по одному из жизненных циклов разработки, описанных в разделе ЖЦ проектов разных типов.

Однако рекомендуемый и более правильный путь — выявлять ER всех клиентов, включая новых, и через систему весов управлять долей таких проектов в портфеле проектов. Только в этом случае СУП гарантирует формирование рентабельного, сбалансированного и стратегически обоснованного портфеля проектов.

Еще одна важнейшая часть стратегического планирования — создание архитектурной стратегии. Делается это, чтобы определить цели и направления развития продуктов с точки зрения новых технологий производства ПО, появившихся на рынке.

Архитектурную стратегию создает главный архитектор (Lead Architect) компании с участием менеджеров по продуктам и архитекторов продуктов. При создании архитектурной стратегии существующая архитектура продуктов анализируется на предмет желаемых изменений архитектуры с точки зрения ее совершенствования и/или использования новых технологий разработки ПО. В результате формируется набор уникальных требований R&D Requirements (RDREQ), которые отражают потребности в изменении архитектуры существующих продуктов и их компонентов и использовании новых технологий при создании новых продуктов и их компонентов.

Деятельность по обновлению архитектурной стратегии является регулярной операционной деятельностью отдела архитектуры и технологических исследований, выполняется с середины до конца каждого месяца и имеет скользящий горизонт планирования на 18 месяцев.

В результате стратегического планирования создан следующий набор артефактов:

- ◆ маркетинговая стратегия (скользящий горизонт планирования на 18 месяцев, выполняется раз в два месяца);

- ◆ стратегия развития продуктов (скользящий горизонт планирования на 18 месяцев, выполняется ежемесячно);
- ◆ архитектурная стратегия (скользящий горизонт планирования на 18 месяцев, выполняется ежемесячно).

Цели на первые полгода максимально детализируются, на оставшийся год цели не детализируются и остаются на высоком уровне. Данные стратегии итеративно согласуются между собой.

## Тактическое планирование

На этапе тактического планирования менеджер продукта создает/актуализирует концепцию создания и развития продукта, а архитектор продукта — концепцию развития архитектуры продукта. Эти концепции взаимно влияют друг на друга, разрабатываются параллельно и итеративно, взаимно уточняя и корректируя содержание, предложения по планированию и варианты построения тактики для достижения стратегических целей.

Концепция создания и развития продукта содержит требования типа «бизнес-видение» (BVISION), сформулированные в результате анализа запросов на расширение функциональности от клиентов, изменившихся требований их бизнеса и бизнес-правил, а также на основании результатов маркетингового анализа (выявленных желаемых характеристик системы). Концепция также содержит выводы и предложения о том, в какой последовательности, в какое время и какие BVISION развивать, какие TVISION потребуются развивать одновременно и к чему это приведет с точки зрения выполнения маркетинговой стратегии и стратегии развития продуктов. Иначе говоря, концепция создания и развития продуктов через набор уникальных и не повторяющихся BVISION предлагает варианты выполнения стратегических целей в маркетинговой стратегии и стратегии развития продуктов.



---

Концепция развития архитектуры компонентов содержит TVISION, сформулированные в результате анализа технологических трендов и обработанных запросов на создание и развитие продуктов. Концепция также содержит предложения: как и когда реализовывать TVISION, какие BVISION связаны с реализацией этих TVISION и к чему приведет реализация TVISION с точки зрения выполнения стратегических целей в архитектурной стратегии, то есть с точки зрения оптимального развития архитектуры всех компонентов, из которых состоят продукты.

---



---

В ходе создания концепций с целью поиска баланса между потребностями в развитии продуктов и технологическими направлениями развития BVISION и TVISION анализируются, взвешиваются, приоритизируются, обсуждаются на комитетах по продуктам. В ходе анализа и создания концепций формируется набор управленческих сущностей идей (FEAT), каждая из которых представляет собой набор подмножеств BVISION и TVISION, связанных

между собой с точки зрения функциональности и архитектуры компонентов и продуктов и являются неделимыми с точки зрения полезности для бизнеса. Иными словами, Feature — сущность, имеющая коммерческую ценность для бизнеса, позволяющая установить взаимосвязь и баланс между требованиями и ожиданиями бизнеса (BVISION) и потребностями в развитии архитектуры (TVISION) и являющаяся основным элементом для определения границ проектов при анализе их рентабельности и рентабельности всего портфеля проектов на этапе операционного планирования.

Набор идей фиксируется в концепции создания и развития продукта в виде отдельного раздела документа.

Горизонт планирования концепций скользящий, на 12 месяцев, концепции пересматриваются и при необходимости обновляются каждый раз при изменении любой из стратегий.

В результате тактического планирования создан следующий набор артефактов:

- ◆ концепция создания и развития продукта (с набором идей для реализации);
- ◆ концепция развития архитектуры продукта.

Тактические мероприятия на первые полгода максимально детализируются, на вторые полгода не детализируются и остаются на высоком уровне.

## **Операционное планирование (планирование производства и мощностей)**

Операционная стратегия служит для согласования спроса и возможностей компании на стратегическом уровне. Со стороны спроса выступают факторы конкурентоспособности компании: качество, скорость, уровень обслуживания, гибкость, стоимость. Со стороны возможностей выступают мощность, сеть поставок, процессы и технология, развитие организации.

Со стороны спроса операционная стратегия базируется на созданных на этапе тактического планирования концепциях создания и развития продуктов и развития архитектуры продуктов и набора идей как результата тактического планирования.

Со стороны возможностей операционная стратегия описывает потребности для реализации идей в ресурсах компании, мощностях, технологиях и методологиях, включает в себя план по привлечению и развитию нужных навыков персонала и план инвестиций в материально-техническую базу.

Для выявления указанных потребностей проводится анализ с использованием инструмента «операционная стратегия СУП», в котором в строках в качестве факторов конкурентоспособности отражаются сгруппированные по основным конкурентным преимуществам компании идеи: качество, скорость, гибкость, издержки, уровень обслуживания клиентов, для которых проводится анализ по возможностям компании, технологиям разработки, развитию и организации и мощностей, отражаемым в столбцах. На пересечении строк и столбцов

указываются конкретные мероприятия, направленные на усиление возможностей компании по обеспечению требуемого развития ее конкурентных преимуществ.



На основании набора идей формируется портфель проектов и считается рентабельность проектов и рентабельность портфеля. При формировании портфеля проектов весь набор идей разбивается на набор связанных идей, которые могут быть выпущены и проданы в качестве отдельной (промежуточной) версии «коробки» или поставлены и проданы как отдельная (промежуточная) версия решения для клиента. Каждый такой набор идей определяет границы одного проекта в рамках портфеля проектов. Для каждого проекта определяются цели, бизнес-обоснования (BVISION), оценивается трудоемкость и высчитывается показатель рентабельности на основании прогноза продаж. На базе рассчитанных показателей рентабельности проектов определяется интегральный показатель рентабельности портфеля проектов. Если показатель рентабельности портфеля/проекта не находится в границах, заданных в бизнес-стратегии, портфель проектов переформируется.

В результате этой деятельности некоторые проекты могут быть урезаны, некоторые расширены, для других проектов может потребоваться значительное усиление возможностей компании. Мероприятия по усилению возможностей также должны быть приняты во внимание, исходя из желаемых сроков окончания проекта, что может привести к изменению границ проекта. В ходе операционного планирования состав идей может измениться, некоторые BVISION и TVISION могут быть пересмотрены и даже отменены. Все эти изменения производятся только с разрешения комитета по продуктам.

В результате итеративного операционного планирования создаются:

- ◆ операционная стратегия (включая план по привлечению и развитию нужных навыков персонала и план инвестиций в материально-техническую базу);
- ◆ карточки проектов (с перечнем идей, обозначенными целями, бизнес-потребностями в выполнении проекта, техническим видением, влияющим на выполнение проекта, оценкой трудоемкости и рентабельности проекта);
- ◆ портфель проектов, состоящий из перечня карточек проектов и интегральной оценки рентабельности портфеля проектов;
- ◆ операционный план-расписание компании (первичная версия плана-расписания на уровне планов-проектов портфеля проектов, привязанных к идеям, со сроками, полученными в результате первичной оценки трудоемкости, порядка величин +75 %, -50 %).

Горизонт операционного планирования скользящий, на 12 месяцев, план пересматривается и при необходимости обновляется каждый раз при изменении любой из концепций тактического уровня.

Мероприятия операционного плана на первые полгода максимально детализируются, на вторые полгода не детализируются и остаются на высоком уровне.

Высокоуровневый план-расписание уточняется по принципу набегающей волны при следующем выполнении операционного планирования.

Для управления всеми видами ожиданий, характеристик и запросов заинтересованных лиц используются инструменты и процедуры СУТ, что ускоряет работы и позволяет легко отслеживать влияние изменений на все планы на следующей итерации планирования.

Посредством выполнения перечисленных мероприятий реализуется принцип оценки требования с разных точек зрения жизненного цикла продукта: «Анализ влияний базируется на требованиях, а также на определении приоритетов и управлении портфелем — динамическом процессе принятия решения, направленном на создание правильного сочетания продуктов и выбор правильных проектов для реализации данной стратегии», а также принцип взаимосвязи состава портфеля и отдельных продуктов жизненного цикла продукта: «Необходимо разработать план проекта, который непосредственно привязан к требованиям. Если план или состав продукта меняются, вся группа должна одобрить изменения и синхронизировать их с остальной частью проекта. Требования, а также данные о других, связанных с текущим продуктом решениях и информация о проекте должны быть доступны интерактивно».

Созданный портфель проектов балансирует бизнес-потребности и технологические потребности развития продуктов.

## 6.5.4. Паттерн «Управление портфелем проекта»



Данный паттерн является продолжением стратегического планирования для той же компании. Компания занимается:

- ◆ развитием коробочных продуктов — 50 %;
- ◆ услугами по сопровождению и развитию адаптированных решений клиентов — 40 %;
- ◆ разработкой уникальных решений под ключ — 10 %.

### Состав портфеля проектов



В компании в СУП выделяем четыре портфеля проектов: портфель коробочных проектов, портфель адаптивных решений, портфель клиентских проектов и портфель проектов внутренней автоматизации и совершенствования процессов.

- ◆ Портфель коробочных проектов — портфель, состоящий из проектов по созданию новых версий «коробок».
- ◆ Портфель адаптивных решений — портфель, состоящий из проектов по созданию уникального решения из функциональности «коробок» путем ее адаптации и доработки.

- ♦ Портфель клиентских решений — портфель, состоящий из проектов по созданию проектов под ключ.
- ♦ Портфель проектной внутренней автоматизации и совершенствования процессов включает в себя все проекты по созданию и развитию активов производства предприятия.

Бизнес-стратегия компании определяет принципы финансового управления для каждого портфеля проектов (табл. 6.18).

**Таблица 6.18.** Принципы финансового управления в зависимости от типа портфеля проектов

Тип портфеля	Принципы финансового управления
Портфель коробочных проектов	В первую очередь запускаются проекты с максимальным IRR, если комитетом по продуктам не принимается решение о запуске перспективных или имиджевых проектов с более низким IRR
Портфель адаптивных решений	В первую очередь запускаются проекты с максимальной рентабельностью. Порог рентабельности портфеля — 15 %. Проекты с рентабельностью ниже 15 % запускаются только по особому решению комитета по продуктам
Портфель клиентских решений	В первую очередь запускаются проекты с максимальной рентабельностью. Порог рентабельности портфеля — 25 %. Проекты с рентабельностью ниже 25 % запускаются только по особому решению комитета по продуктам
Портфель проектов внутренней автоматизации и совершенствования процессов	В первую очередь запускаются проекты с минимальной совокупной стоимостью владения (TVC), если управляющим комитетом не принимается решение о более раннем запуске проектов с более высокой TVC

## Управление портфелем проектов

Управление любым портфелем проектов происходит в соответствии с методологией управления портфелем проектов. Это постоянная деятельность, которая начинается на этапе операционного планирования и состоит в окончательной идентификации идей, их оценке и организации в проекты, приоритизации проектов, выборе и оптимизации состава проектов и последовательности их выполнения с точки зрения рентабельности и потребностях в бизнес- и технологическом развитии продуктов.

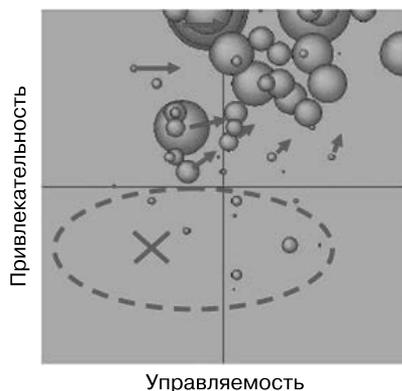


В дальнейшем, после авторизации портфеля проектов, по мере выполнения проектов из портфеля производится постоянный мониторинг и контроль общего состояния портфеля проектов, его рентабельности, целостности и актуальности с точки зрения выполнения концепций развития продуктов и развития архитектуры продуктов. В результате мониторинга проектов определяется необходимость в оценке идей и пересмотре состава проектов из портфеля проектов. В случае возникновения такой необходимости ССВ проекта инициирует анализ портфеля проектов, после чего ССВ продукта

принимает решение об актуальности ранее выявленных идей или корректирует их, утверждая на комитете по продуктам, отвечающем за портфель проектов и согласованное движение по комплексному развитию всех продуктов компании. Таким образом обеспечивается возможность трансформации того или иного проекта для оптимального и максимально возможного приближения совокупной проектной деятельности к целям бизнеса.

С технической точки зрения каждая карточка проектов содержит характеристики для оценки привлекательности и управляемости проекта для целей бизнеса (табл. 6.19). Эти характеристики регулярно пересматриваются и оцениваются ССВ проекта и ССВ продукта, в результате чего изменяется общая картина привлекательности-управляемости портфеля проектов (рис. 6.11) и по проектам, не находящимся в верхнем правом квадранте, с санкции комитета по продуктам принимаются корректирующие управленческие решения.

#### Квадрант «Привлекательность — управляемость»



**Рис. 6.11.** Портрет портфеля проектов с точки зрения привлекательности и управляемости

**Таблица 6.19.** Характеристики для оценки привлекательности и управляемости проекта

Характеристика	Краткое описание
Важность	Средний уровень приоритетов идей, входящих в границы проекта
Спонсорство	Степень возможностей спонсора проекта и степень его влияния на проект
Окупаемость	Степень возврата инвестиций или получения технологических и/или конкурентных преимуществ
Уровень рисков	Оценка влияния рисков на проект (сумма произведений вероятности возникновения каждого риска на степень влияния этого риска на проект)
Уровень реализации	Степень выполнения запланированных мероприятий проекта
Срочность	Степень безотлагательности и возможность приостановки проекта
Оценка затрат	Реалистичный бюджет проекта
Инновационность	Степень новизны используемых в проекте решений и ожидаемого результата для компании

Вычисление значений оценок привлекательности и управляемости каждого проекта на основании оценок характеристик проекта производится с учетом корректирующих коэффициентов (табл. 6.20).

**Таблица 6.20.** Весовые коэффициенты для расчета привлекательности и управляемости портфеля коробочных проектов

	<b>Важность</b>	<b>Спонсорство</b>	<b>Окупаемость</b>
<b>Привлекательность</b>	0,60	0,15	0,25
	<b>Риски</b>	<b>Реализация</b>	<b>Срочность</b>
<b>Управляемость</b>	0,40	0,15	0,45

Корректирующие весовые коэффициенты определяются РМО для каждого портфеля проектов, утверждаются Steering Committee и отражают стратегию управления портфелем проектов в компании — то, каким компания видит идеальный проект с точки зрения привлекательности и управляемости.

С помощью указания коэффициентов, приведенных в табл. 6.20, определяется стратегия управления портфелем проектов. Стратегия с точки зрения привлекательности определяет оценку важности проекта, то есть ориентацию проекта на выполнение стратегии компании, с весом 60 % от общей оценки. Оценки характеристики «срочность» будут учитываться с весом 45 %.



Например, при следующей оценке характеристик проекта № 1: важность — 1, спонсорство — 2, окупаемость — 3 — и характеристик проекта № 2: важность — 2, спонсорство — 1, окупаемость — 2 — итоговые оценки привлекательности проектов № 1 и 2 составят  $(1 \times 0,6 + 2 \times 0,15 + 3 \times 0,25) = 1,65$  и  $(2 \times 0,6 + 1 \times 0,15 + 2 \times 0,25) = 1,85$  соответственно. Иначе говоря, несмотря на то, что оценки характеристик «спонсорство» и «окупаемость» проекта № 2 ниже, чем оценки этих же характеристик проекта № 1, проект № 2 ближе к идеальному с точки зрения привлекательности в соответствии с текущей стратегией управления проектами компании.

То же и с параметром «управляемость». В настоящий момент наиболее интересующим фактором с точки зрения стратегии управления проектами, утвержденной в компании, являются проектные риски (40 %).

## Этапы управления портфелем проектов



С точки зрения этапности в управлении портфелем проектов в СУП выделяются следующие этапы.

- ♦ **Управление идеями.** На этом этапе формируются карточки проектов, содержащие перечень идей, обозначенные цели, бизнес-потребности в выполнении проекта, техническое видение, влияющее на выполнение проекта, оценку трудоемкости и рентабельности проекта. Выполняется на этапе операционного планирования или по запросу ССВ проекта на этапе оптимизации и балансировки проектов.

- ♦ **Идентификация и описание.** Актуализируются карточки проектов, оцениваются критерии привлекательности и управляемости проекта, которые запланированы к выполнению или уже выполняются, при необходимости уточняются принципы портфельного управления. Выполняется на этапе операционного планирования.
- ♦ **Оценка и выбор.** Оцениваются трудоемкость и рентабельность проектов, проекты приоритизируются, группируются по релевантности, строится сетевая диаграмма взаимодействия проектов. Выполняется на этапе операционного планирования.
- ♦ **Оптимизация и балансировка портфеля.** В проектах в процессе мониторинга и контроля хода выполнения проектов определяется необходимость оценки идей и пересмотра состава проектов из портфеля проектов. В случае возникновения такой необходимости ССВ проекта инициирует анализ портфеля проектов, после чего ССВ продукта принимает решение об актуальности ранее выявленных идей или корректирует их, утверждая на комитете по продуктам, отвечающем за согласованное движение по комплексному развитию всех продуктов компании.
- ♦ **Авторизация портфеля.** Комитет по продуктам утверждает сформированный портфель проектов к выполнению, приоритизацию идей и проектов. Приоритеты проектов влияют на приоритет обеспечения проектов ресурсами в случае ресурсных конфликтов.

## Специфика управления проектами в зависимости от портфелей



Бизнес-стратегия компании определяет следующие принципы проектного управления для каждого портфеля проектов (табл. 6.21).

**Таблица 6.21.** Принципы проектного управления в зависимости от типа портфеля проекта

Тип портфеля	Принципы проектного управления
1	2
Портфель корпоративных проектов	<p>Управление проектами менее формальное, чем портфелями адаптивных и клиентских решений (выражается в упрощенной форме устава проекта, отчетов о выполнении проекта).</p> <p>Создается только необходимый минимум проектных артефактов:</p> <ul style="list-style-type: none"> <li>• устав проекта (упрощенный);</li> <li>• план управления проектом (упрощенный): <ul style="list-style-type: none"> <li>– Communication Management Plan;</li> <li>– Quality Management Plan;</li> <li>– Risk Management Plan;</li> <li>– Requirements Management Plan;</li> </ul> </li> <li>• WBS проекта;</li> <li>• базовый план расписания проекта;</li> <li>• план управления документами;</li> </ul>

1	2
	<ul style="list-style-type: none"> <li>• реестр вопросов и ответов;</li> <li>• реестр рисков;</li> <li>• реестр проблем и решений;</li> <li>• реестр работ и артефактов (создается в WAS);</li> <li>• отчет о ходе выполнения проекта (упрощенный).</li> </ul> <p>С точки зрения разработки значительное внимание уделяется разработке архитектуры (при выборе ЖЦ проекта приоритетом пользуется ЖЦ спиральной итеративной разработки, управляемой архитектурой)</p>
<p>Портфель адаптивных решений</p>	<p>Степень формальности управления проектом зависит от отношений с клиентом и рисков коммуникаций с клиентами. По умолчанию управление проектом формальное, отчетность — формальная и регулярная.</p> <p>В случае адаптивного решения заказчиком проекта выступает менеджер по продукту.</p> <p>Устав проекта, требования к качеству, риски недостижения качества, описание содержания проекта, план управления проектом обязательно согласуются с заказчиком проекта.</p> <p>В обязательном порядке определяются перечень формальных вех проекта (milestones) и чек-листы по их прохождению</p>
<p>Портфель клиентских решений</p>	<p>Создается набор проектных артефактов:</p> <ul style="list-style-type: none"> <li>• реестр требований к качеству продукта;</li> <li>• реестр рисков по достижению требований к качеству продукта;</li> <li>• заявка на создание продукта/услуги;</li> <li>• предварительное описание содержания проекта;</li> <li>• описание содержания проекта;</li> <li>• RBS проекта;</li> <li>• устав проекта;</li> <li>• план управления проектом; <ul style="list-style-type: none"> <li>– Project Scope Management Plan;</li> <li>– Staffing Management Plan;</li> <li>– Communication Management Plan;</li> </ul> </li> <li>• правила проектной переписки;</li> <li>• правила формирования адресатов переписки;</li> <li>• каналы коммуникации;</li> <li>• правила использования каналов коммуникации;</li> <li>– Quality Management Plan;</li> <li>– Risk Management Plan;</li> <li>– Schedule Management Plan;</li> <li>– Requirements Management Plan;</li> <li>– Configuration Management Plan;</li> <li>– Procurement Management Plan;</li> <li>– Contract management Plan;</li> <li>• план управления документами;</li> <li>• WBS проекта;</li> <li>• базовый план расписания проекта;</li> <li>• реестр вопросов и ответов;</li> <li>• реестр рисков;</li> <li>• реестр проблем и решений;</li> <li>• реестр работ и артефактов (создается в WAS);</li> <li>• отчет о ходе выполнения проекта.</li> </ul>

Таблица 6.21 (продолжение)

1	2
	При выборе ЖЦ проекта ЖЦ спиралевидной итеративной разработки, управляемой архитектурой, и ЖЦ спиралевидной итеративной разработки, управляемой функциональностью, равнозначны и выбираются в зависимости от характеристик проекта и определенных требований к качеству проекта и продукта
Портфель проектной внутренней автоматизации и совершенствования процессов	<p>Управление проектами еще менее формальное, чем для портфелей корпоративных проектов (выражается в упрощенной форме устава проекта, отчетов о выполнении проекта и отсутствии многих проектных артефактов). Создается только необходимый минимум проектных артефактов:</p> <ul style="list-style-type: none"> <li>• устав проекта (упрощенный);</li> <li>• план управления проектом (упрощенный): Quality Management Plan; Risk Management Plan; Requirements Management Plan;</li> <li>• WBS проекта;</li> <li>• реестр вопросов и ответов;</li> <li>• реестр рисков;</li> <li>• реестр проблем и решений;</li> <li>• реестр работ и артефактов (создается в WAS).</li> </ul> <p>При выборе ЖЦ проекта наиболее приоритетен ЖЦ итеративной/водопадной разработки</p>

Во всех случаях управление проектами происходит по процедурам проектного управления, описанным в паттерне «СУП на РМВОКе».

### 6.5.5. Плохие паттерны

Самый плохой паттерн для этой позиции — неумение увидеть пробуксовывающие вопросы и принять на себя ответственность за преодоление причин этой пробуксовки. И вообще уход от ответственности в любом виде. Вы должны быть каменной стеной для своих сотрудников. Они должны физически ощущать вашу поддержку, хотя вас не видно и не слышно.

Другая сторона этого антипаттерна — несоответствие стиля управления уровню зрелости группы. Излишнее делегирование и дистанцирование, самоустранение от проблем. Хуже может быть только активная вовлеченность руководителя департамента в операционную деятельность в ущерб стратегическим вопросам.

### 6.5.6. Так что же такое управление подразделением разработки ПО

На мой взгляд, самый лучший руководитель подразделения разработки ПО — это генерал своей армии отцов-командиров и их воинов, мудрый, дальновидный, уверенный в себе. Его не видно, но все знают, что любое действие в подразделении производится в соответствии со стратегией, которую он определил.

У него есть видение того, куда и зачем он ведет всю армию, даже если его командиры такого видения не имеют. Он предвидит проблемы и активно снимает их. Он на переднем фронте в случае проблем и на задворках — в случае успеха.

# 7. Что умеет мастер управления проектной средой

---

---

Вот мы и подошли к заключительным главам этой книги. Очень хочется, чтобы эта книга, как и первая книга серии — «Путь аналитика», — принесла вам практическую пользу. В данной главе я еще раз кратко резюмирую, что же должен уметь мастер управления проектной средой.

Мастер управления проектной средой — это садовник, который терпеливо возделывает почву и формирует среду своего проекта, заботливо и аккуратно растит цветок своего проекта, планируя и выполняя действия, необходимые для достижения выдвинутых требований к качеству результата, и снимая риски, мешающие достичь этого результата. Цветок должен заставить заказчика, увидевшего результат, сказать: «Так вот ты какой, цветочек аленький», — и уверенно взять его в руки, чтобы получить реальный инструмент удовлетворения своих ожиданий.

Кроме этого, мастер управления проектной средой — это наставник, лидер, психолог, старший товарищ и отец-командир для своей команды. Он умеет управлять энергетическим полем проекта, мотивировать сотрудников на достижение результата и знает, как превратить независимых одиночек в команду профессионалов, работающих слаженно, с интересом, удовольствием и осознанием собственного вклада в выращиваемый цветок. Профессионалов, причастных к конечному результату настолько, чтобы и через семь лет каждый из них мог с гордостью продемонстрировать «майку лидера» с надписью «Самый трудный проект 2005 года» и восхищенно сказать: «Вот это была команда! Какой проект мы сделали! Какой продукт мы выпустили!»

Ну и еще, мастер управления проектной средой — это капитан, который умеет вести корабль проекта сквозь рифы и мели, владея методологиями разработки ПО и управления проектами, зная, как обходить рифы, выпаривать пресную воду из соленой морской, который уверенно стоит на палубе и в шторм, и в шторм, не теряя из виду курс и звезды (цели проекта).

Вот такие метафорические образы родились у меня в заключение нашего с вами совместного пути. И именно сейчас я хотел бы поделиться с вами еще одним паттерном.

## 7.1. Паттерн «Выживание в проектной среде»



В книге «Энциклопедия относительного и абсолютного знания» [16], которую я рекомендую просто как сборник интереснейших фактов и историй, ее автор Бернард Вебер пишет: «В 1974 году философ и психолог Анатолий Рапопорт из Торонтского университета высказывает мысль о том, что самая эффективная манера общения людей заключается в:

- ◆ сотрудничестве;
- ◆ взаимности;
- ◆ прощении.

Другими словами, если индивидуум, структура или группа сталкиваются с другими индивидуумами, структурами или группами, им выгоднее всего искать союза. Затем важно по закону взаимности отплатить партнеру тем, что получаешь от него. Если он вам помогает, помогите, если он нападает на вас, нападите в ответ таким же образом и с той же интенсивностью. И наконец, надо простить и снова предложить сотрудничество.

В 1979 году математик Роберт Аксельрод организовал конкурс между автономными компьютерными программами, способными реагировать, подобно живым существам. Единственным условием было: каждая программа должна быть обеспечена средством коммуникации и иметь возможность дискутировать с соседями.

Роберт Аксельрод получил 14 дискет с программами, которые прислали его коллеги из разных университетов. Каждая программа предлагала разные модели поведения (в самых простых — два варианта образа действий, в самых сложных — сотня). Победитель должен был набрать наибольшее количество баллов.

Некоторые программы пытались как можно быстрее начать эксплуатировать соседа, украсть у него баллы и сменить партнера. Другие старались действовать в одиночку, ревниво охраняя свои достижения и избегая контакта с теми, кто способен их украсть. Были программы с такими правилами поведения: «Если кто-то проявляет враждебность, нужно попросить его изменить свое отношение, затем наказать». Или: «Сотрудничать, а затем неожиданно предать».

Каждая из программ двести раз вступала в борьбу с каждой из конкуренток. Программа Анатолия Рапопорта, вооруженная моделью поведения СВП (сотрудничество, взаимность, прощение), вышла победительницей.

Более того, программа СВП, помещенная наугад в гущу других программ и сначала проигрывавшая агрессивным соседкам, не только одержала победу, но и «заразила» остальных, как только ей дали немного времени. Соперницы поняли, что ее тактика наиболее эффективна для зарабатывания очков, и подстроились под нее. Проверка временем доказала правильность метода. Дело тут не в доброте, а в вашей собственной выгоде, доказанной компьютерными программами».

Это именно та модель эффективного взаимодействия в не очень дружелюбной среде, к которой я пришел на собственном опыте. Она работает. Как жаль, что я прочел эти строки почти в 40 лет. Впрочем, не уверен, что в 20 лет я бы понял, о чем они...

---

## 7.2. Паттерн «Особенности выполнения государственных контрактов и взаимодействия с государственными заказчиками»

Я решил выделить паттерн по работе с госсектором по причине того, что выполнение оплачиваемых из госбюджета проектов обладает особой спецификой.

Этот паттерн я решил разместить в главе 7 «Что умеет мастер управления проектной средой», потому что это хорошая иллюстрация «выживания» и использования специфики проектной среды в интересах проекта. IT-проекты, оплачиваемые государством, были, есть и будут одними из самых масштабных и интересных проектов, особенно в плане импортозамещения, что сейчас, в 2015 году, приобрело крайнюю важность. Я уверен, что этот паттерн будет вам полезен.

Обратите внимание на то, что **данный паттерн применим и в работе с крупным корпоративным сегментом**, в деятельности которого вопросы регламентных процедур и взаимодействий зачастую играют не меньшую роль, чем в органах государственной власти.

Одной из ключевых особенностей работы с государственным сектором является строгое регламентирование законодательством всей деятельности государственных служащих, в том числе процесса размещения и выполнения государственных контрактов.

### 7.2.1. Заключаем государственный контракт

Основной момент, о котором вы должны знать и помнить при реализации проекта с государственными органами, — в госконтракт практически невозможно внести существенные изменения! Ни по объему работ, ни по срокам, ни по стоимости.



Исключения составляют отдельные случаи, предусмотренные законодательством, например **возможность изменения не более чем на 10 % объема реализуемых работ или поставляемых товаров с обязательным пропорциональным изменением цены контракта. Но такое изменение возможно только в том случае, если оно изначально было предусмотрено документацией о закупке и контрактом.**

---

Старайтесь **всегда** прописывать эту возможность в документации и контракте. Из-за неопределенности требований на этом этапе вы зачастую можете недооценить объем работ. Кроме того, в ходе выполнения работ из-за неоднозначности требований, что также нормально для этого этапа, заказчик будет вносить улучшения и доработки, которые для вас будут означать изменение скоупа проекта.

Это единственно возможный случай, когда вы можете попытаться торговаться с заказчиком: «О'кей, мы делаем то и это, но давайте тогда вот это вычеркнем», — в этом случае вы реализуете чуть больший объем функциональности, чем планировали, но снизите риски упасть ниже нормы рентабельности по проекту. А вопрос пропорционального изменения цены контракта можно постараться решить так, что ее увеличение на 10 % покроет ваши издержки на реализацию дополнительных желаний заказчика.

Отсюда вытекает самый важный момент — тонкости выполнения государственного контракта важно предусмотреть до его заключения. Давайте рассмотрим данную тему подробнее.

Так как ответственность за выполнение работ по госконтракту в срок и с должным качеством возникает у вас сразу после его подписания, а продолжение работ и их качественное выполнение не всегда зависят только от вас, стоит **аргументированно доказать, что для достижения общих целей госконтракта необходимо, чтобы в нем была прописана ответственность заказчика за предоставление нужной вам информации**, влияющей на выполнение работ по контракту, поскольку ее отсутствие делает дальнейшее его выполнение невозможным. И поверьте, эта якобы простая формальность не один раз спасет вас в ходе реализации проекта.



Например, вы должны выполнить работы по внедрению программного обеспечения. Для того чтобы вы могли их начать, заказчик должен предоставить соответствующие технические средства определенной характеристики. Но по какой-то причине (например, поздно провели процедуру закупки этой техники) заказчик тянет с решением данного вопроса. Или сдвигает сроки предоставления нужных серверов. Даже при возникновении такой ситуации на вас лежит ответственность за выполнение работ в срок, хотя в переносе срока начала работ вы не виноваты.

Наличие в договоре пункта об ответственности заказчика за предоставление необходимой информации, нужных технических ресурсов и т. п. **как минимум мотивирует его решать вопросы быстрее**. В официальном письме к заказчику вы всегда сможете сослаться на данный пункт контракта. Это придаст дополнительную мотивацию конкретному исполнителю все-таки заняться вопросом и дать вам какой-то ответ. В противном случае у вас не будет рычагов воздействия на исполнителей с той стороны из-за специфики проектной среды.



На этапе проведения процедуры закупки (если вы планируете участвовать в конкурсе) внимательно изучите конкурсную документацию и **обязательно уточните у заказчика те моменты, где вы видите риски**.

Постарайтесь, чтобы каждый ваш вопрос содержал ответ — тот, который нужен вам. Например, вы видите, что в ТЗ указана реализация функции, которая звучит слишком общо или предполагает двоякое толкование. Согласно действующему законодательству (Федеральный закон от 5 апреля 2013 г. №44-ФЗ «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд») вы можете отправить официальный запрос о разъяснении условий документации. При этом лучше, если он будет начинаться со слов: «Правильно ли мы понимаем, что...» — а дальше излагать приемлемый для вас вариант.



---

Подобная форма коммуникаций предпочтительна при выполнении не только проектов для государственных органов, но и любых проектов, где нет четко-го владельца процессов, вовлеченного в выполнение работ. То есть там, где исполнители или очень опосредованно заинтересованы в результатах, или участвуют в проекте вынужденно, в силу своих должностных обязанностей или прямого указания руководства.

---

Качественное выполнение проекта требует выделения трудовых ресурсов и от заказчика.



---

Постарайтесь убедить заказчика в необходимости внесения в государственный контракт (ГК) пункта о том, что обязанностью заказчика является определить ответственных за проект лиц, которые должны уделять ему не менее определенного количества своего рабочего времени (как правило, 20–50 % в зависимости от сложности контракта). Это будет дисциплинировать заказчика и даст возможность ответственному лицу официально частично освободиться от рутинной работы и заняться проектной деятельностью. Иначе в ответ на свои вопросы вы можете услышать, что он и так загружен работой, — и в большинстве случаев это будет правдой.

---

В дальнейшем, отталкиваясь от этой информации, нужно составить приказ/распоряжение о создании рабочей группы.

## 7.2.2. Инициация проекта

При старте проекта детализируйте содержащуюся в ГК информацию о том, какие должности с какими временными затратами и в каких ролях будут выступать в проекте. Окончательно зафиксировать договоренности лучше **приказом о создании рабочей группы** по профильному министерству. Так, например, в случае внедрения системы целеполагания при планировании и составлении бюджета субъекта РФ такими министерствами являлись министерство финансов субъекта РФ и министерство экономического развития.



---

В этом приказе должны фигурировать конкретные фамилии и привязка этих фамилий к ролям в проекте. Это следующий шаг по детализации контракта.

---

Такой же приказ вы должны издать и по своей организации.

Также постарайтесь создать и подписать документ, который, по сути, будет чем-то средним между уставом проекта и планом управления проектом по РМВОК. В моей практике такой документ назывался «Регламент взаимодействия рабочих групп», просто потому, что такая формулировка близка и понятна чиновникам.

В этом регламенте вы можете попытаться установить правильные методы и практики управления проектом. Тем не менее всегда помните, что с формальной точки зрения единственным (но, как правило, недостаточным для вас) регламентом исполнения проекта является ГК. Но если у вас получится создать и, что самое главное, подписать такой документ, это пойдет на пользу всему проекту и всем заинтересованным сторонам. Вы можете в нем прописать планы управления коммуникациями, рисками, конфигурацией, требованиями и т. д.

Теперь давайте поговорим о заинтересованных сторонах...

### 7.2.3. Психология государственного служащего

Специфика работы чиновника на госслужбе в том, что он выполняет определенные функции. Это не менеджер, нанятый для реализации определенного проекта, — как правило, для него персонально неважен результат выполнения проекта.

Бывает даже так, что он вообще не заинтересован в проекте напрямую, потому что его основная мотивация — не сделать результативный проект, а выполнить свою функцию, сохранить должность и продемонстрировать начальству хорошую работу.



И это не хорошо и не плохо — это естественная природа вещей. Это среда вашего проекта, которую вы должны понять, прочувствовать и использовать в своих интересах.

Из положительных сторон и особенностей работы с чиновниками, в отличие от бизнеса, стоит отметить то, что они не ориентированы на конфликт, **никто не преследует цель изначально занести вас в список недобросовестных поставщиков**. Никто в этом не заинтересован.

Заказчик всегда изначально нацелен на то, чтобы закрыть контракт. В бизнесе спонсор проекта может неожиданно принять решение о разрыве договора из-за неудовлетворенности ходом выполнения проекта или его промежуточными результатами. Здесь возможно существенное расширение скоупа проекта с попыткой «продать» вас по срокам и бюджету.

В случае с ГК у вас появляются преимущества, обусловленные природой и ограничениями самого ГК, — рамки проекта определены, и разрыв контракта по инициативе спонсора значит для него поставить под сомнение основную мотивацию: я выполняю свои функции, сохраняю должность и показываю начальству хорошую работу. Эта мотивация работает для госчиновников практически любого уровня.

Поэтому **договаривайтесь**. В сложных случаях предлагайте альтернативные варианты. При этом, конечно же, вы должны помнить, что чиновник — лицо ответственное и подотчетное. Все ваши предложения по проекту и его выполнению не должны быть рискованными для него.

## 7.2.4. Формализация процесса выполнения контракта

Конкурсные процедуры и заключаемые на их основании государственные контракты значительно формализованы, поэтому первое, о чем необходимо помнить при выполнении контрактов, — формализация самого процесса выполнения контракта.

Ведение бумажного и электронного документооборота — один из гарантов успешного выполнения проекта.

**Все договоренности и принятые решения фиксируйте в виде протоколов.**



---

Старайтесь письменно оформлять все важные договоренности, фиксирующие решение вопросов, связанных со сроками или объемом выполнения контракта, составом выполняемых работ, ответственными исполнителями и др. Лучшая форма оформления таких договоренностей — дополнительное соглашение к договору, однако при работе с госсектором такой возможности у вас не будет — ГК должен быть исполнен в согласованном объеме в согласованные сроки. Поэтому **единственно возможная форма — официальные протоколы**.

---

Это не спасет вас от необходимости исполнить обязательства по контракту, но часто является эффективным инструментом влияния на исполнителей со стороны госорганов.

Старайтесь документально оформлять открытые вопросы, требующие решения заказчика — государственного органа. Протоколы отправляйте участникам совещания по электронной почте с требованием в электронном виде согласовать протоколы. Протоколы с важными решениями по ГК с подписями сторон стоит иметь и в бумажном виде.

Зачастую количество участников совещания настолько велико, что будет сложно собрать все подписи, требующиеся по протоколу. Но даже если в отдельных случаях у вас не получится собрать подписи всех сторон протокола, все равно будет зафиксировано решение заседания рабочей группы по выполнению контракта. Не стоит этим пренебрегать и недооценивать силу документа.

Психология государственного служащего в следующем: ставя свою подпись под протоколом, он понимает, что, нарушая его, как минимум рискует не достичь одной из трех целей деятельности на своей должности. А значит, хоть какое-то внимание с его стороны к предмету подписания гарантировано.

Иногда получить такую подпись довольно сложно, для этого старайтесь, чтобы в самом начале пути в профильном министерстве, отвечающем за проект (или его часть), был издан приказ о рабочей группе.

## 7.2.5. Официальные письма

Как протоколы — правильный инструмент для фиксирования договоренностей, так и официальные письма — отличный инструмент для фиксирования открытых вопросов, требующих решения от государственного заказчика. Официальные письма необходимы всегда в случае, когда нужно уведомить заказчика о каком-либо важном факте, связанном с реализацией или сложностями в реализации ГК. Электронные письма принимать во внимание не будут.

Старайтесь в официальных письмах четко сформулировать вопросы и **по возможности предложить решение**, интересное вам и потенциально устраивающее заказчика.

Плюс официальных писем в том, что вам обязательно ответят. Это предусмотрено все теми же формализованными процедурами. Официальное письмо будет передано исполнителю, который постарается если не максимально эффективно, то как минимум по сути дать вам ответ. Хотя, вероятно, одним письмом вы не отделаетесь и придется вступить в официальную переписку.

Эффективность этой переписки во многом зависит от вас. Задавайте конкретные вопросы, предлагайте варианты ответа. Если вам нужна какая-то информация, старайтесь мотивировать чиновника к выполнению его базовых функций. Например, составляйте запрос по такой форме: «Для исполнения такого-то пункта контракта в такие-то сроки нам необходима такая-то информация. Прошу предоставить эту информацию до такого-то числа».

## 7.2.6. Идите со всем готовым

В заключение, возможно, повторюсь, но считаю важным еще раз это сделать. **Не задавайте только вопросы, старайтесь всегда предложить решение, идите к чиновнику с продуманным вариантом действий.**

Это золотое правило работы в проектах по ГК, да и в принципе с любым заказчиком, у которого выполнение проекта не связано с решением конкретной проблемы, где он лучше вас знает, что именно ему нужно.

Госслужащих часто ругают из-за низкой инициативности и эффективности их деятельности. Да, это тоже есть. Но особенностями госслужбы является то, что здесь много зачастую рутинной ежедневной работы, низкая квалификация персонала и нет возможности, времени и сил подумать над тем, **как** это сделать лучше. К тому же в нашей стране в последние несколько лет (если не десятков лет) постоянно идут какие-то реформы в сфере то образования, то здравоохранения, то финансов и т. д. Все это отнюдь не способствует уменьшению работы чиновника, наоборот, увеличивает, и при этом никто, естественно, его текущую работу не отменяет.

Поэтому раз именно вы выполняете проект и объективно имеете больше возможностей для формирования квалифицированной команды проекта с вашей стороны — команды, которая знает, как лучше, и умеет предложить эффективные решения, то всегда предлагайте решение, всегда (когда это целесообразно, а целесообразно почти всегда и везде) **занимайтесь подготовкой различных концепций, планов, документов, мероприятий** и т. п. Проще говоря, вы должны быть готовы работать в проекте за обе стороны. Тогда и только тогда проект будет идти оперативно. Если заказчик тоже хочет поучаствовать в решении проблем и готов что-то взять на себя, поверьте, он даст вам об этом знать.



---

Например, проект по внедрению программного обеспечения. Для продолжения работ по нему заказчик должен определиться, как будет автоматизирован отдельный процесс. Конечно же, в данном примере речь не идет о сложных специфических процессах, постановка которых сама тянет на отдельный государственный контракт и требует участия людей со специфическими знаниями функциональности соответствующего сектора деятельности государственных органов. У вас есть два варианта действий: или вы ждете от заказчика, когда он скажет, **как** ему надо (а с вероятностью 99 % он пока и сам этого не знает), или проводите мозговой штурм силами **своей команды**, по возможности вовлекая в процесс самых инициативных исполнителей со стороны заказчика, генерируете идеи и предлагаете для обсуждения заказчику процесс «как должно быть». Работать с выдвинутыми идеями всегда легче. К тому же не забывайте: у заказчика часто нет людей, нет нужной квалификации и мало времени.

---

# 8. Кадры решают все

---

---

Важнейшим навыком менеджера является способность определить умения и потенциал специалистов — кандидатов в вашу команду. В широком смысле этот навык относится к профессиональной сфере рекрутера (специалиста по найму), который ищет и отбирает кандидатов по собственным методикам. Классический подход рекрутера заключается в утверждении, что нет плохих или хороших кандидатов, но есть подходящие или не подходящие для данной вакансии.

Задача рекрутера — закрыть вакансию путем поиска лучшего кандидата на рынке труда в установленные сроки, а ваша задача — найти «товарища по оружию», который будет работать и развиваться вместе с вашей командой. Менеджер несет прямую ответственность за подбор членов своей команды, то есть жизненно заинтересован в результате. В зависимости от принятой в компании практики набора специалистов менеджер либо взаимодействует с HR-подразделением, либо сам в той или иной мере выполняет функции рекрутера.

Зачастую рекрутеры просто сопоставляют профиль кандидата с эталоном, а более продвинутые из них даже пытаются постичь суть профессиональных терминов типа «сервер Apache» и «язык UML». Если поиском занимается не сам менеджер, а рекрутер, он должен понимать эти особенности и действовать в связке с менеджером. В этом случае задача менеджера — составить для рекрутера исчерпывающий профиль должности, который позволит ему максимально точно проводить отбор.

## 8.1. Паттерн «IT-рекрутинг»

Современный IT-рекрутинг выделяют в отдельную дисциплину, где глубже рассматривается личностный портрет IT-специалиста, разбираются IT-компетенции и т. д. В качестве отличительной особенности профиля кандидатов выделяют повышенные требования к интеллектуальным способностям и знание широкого перечня технологий, набор которых непрерывно меняется.

Технология работы рекрутера совмещает пассивные и активные мероприятия, а также создание HR-бренда компании.

Классический пассивный поиск — это размещение объявления, желательно креативного содержания. Объявление размещается на сайте компании и на специализированных ресурсах (hh.ru, superjob.ru и т. п.). Требования должны быть описаны доступным и понятным языком, обязательно следует указать преимущества работы в вашей компании, соответствующие мотивирующим факторам. Дополнительно советую запрашивать у соискателей вместе с резюме примеры их реальных работ (готовых или выполненных по вашему заданию), а у недавних выпускников вузов — копию диплома с оценками.

Главная витрина вакансий — это сайт компании. Впечатление от сайта будет первым впечатлением о компании потенциального соискателя должности. Именно с раздела сайта с призывом «Приходите к нам работать!» необходимо начинать формирование HR-бренда IT-компании.



---

«HR-бренд — образ вашей компании как хорошего места работы. Здесь следует указать набор экономических, профессиональных и психологических выгод, которые получает работник, и способ, которым вы формируете идентичность своего бизнеса, начиная с базовых основ и ценностей и заканчивая тем, как вы доносите ее до заинтересованных лиц» [20].

---

### 8.1.1. Что важно?

Рассказать о компании: сколько лет работает, миссия, сведения о коллективе, социальная политика, корпоративные мероприятия, последние достижения, преимущества работы в компании, фотографии офисных помещений и т. п. Будет замечательно, если вы закажете креативный видеоролик.

Привести информацию о реальных проектах: применяемые новейшие технологии, достигнутые успехи, отзывы клиентов, личные достижения и уникальные профессиональные навыки сотрудников (только выбирайте самых лояльных, иначе их начнут переманивать конкуренты).

Показать, что руководство — это тоже люди, особенно если директор — настоящий IT-гений. А лучше запишите обращения топ-менеджеров на видео.

Заманить «печеньками», креативно описав то, что отличает вашу компанию от других: социальный пакет, свобода выбора одежды и рабочего графика, спортивный уголок, велопарковка, бесплатный кофе и самые удобные в мире кресла, профессиональный рост и обучение у лучшего в России специалиста по BigData.

Описать вакансии: укажите требуемые компетенции, опыт, уровень дохода, нужную форму резюме, а также способ связи с вами. Проследите, чтобы требования были написаны понятным языком.

Система пассивного поиска новых специалистов в компании должна работать непрерывно. Однако если вам нужны уникальные специалисты высокой и сверхвысокой квалификации, то пассивным поиском, вы их, скорее всего, не найдете. Самые лучшие специалисты всегда трудоустроены, довольно редко меняют место работы и сами могут выбрать, где им было бы интересно работать.



---

«Отношение производительности труда лучших сотрудников к показателю заурядных составляет примерно 10:1, и, как показывает практика, производительность не зависит от должности, опыта и оплаты труда» [21]. «Выдающиеся программисты — да и лучшие специалисты в любой отрасли — просто никогда не появляются на рынке. Выдающийся программист за свою карьеру может сменить примерно четыре места работы» [22].

---

Только активный поиск позволит найти настоящих «звезд», после чего ваша задача — найти для них персональный мотив для перехода в вашу компанию.

## 8.1.2. Охота за головами

Активный вариант поиска — это работа в социальных сетях, специализированные сервисы (LinkedIn, AmazingHiring), профессиональные форумы разработчиков и охота за головами на профессиональных IT-мероприятиях. Хороших кандидатов можно найти по рекомендации собственных сотрудников, среди работников конкурирующей компании, клиентов, поставщиков и т. п. Иными словами, при активном поиске вы должны сами найти кандидата и варианты выхода на него с предложением о работе.

Активный поиск как раз и называется охотой за головами: подумайте, кого искать, где такие трудятся и как их заманить к себе. Будьте настоящим охотником: выслеживайте, проявляйте терпение и бейте наверняка.

Следующий шаг — подбор мотивации для найденного кандидата, ведь лучшие специалисты чаще всего на хорошем счету и руководство их ценит. Очень часто, когда «звездный» специалист озвучивает руководству свое желание сменить работу, ему моментально предлагают передумать, подкрепив предложение весомыми аргументами.

Помните афоризм «Самый лучший охотник — тот, кто имеет личные счета к жертве»? Давайте разбираться, что же мотивирует интересующих нас кандидатов.

Представители отрасли обладают особым менталитетом, ценят личную независимость и красоту решаемых задач. Их жизненное кредо — непрерывное развитие, профессиональные амбиции и профессиональная гордость. Размер заработной платы является скорее обязательным условием, а истинные факторы мотивации надо искать немного глубже.



---

О настоящих программистах. Когда мне было 10 лет, ребята во дворе показали мне, как программировать на Basic. Я не знал, что это вызовет такую зависимость. Сегодня пишу на C++. Не пойму, за что мне платят деньги? Я бы делал это бесплатно...

---

Чем выше творческий потенциал и профессиональный опыт, тем большее значение имеет верхний уровень пирамиды потребностей человека (см. паттерн «Маслоу по Фрейд»). Обобщенно я бы сформулировал это явление как потребность к самореализации творческой личности.

В табл. 8.1 приведена моя оценка приоритетов мотивирующих факторов, которыми руководствуются IT-специалисты при выборе места работы. Важно помнить, что по мере профессионального развития от стажера до руководителя приоритеты мотивации существенно меняются.

**Таблица 8.1.** Мотивирующие факторы при выборе места работы

Мотивирующие факторы (они же составляющие HR-бренда)	Степень влияния при выборе или смене места работы		
	Стажер/младший специалист	Специалист/старший специалист	Ведущий специалист/руководитель
Система оплаты труда, социальный пакет	Высокая	Средняя	
Условия труда: офис, правила, коллектив	Средняя	Высокая	
Социальная защищенность, стабильность компании и проекта		Высокая	Средняя
Степень уважения компании в профессиональной среде, насколько «модные» технологии и крутой проект		Высокая	Высокая
Возможность профессионального роста: наставники, возможность обучения	Высокая	Средняя	
Возможность карьерного роста		Средняя	Высокая
Повышение сложности задач и степени самостоятельности (ответственности) в проекте	Средняя	Высокая	
Личная оценка, возможность широкого признания достижений			Высокая

Исходя из приведенных факторов, следует выстраивать HR-бренд компании, составлять креативные объявления, а также оценивать вероятность ухода специалистов из вашей команды.

## 8.2. Паттерн «IT-интернатура»

Поиск состоявшихся специалистов — это один из возможных сценариев пополнения вашей команды. Альтернативный способ заключается в том, что вы выращиваете нужных специалистов в условиях стажировки или IT-интернатуры. В дальнейшем я буду называть таких специалистов-стажеров просто интернами.

Если в какой-то момент вы начнете ощущать кадровый голод, систематически не находя на рынке труда подходящих кандидатов, значит, наступил момент

начинать работу с интернами. А правильнее будет сказать, что этот момент наступил еще полгода или год назад.

Иными словами, на определенной стадии развития появление интернов становится обязательным условием сохранения темпов роста компании. Эта стадия может наступить прямо в момент основания компании, если вы не имеете возможности нанять готовых специалистов, например, потому, что не можете им много платить, а ваш офис состоит из одной маленькой комнатки далеко не в самой престижной части города.

Моя практика IT-рекрутинга и работы с интернами показала, что студенты-интерны остаются высоколояльными компании во время обучения в вузе и в среднем в первые два года после завершения обучения. В этот период они редко меняют место работы. Да и в дальнейшем лояльность бывших интернов всегда выше, чем прочих специалистов.

Второй важный эффект из моей практики: хороший ученик всегда развивает своего учителя. Первоначально вы вкладываете в интерна, направляете его развитие, а через какое-то время уже интерн, будучи ориентированным на развитие, подстегивает развитие вашей команды.



---

Работая с интернами, вы получаете возможность выбрать среди большой массы новичков наиболее талантливых и перспективных, а затем выработать и развить у них нужные вам знания и навыки. Важнейшим эффектом работы с интернами становится появление новых «звезд», которые длительное время сохраняют высокую лояльность вашей команде и проекту, в котором принимали участие.

---

**Типовой профиль интерна.** Нам необходимы молодые специалисты, которые обладают отличным потенциалом развития, высокомотивированы и имеют базовые теоретические знания, однако заведомо не имеют нужных вам профессиональных компетенций и опыта. В зависимости от ситуации в вашей компании в этот список можно добавить низкие зарплатные ожидания и, что очень важно для стартапов, высокую степень внутренней личной мотивации на достижение результатов.

Идеальным интерном можно считать студента или выпускника высшего учебного заведения с профильным образованием, однако на практике бывает, что интересные интерны получают из кандидатов-самоучек без IT-образования.

**Идем в вузы.** Искать потенциальных интернов лучше всего в вузах. Действующие федеральные образовательные стандарты предусматривают широкое взаимодействие учебных заведений с предприятиями вплоть до создания базовой кафедры по интересному вам направлению подготовки. Приведу перечень вариантов работы в порядке роста трудоемкости реализации.

1. Проводить разовые занятия со студентами.
2. Принимать студентов на производственную практику.
3. Приглашать студентов и выпускников в интернатуру.

4. Организовывать студенческие конкурсы, гранты.
5. Читать студентам одну или несколько дисциплин.
6. Организовать базовую кафедру от вуза.

Отличными критериями оценки студентов являются отзывы преподавателей вузов и высокий средний балл в дипломе (зачетке), конечно же, при условии, что вы доверяете конкретному учебному заведению.

**Как работать с интернами?** Важно понимать: мы рассчитываем на потенциал интернов, а не на достижение быстрых производственных результатов. Будьте терпеливы, но настойчивы.

Если уж мы договорились называть интернами наших высокомотивированных новичков, вам волей-неволей придется называться не только менеджером, но и наставником. Я бы рекомендовал менеджеру лично заниматься интернами или как минимум лично отбирать их и оценивать потенциал развития.

Количество интернов должно быть заведомо больше ваших сегодняшних потребностей в специалистах, так как не все они пройдут интернатуру, да и не всегда можно спрогнозировать, сколько персонала вам потребуется через год. Продолжительность интернатуры должна составлять не менее трех месяцев, а для студентов — не менее шести месяцев. Стандартной вузовской производственной практики будет явно недостаточно.

Итак, вы набрали интернов. Ваша первая задача — закрепить их за членами вашей команды. Еще лучше, если несколько интернов будут совместно работать над одним проектом под патронажем одного наставника. Помните, что многие IT-специалисты слишком требовательны к себе и другим, а потому ждут от интерна результата прямо здесь и сейчас. Задача истинного наставника — задать вектор развития и разглядеть потенциал интерна (или убедиться в его отсутствии), а это дано не каждому.

Интерн должен вписаться в коллектив и усвоить корпоративную культуру. Вы должны убедиться в таких его качествах, как способность работать в команде, взаимодействовать с другими людьми, подчиняться и выполнять установленные правила. Все эти качества в интерне тоже можно развить, за исключением особо безнадежных случаев.

Интерн должен конкретной работой доказать наставнику способность учиться, развиваться и доводить дело до конца. Первые проекты, которые я поручаю студентам, часто носят исследовательский характер, а результат работы представляет собой прототип решения, который не обязательно будет применен в реальном производстве. В дальнейшем новые задачи всегда должны быть привязаны к вашему производственному процессу.

Интерн должен знать об оплате своего труда с первого дня пребывания в компании: либо вы платите сразу, либо обозначаете точную дату, когда будете готовы обсуждать оплату его труда (или ее повышение до уровня зарплаты обычного специалиста). Студенты более охотно выбирают для производственной практики то место, где они выполняют реальную работу, а их труд оплачивается.

В отдельных компаниях существует практика, когда на работу берут почти неограниченное число студентов-практикантов какой-либо профильной кафедры и всем без исключения платят небольшую стипендию, расширяя таким образом исходную базу для отбора будущих специалистов.

Кто из интернов останется работать у вас, а от кого нужно отказаться, должен решать менеджер по рекомендации конкретного наставника.

## 8.3. Паттерн «Профиль должности»

Профиль должности необходим для постановки задачи поиска специалиста. Он представляет собой описание требований: профессиональных компетенций и личностных характеристик, которым должен соответствовать кандидат для работы в вашей команде.



---

С точки зрения интересов бизнеса, компетенцией специалиста следует считать продемонстрированную способность применять знания и навыки для достижения конкретных результатов. Уровень компетенции определяется сложностью задач, от выполнения типовых операций до инновационных и творческих задач, для решения которых необходимы когнитивные и аналитические способности.

---

Источником информации для заполнения профиля должности могут служить паттерн «Фундамент отдела», а также профессиональные стандарты в сфере ИТ, утвержденные Министерством труда и социальной защиты России. Эти стандарты отражают современные требования к ИТ-специалистам в виде трудовых функций, привязанных к уровням профессионального развития вплоть до руководящих должностей. В ближайшем будущем профессиональные стандарты должны стать отраслевым эталоном для независимой оценки квалификации специалистов.

Советую также ознакомиться с ГОСТ Р 55767-2013 и ГОСТ Р 56156-2014, которые идентичны международным стандартам, описывающим европейскую рамку компетенций e-CF ([www.ecompetences.eu](http://www.ecompetences.eu)). Полезно также ознакомиться с моделью классификации ИТ-навыков SFIA ([www.sfia-online.org](http://www.sfia-online.org)).

Профиль должности может применяться как оценочный шаблон при проведении собеседования как менеджером, так и рекрутером, который в любом случае захочет получить от вас такого рода документ.

Итак, составляем профиль должности за пять простых шагов.

### 8.3.1. Шаг 1. Определяем должностной уровень

Важно определить уровень развития требуемого специалиста, то есть того, кто вам нужен именно сейчас, по шкале профессионализма от младшего инженера до

руководителя группы. Хороший вариант ранжирования профессиональных навыков принят в e-CF, где профессионализм определяется тремя составляющими: сложностью, самостоятельностью и ответственностью. Именно эти три составляющие вы и должны раскрыть в контексте своей производственной деятельности.

Далее приведена упрощенная таблица примерной оценки должностного уровня, подготовленная с учетом ряда положений e-CF и SFIA (табл. 8.2).

**Таблица 8.2.** Примерная оценка должностного уровня, подготовленная с учетом ряда положений e-CF и SFIA

Должность	Сложность	Самостоятельность	Ответственность
Младший специалист, стажер	Решение типовых практических задач под руководством наставника	Планирование своей работы	Индивидуальная ответственность
Специалист	Решение различных типов задач, требующих анализа ситуации	Планирование работы группы в рамках задачи	Ответственность за решение задачи, за проект
Старший специалист	Решение задач проектирования, применение технологических и методологических знаний	Самостоятельная деятельность, проявление инициативы	Ответственность за результат работы других людей
Ведущий специалист	Управление направлениями профессиональной деятельности, выработка технологических или методических решений	Определение собственных задач и задач подчиненных, взаимодействие со смежными подразделениями	Ответственность за результаты перед клиентами
Менеджер, руководитель, архитектор	Решение инновационных задач, развитие области профессиональной деятельности, разработка новых методов и технологий	Определение стратегии, управление производственными процессами	Ответственность за выбор целей и результаты деятельности компании

Дополнительно следует описать варианты взаимодействий при исполнении должностных обязанностей (табл. 8.3). Такое описание поможет нам на следующих шагах.

**Таблица 8.3.** Варианты взаимодействий при исполнении должностных обязанностей

Направление взаимодействия	Вариант описания требований (пример)
Внутри своей функциональной команды	Достаточно индивидуального выполнения заданий
В рамках проекта	Обсуждает постановку задачи с системными аналитиками, согласовывает архитектуру и дизайн с менеджером по продуктам и т. п.
Внутри компании	Готовит тендерное предложение по заданию группы сбыта, согласует закупки с финансовым департаментом и т. п.
Взаимодействие с клиентами (вариант: с обслуживаемыми подразделениями)	Проводит демонстрации продукции, отвечает за согласование технического проекта, выполняет работы по предварительному обследованию объектов, согласует вопросы в рамках управления качеством проекта, работает в службе поддержки и т. п.

## 8.3.2. Шаг 2. Определяем перечень профессиональных компетенций

Вам необходимо определить исчерпывающий перечень требуемых компетенций с учетом должностного уровня, определенного на первом шаге.

Источник информации — база профессиональных стандартов, представленная на сайте Минтруда России (<http://profstandart.rosmintrud.ru/>) (табл. 8.4). В структуру каждого из стандартов заложены градации профессионализма и требования к уровню образования и стажу работы. Здесь очень пригодится также информация о взаимодействии, подготовленная на первом шаге. Так, например, требования к младшим должностям программиста не предусматривают взаимодействия вне своей команды, а вот для старшего инженера наличие этого навыка является обязательным.

**Таблица 8.4.** Перечень стандартов, актуальных на момент написания книги

Код	Вид профессиональной деятельности	Профессиональный стандарт
06.001	Разработка программного обеспечения	Программист
06.003	Проектно-конструкторская деятельность	Архитектор программного обеспечения
06.004	Проектно-конструкторская деятельность	Специалист по тестированию в области информационных технологий
06.011	Поддержание эффективной работы баз данных, обеспечивающих функционирование информационных систем в организации	Администратор баз данных
06.012	Предпринимательская деятельность в области информационных технологий	Менеджер продуктов в области информационных технологий
06.013	Создание информационных ресурсов в сети Интернет и управление ими	Специалист по информационным ресурсам
06.014	Информационные технологии в экономике и государственном управлении	Менеджер по информационным технологиям
06.015	Создание и поддержка информационных систем в экономике	Специалист по информационным системам
06.016	Менеджмент проектов в области информационных технологий	Руководитель проектов в области информационных технологий
06.017	Менеджмент проектов в области информационных технологий	Руководитель разработки программного обеспечения
06.019	Разработка технической документации и методического обеспечения продукции в сфере информационных технологий	Технический писатель
06.022	Проектно-исследовательская деятельность в области информационных технологий	Системный аналитик

Если ни один из стандартов вам не подошел, рекомендую обратиться к европейской рамке компетенций e-CF, где любой требуемый профиль собирается из 36 базовых ИКТ-компетенций, сгруппированных в пять областей деятельности.



- ◆ Вам нужен некто похожий на действующих сотрудников или необходима свежая кровь?
- ◆ Чьи интересы в вашей команде затронет новая должность?
- ◆ Чего вы ожидаете от нового сотрудника в контексте развития проекта?

Стадии развития продуктов (и проектов) удобно оценивать с помощью так называемой VCG-матрицы. На разных стадиях в команде могут потребоваться разные специалисты.

- ◆ На начальной стадии важны бурный энтузиазм и инициатива работников, роли в команде не фиксируются и могут меняться. Как правило, нет возможности принимать на работу узких специалистов. Вам нужна лояльная и мотивированная творческая команда, ориентированная на достижение результата.
- ◆ На стадии роста начинают проявляться амбиции коллектива, четко выделяются и фиксируются роли и специализации, появляется конкуренция за лидерство, кого-то приходится менять. Важно повернуться лицом к клиентам. Вам нужны суровые профессионалы и новые профессиональные специализации.
- ◆ На стадии зрелости важна стабильность, приобретают особую важность инструкции, процедуры и процессы. В приоритете качество для клиентов. Социальные программы компании работают вкупе с дисциплиной и традициями. Вам нужны команда стабильного развития и рациональный педантизм.
- ◆ На стадии насыщения ветераны коллектива настойчиво требуют уважения к себе, а кто-то жаждет новых вершин. Менеджер либо оставляет все как есть, либо добивается качественного скачка проекта. В первом случае наш путь — консерватизм, во втором — вам нужны революционеры.

### 8.3.4. Шаг 4. Учитываем особенности компании

Ценности компании и взгляды сотрудника должны как минимум не противоречить друг другу.

Нельзя не учитывать корпоративную культуру: историю, традиции, нормы, правила, степени влияния в коллективе, практику взаимодействия, особенности непроизводственных процессов и т. п. Существенное влияние имеют стиль руководства, способ управления (проектный, функциональный или иной) и персональные требования топ-менеджеров компании.

Вы должны как минимум ответить на следующие вопросы.

- ◆ Какие требования обязательны при взаимодействии с клиентами?
- ◆ Какие внутренние правила компании существенны в контексте данной должности?
- ◆ Что обязательно или недопустимо в контексте данной должности?

### 8.3.5. Шаг 5. Определяем приоритеты

Все, что вы смогли сформулировать на предыдущих шагах, нужно ранжировать по значимости, отбросив несущественное и выделив наиболее важное. Формулировки ваших требований необходимо максимально конкретизировать, а лучше всего привести конкретные примеры идеальных, допустимых и недопустимых вариантов характеристик, составляющих профиль должности.

## 8.4. Паттерн «Оценка кандидата»

Паттерн предназначен для оценки кандидатов, резюме которых у вас уже есть. Алгоритм дальнейших действий менеджера.

1. Оценить резюме и разделить их на три категории: «отказ», «сомневаюсь» и «нравится» соответственно для неподходящих кандидатов, тех, кто теоретически может подойти, и тех, кто вам действительно нравится.
2. Для категории «сомневаюсь» уточнить оценки в удаленной форме и перевести их либо в «отказ», либо в «нравится».
3. Собрать группу интервьюеров и провести очное собеседование с кандидатами из категории «нравится».
4. Сделать кандидату предложение о работе или продолжить поиски.

### 8.4.1. Оценка и отбор кандидатов по резюме

Я рекомендую оценивать резюме исходя из соответствия между кривой профессионального развития кандидата и типовыми этапами развития соответствующих специалистов (например, можно заглянуть в главу «Профиль и квалификация менеджеров»).

Необходимо последовательно рассмотреть все места учебы и работы, занимаемые должности и выполненные проекты. У вас должна сложиться картина непрерывного профессионального роста, начиная с первого выполненного проекта, по трем основным критериям: сложность, самостоятельность и ответственность (см. табл. 8.2). Начальная точка роста может прийтись еще на школьные годы, а среди выпускников вуза нам интересен тот, что может похвастаться не только хорошими оценками в дипломе, но и реальными проектами, выполненными во время учебы.

Важно, чтобы кандидат ранее в своей работе ориентировался на достижение результата, а не на исполнение своей маленькой частной функции. Ищите личный вклад, творческий подход и конкретные результаты работы, а не факты участия в процессе или монотонного исполнения процедур. Очень хорошо, если по резюме вы почувствовали стремление к развитию, амбиции и профессиональную гордость кандидата.

Нежелательны:

- ◆ частая смена мест работы: вы же помните, что «звезды» вообще редко меняют работу, однако работа над проектами, в том числе успешная, предполагает и скачки. При очной встрече с кандидатом выясните, с чем связаны переходы;
- ◆ провалы или перерывы в основной профессиональной деятельности. Программиста, например, вообще сложно заставить прекратить программировать даже на время;
- ◆ застой или длительная работа без роста сложности, самостоятельности и ответственности;
- ◆ загулы или незавершенные проекты, брошенная учеба и т. п.

Кривая развития отличного кандидата должна непрерывно расти, на ней нежелательны длительные периоды застоя и провалы в развитии. Все такие факты должны получить простое и рациональное объяснение или стать блокирующим фактором. Пометьте в резюме, какие детали нужно уточнить на собеседовании.

В отношении оценки текущего уровня кандидата и его профессиональных компетенции наиболее интересны последние 3–5 лет: и места работы, и выполняемая работа. Если вам нужен программист на Python — и точка, тут уж ничего не поделаешь, хотя я категорически не рекомендую вам отсеивать кандидатов только за незнание конкретных инструментов, языков программирования, СУБД, нотаций и т. п. Впрочем, тут решать только вам.

Кроме технологий производства, инструментов проектирования и т. п., вы должны сравнить сферу деятельности, рынки и корпоративную культуру последнего места работы с теми же параметрами своей компании, найти схожие черты и различия. Кардинальные расхождения могут стать блокирующим фактором. Здесь тоже нужно сделать пометки для последующего уточнения.

Оформление резюме и написанный кандидатом текст говорят о многом. Резюме — это маленькая, но важная работа, пример письменного изложения мыслей, по которому можно оценить многие качества кандидата. Еще лучше, если в объявлении о наборе сотрудников вы попросите приложить примеры работ соискателя или результат выполнения какого-либо небольшого задания.

Плохо оформленное резюме, имеющее множество ошибок, визуально нечитаемое и т. п. может прислать кто угодно, но только не талантливый и ответственный специалист (разве что студенты по неопытности могут забыть указать даже свое имя). Письменная речь должна быть полноценной с точки зрения языковых конструкций и понятной вам после первого прочтения. Ищите сильные выражения: «завершил», «добился», «создал» и обращайтесь внимание на слабые: «помогал», «мне поручали» и т. п.

Попробуйте выявить негатив, то, чего избегает соискатель. Сравните, например, словосочетания «дружный коллектив» (желание) и «неконфликтный коллектив» (попытка избежать).

Любые странности в резюме, лишняя информация или, наоборот, скупое описание работы в престижной компании должны вас насторожить.



В одном из резюме на должность «маркетолог» я обнаружил, что кандидат зачем-то указал информацию о своей супруге: «Жена — продюсер ТВ-канала». К тому же он не приложил запрошенные в объявлении о вакансии примеры документов из сферы маркетинга, подготовленных им лично. В целом резюме мне показалось неплохим, поэтому я направил кандидату письмо, чтобы снять все вопросы, а также навел справки.

Мне пришел ответ примерно такого содержания: «Серьезные компании часто проверяют не только соискателя, но и его семью, поэтому и указал место работы жены. Примеры аналитических маркетинговых отчетов не приложил по той причине, что это некорректно по отношению к компании, в которой работаю, — отчеты являются коммерческой тайной».

Отказ представить пример своей работы — это не самый удачный вариант реакции соискателя. Мне точно не был нужен сотрудник, который с ходу находит «вескую» причину, чтобы не выполнять поручение, тем более что обезличивание документа от коммерческой и иной составляющей — дело 1–2 часов. И это, скажем так, весьма слабая аргументация, «почему нет». Одновременно (по другим каналам) выяснился удивительный для меня факт, что этот молодой человек всерьез считает место работы своей супруги одним из своих главных достижений.

Не забывайте, что резюме применимо только для первичной сортировки кандидатов — отсеивания непригодных и отбора тех, с кем вы готовы встретиться немедленно.

## 8.4.2. Проведение собеседования

Менеджер должен научиться искусству оценки кандидата, которого он видит в первый раз, за разумный промежуток времени (как правило, не более одного часа). Довольно парадоксальными могут показаться отзывы опытных руководителей, которые говорят, что для оценки кандидата им достаточно резюме и личной беседы в течение 10 минут. Что характерно, предмет такой беседы совсем не обязательно лежит в профессиональной сфере, а оценка «подходит/не подходит» оказывается довольно достоверной. Такой подход к собеседованию называют интуитивным, однако он всецело основан на личном опыте лидера. Итоговая характеристика обычно формулируется довольно простыми фразами: «работать будет», «супер, берем обязательно», «через год станет одним из лучших», «головастик», «заумный, будет ходить кругами», «себе на уме», «бездельник», «живет своим оригами» и т. п.

Руководитель, как правило, без специальной подготовки оценивает специалистов исходя из требований, которые он предъявляет к себе самому, иначе говоря, судит по себе. Хотя известная поговорка и говорит о порочности такого подхода, паттерн ориентирован на использование личного опыта менеджера и во многом базируется на его интуиции.



«Обед состоялся в тот же день, и новый слуга подавал к столу. Обед был заказан Атосом, а лакей рекомендован Портосом. То был пикардиец, которого славный мушкетер нанял в тот самый день по случаю этого самого обеда; он увидел его на мосту Ла-Турнель, где Планше — так звали слугу — плевал в воду, любуясь разбегавшимися кругами. Портос утверждал, что такое занятие свидетельствует о склонности к созерцанию и рассудительности, и, не наводя о нем дальнейших справок, увел его с собой».

А. Дюма. «Три мушкетера»

Интуитивная оценка основана на оценке опыта, личностных качеств и потенциала кандидата, однако, не имея собственного опыта, не стоит полностью доверять интуиции и особенно тому первому впечатлению, которое произвел на вас кандидат. Я рекомендую собрать группу интервьюеров или проводить собеседования в несколько стадий, только постарайтесь не замучить кандидата, а то сбежит, в четвертый раз услышав одни и те же вопросы. Как правило, если совет интервьюеров работает, каждый из них должен иметь право мотивированного вето.

Попробуем расставить приоритеты и сформулировать более понятные правила.

Первое правило, которого следует придерживаться при отборе кандидатов, — это обязательное выявление потенциала профессионального развития специалиста в условиях вашей компании. Известен подход, применяемый, например, Google и Microsoft, когда кандидатов оценивают с помощью логических головоломок, многие из которых вообще не имеют однозначного решения.

Второе правило — это способность доводить конкретные производственные задачи до конца. Вам же не нужен гений-теоретик, который будет годами готовить математические модели, но от которого вы не дождетесь практических результатов.



Важнейшей оценочной характеристикой действительно лучших специалистов является их творческий и интеллектуальный потенциал развития, который может проявиться в условиях реальной производственной деятельности компании. Такая оценка имеет больший вес, нежели наличие или отсутствие у кандидата требуемого вам набора профессиональных компетенций.

Важно помнить, что людям бывает свойственно подсознательно занижать способности потенциальных конкурентов. Моя рекомендация: не бойтесь брать людей, планка которых может быть выше вашей собственной, иначе будете работать с теми, кого заслуживаете.

Примерный план-конспект собеседования таков.

### 1. Вводная часть.

*Цель:* установить контакт, снять первый стресс у кандидата.

*Продолжительность:* 1–2 минуты.

*Что говорить:* представить всех участников, кратко рассказать о цели встречи и установить контакт (задать абстрактный, ничего не значащий вопрос:

«Как добрались?» и т. п.). Рассказать пару слов о том, кого ищете, и/или о компании — все равно что.

Объявить, что вы сейчас зададите ряд стандартных вопросов, которые задаете всем и всегда.

## 2. Беседа с кандидатом.

*Цель:* оценка кандидата

*Продолжительность:* 30–40 минут.

## 3. Стандартные вопросы.

*Цель:* оценить общий настрой, заинтересованность, мотивы прихода к вам.

*Вопросы*

- ◆ Как узнали о нашей компании? Что знаете о компании?
- ◆ Есть ли причина, по которой хотели работать бы именно у нас?
- ◆ Как думаете, чем вам предстоит заниматься?
- ◆ В какой сфере развивались в последнее время? В какой хотели бы развиваться?
- ◆ Чего ожидаете в смысле карьерного роста в нашей компании?

Задайте уточняющие вопросы по резюме, если вы готовились и делали там пометки.

Задайте 2–3 дополнительных общих вопроса о хобби, увлечениях, месте жительства и т. п., для того чтобы послушать высказывания кандидата на абстрактные темы.

## 4. Беседа об опыте работы.

*Цель:* оценка уровня кандидата (сможет ли работать в вашей в команде), а также глубины осознания профессиональной деятельности.

*Форма:* беседа проходит в форме цепочки вопросов и ответов, углубления в тему и уточнения там, где кандидат отвечает кратко и поверхностно и т. п. Читайте резюме, ищите информацию о последних местах работы и выполненных проектах. Если специалист молодой, говорите о дипломной работе, предметах, оценках.

Найдите интересный самому кандидату проект, а затем выберите второй проект, который интересен уже вам, и можете помучить кандидата вопросами. Обращайте внимание на интерес к проекту и глубину понимания того, о чем говорит кандидат: насколько свободно и уверенно он держится, какие примеры приводит, испытывает ли он гордость за результаты, стремится ли к развитию и т. д.

*Вопросы о проектах:* что за проект, где применяется, какие использовались технологии, в чем сложность и др. Расспросите о роли кандидата в проекте, о том, что он делал самостоятельно, за что отвечал. Детально обсуждайте, попытайтесь выяснить мелочи, заставить кандидата смотреть с разных сторон,

абстрагироваться (обсудите нужды заказчиков, частности наподобие протоколов, форматов и т. д., нефункциональные требования). Выясните, что ему было интересно, развивался ли кандидат профессионально на этом проекте.

*Вопросы о командной работе:* сколько было людей в отделе, кто руководил, как ставились задачи, кто писал документацию, какова роль кандидата, с кем он взаимодействовал, как его контролировали или как он контролировал других, какая применялась система мотивации. Узнайте, как сам кандидат оценивает компанию, систему управления, команду, начальника и т. п.

## 5. Простые профессиональные вопросы.

*Цель:* получить быструю первичную оценку профессиональных компетенций.

*Форма:* выберите вопросы общего характера и профессиональные вопросы, довольно простые для обычного специалиста. Для чистоты эксперимента сначала уточните значение 3–4 аббревиатур из резюме кандидата. Не допускайте характерной ошибки: кандидат не обязан угадывать ваш взгляд, ваше мнение или цитату из учебника, а вполне может сформулировать свой вариант правильного ответа.

*Примеры вопросов-определений:* «Что такое, на ваш взгляд... (подставьте базовый термин из профессиональной сферы, например «маркетинг»)»? «Как вы понимаете...?», «В чем различие между...?».

*Примеры вопросов на проверку элементарных знаний:* «Что такое объектно-ориентированное программирование?», «Нарисуйте на бумаге простую реляционную базу данных. Расскажите о любом методе сортировки», «Для чего применяется UML?», «Для чего применяются индексы в СУБД?».

## 6. Задание на оценку мышления.

*Цель:* оценить мышление кандидата, способность понимать задачу, систематизировать информацию, формулировать свои мысли письменно, убедиться в его способности доводить работу до конца.

*Форма:* дайте кандидату простую аналитическую задачку, не связанную со сферой его профессиональной деятельности. Обязательно предупредите, что можно задавать любые уточняющие вопросы. Попросите оформить результат на бумаге, «чтобы его можно было показать генеральному директору».

Задание должно быть немного скучным. Кандидат обязательно должен правильно вас понять, задавать вопросы, довести работу до конца, исправить работу по вашим замечаниям.

*Примеры заданий:* попросите кандидата представить, что он готовит популярную лекцию для студентов о целях деятельности банков и о том, как они зарабатывают деньги. Он должен подготовить пару схем и конспект лекции — все на одном листе. Изложить сведения следует так, чтобы поняла даже ваша бабушка. Темы могут быть и такие: «Что такое радиосвязь и как она работает», «Что такое кровеносная система человека» и др.

*Оцените процесс решения:* умные ли были вопросы, не упростил ли кандидат задачу, не терял ли время на лишнее обсуждение деталей, не поторопился ли, не пытался ли изложить все устно, а не письменно и т. п.

*Оцените результат:* оформление (заголовок, дата, кто выполнил и т. п.), завершенность изложения, полностью, понятно ли написано, системно ли изложена суть, а не только примеры и т. д.

## 7. Оценка профессиональных компетенций.

*Цель:* оценить на реальном примере, сможет ли кандидат выполнять работу.

*Продолжительность:* 30 минут.

*Форма:* дайте кандидату реальную практическую задачу, которую он должен выполнить не более чем за 30 минут, например: написать программный код, разработать дизайн интерфейса, подготовить план рекламного ролика и т. п.

Для отдельных должностей задание может быть сформулировано в виде кейса (ситуационного задания, например: «Вам поручено сделать ролик о компании, ваши действия?»). Идеально использовать в задании реальную проблематику вашей компании. Кандидат просто обязан задавать уточняющие вопросы. Он не решает логическую головоломку, а работает с кейсом. Сам факт нахождения решения не столь важен, как тот алгоритм поиска, который выстраивает кандидат, однако при этом не должно быть отговорок и ухода от выполнения работы.

В качестве альтернативы можно договориться с кандидатом об 1–2-дневной стажировке в вашей команде.

## 8. Завершение.

*Цель:* определиться с дальнейшими действиями, закончить собеседование.

*Вопросы:* узнать о зарплатных ожиданиях или среднем доходе на прежнем месте работы (не все отвечают, надо настоять). Узнать, когда он готов выйти на работу.

Дайте возможность кандидату задать свои вопросы и договоритесь о дальнейших действиях и контактах. Можно сразу сделать предложение о работе или назначить следующее собеседование, например, у генерального директора.

### 8.4.3. Где и как проводить собеседование

Кандидату должно быть комфортно, вы должны дать ему возможность снять верхнюю одежду, расположиться, спокойно приготовить документы. С определенного момента беседы может случиться так, что вы уже сами начнете рекламировать работу в вашей компании. Будет очень обидно, если отличный кандидат откажется от вашего предложения о трудоустройстве из-за того, что вы беседовали с ним, сидя на подоконнике в коридоре.

### 8.4.4. Об отношении к людям

Собеседование всегда стресс. Кандидат находится в заведомо зависимом положении и вынужден отвечать на любые вопросы, потому важно сделать так, чтобы беседа не напоминала допрос. Будьте внимательны к ответам, не отвлекайтесь от беседы без крайней нужды. Воздержитесь (если это не кандидат на должность кризис-менеджера) от стрессовых методик, психологических тестов и вопросов, на которые сами бы не знали, как ответить. Не стоит ждать возникновения желания работать у вас после подобного собеседования, а особенно этого не любят IT-профессионалы.

### 8.4.5. Не тратьте время зря

Если очевидно, что кандидат вам не подходит, найдите достойный способ прекратить собеседование. Укажите точный срок, когда вы дадите ответ по результатам собеседования, предложите кандидату позвонить, если он не получит ответа за 10 дней.

Если кандидат вам подходит, сразу сделайте ему предложение о работе.

### 8.4.6. Кандидат дал свое согласие

Специалист, который хочет завершить работу или передать дела на старом месте, заслуживает уважения. Всегда устанавливайте разумный срок для завершения текущих задач, никогда не пытайтесь добиться выхода на работу «завтра».

Однако не забывайте, что чем больше вам подходит кандидат, тем выше вероятность того, что он:

- ◆ останется на старом месте;
- ◆ получит альтернативное предложение о работе;
- ◆ откажется без внятного объяснения причин;
- ◆ уволится в первые две недели после выхода на работу в вашу компанию.

Попытайтесь ненавязчиво поддерживать контакт с кандидатом и отслеживать его во время перехода и адаптации в вашей команде.

# 9. Как продать себя как мастера управления проектной средой

---

---

Так же как вы как управленец в IT ищете кадры для формирования своей команды, так и бизнес ищет вас, для того чтобы сформировать свою команду.



---

В этой главе я хочу поделиться с вами уникальной возможностью — прочитать описание практики поиска сотрудника на позицию СЮ (включая руководство разработкой ПО и внедрением) от второго лица компании — первого заместителя генерального директора.

---

*На ресурсе для трудоустройства определены и опубликованы требования к квалификации. Подчеркиванием выделены обязательные:*

- ◆ опыт руководящей работы (не менее трех лет) в должности руководителя/заместителя руководителя производства АСУ;
- ◆ опыт работы в компаниях, занимающихся созданием и внедрением АСУ (не менее двух лет) в должностях: архитектор (программист-разработчик АСУ, бизнес-аналитик, системный аналитик);
- ◆ опыт руководителя проектов (крайне желателен);
- ◆ опыт выстраивания и формализации производственных процессов между подразделениями (аналитика, разработка, тестирование, внедрение, сопровождение);
- ◆ опыт формирования бюджета производства, контроля расходов (желателен);
- ◆ опыт создания мотивационных схем для сотрудников производственных подразделений;
- ◆ опыт организации обучения сотрудников.

*Определены требования к личностным характеристикам (без приоритизации). Подчеркиванием выделены обязательные:*

- ◆ приятная, неотталкивающая внешность;
- ◆ поставленный голос;

- ◆ *развитое логическое мышление;*
- ◆ *инициативность;*
- ◆ *энергичность;*
- ◆ *представительность;*
- ◆ *смелость;*
- ◆ *уверенность в себе;*
- ◆ *тактичность;*
- ◆ *умение четко выражать мысли;*
- ◆ *образованность;*
- ◆ *организаторские способности;*
- ◆ *целеустремленность;*
- ◆ *работоспособность;*
- ◆ *самостоятельность;*
- ◆ *обязательность;*
- ◆ *адекватность самооценки;*
- ◆ *амбициозность;*
- ◆ *самостоятельность;*
- ◆ *чувство юмора.*

*Поиск руководителя такого уровня занял порядка шести месяцев. При этом активный поиск – 3–4 месяца.*

*Было отобрано более 100 кандидатов.*

*Были проведены собеседования примерно с 30 кандидатами.*

*Была одна неудачная попытка выхода отобранного кандидата на работу (с ним растались через неделю).*

*В конце поиска на условиях аутсорсинга был привлечен HR-специалист. Условия: HR-специалист занимается поиском не более двух месяцев. Проводит по три собеседования (минимум) предварительно отобранных (проведено первоначальное собеседование с HR) кандидатов в неделю.*

*Компания предоставляет доступ к необходимым ресурсам для поиска кандидатов с размещением на них вакансии.*

*На что обращали внимание при первоначальном отборе резюме.*

- ◆ *Образование – желательно ведущие технические вузы. Плюс, если вуз окончен с красным дипломом (это говорит о том, что человек реально учился. Прежде технические вузы нельзя было окончить с красным дипломом, реально не вкальывая).*
- ◆ *Плюс (сильный) наличие степени MBA (таких кандидатов было немного).*
- ◆ *Опыт в сфере разработки ПО, опыт в проектной деятельности и др. (в соответствии с требованиями к квалификации).*

- ◆ Как построено резюме. Желательно, чтобы в формулировках видна была масштабность задач, но при этом они должны быть конкретными.

Например, человек с опытом работы топ-менеджером указывает, что одной из его задач было составление ТЗ. Даже если он это и делал, не нужно отмечать это, претендуя на руководящую должность данного уровня. Скорее всего, это говорит об отсутствии масштаба мышления или, возможно, низких управленческих качествах.

- ◆ Фотография. Сразу отклонялись резюме, содержащие снимки кандидатов с детьми, восемь разных видов и т. п. Фото должно быть деловым.
- ◆ Возраст. Необязательное требование, но, учитывая, что кандидат должен обладать определенным опытом работы, вряд ли ему может быть 25 лет. Поэтому рассматривались кандидаты старше 29 лет. Негласно был определен возраст 35–45 лет.
- ◆ Опыт работы на одном месте (желательно 2–5 лет).
- ◆ Настороженно относились к указанию в резюме «консультант», «эксперт в сфере IT». Подразумевалось: это те, кто много говорит, но ничего не делает.

Вопросы, которые задавали на собеседовании.

- ◆ На профессиональные темы, позволяющие понять, каковы познания человека в предметной области, и знание основ управления в IT, управления персоналом, управления финансами (например, «Какие методологии разработки, вы считаете, должны быть использованы при разработке подобных систем и почему?» (давалась характеристика), «Как бы вы построили производственный процесс по разработке и внедрению аналогичного ПО?», «Как должен быть построен процесс планирования и выпуска версий ПО?», «Расскажите о разработанных вами мотивационных схемах», «Каково ваше мнение, кого и как нужно премировать в производственном подразделении?» и многие другие).
- ◆ Дополнительные вопросы, позволяющие понять, каковы профессиональные и личные качества кандидата (например, «Вы пришли в компанию на эту должность. Что конкретно будете делать, чтобы войти в курс дела, и как?»).
- ◆ Вопросы об уровне зарплатных ожиданий.
- ◆ Вопрос о готовности представить рекомендации. В случае положительного решения обязательно запрашивались рекомендации — делали телефонный звонок бывшим работодателям-руководителям (как минимум двум).

В среднем собеседование длилось от одного до двух часов. С отобранными кандидатами часто проводили еще одно собеседование.

Критерии успешного прохождения собеседований и формирование компанией предложения.

- ◆ Располагающие внешность и личные качества.
- ◆ Соответствие кандидата основному перечню требований к личностным характеристикам. На данной позиции, кроме профессиональных знаний, очень важны личностные характеристики. То есть, занимая определенные позиции, невозможно

*быть только лидером в профессиональной сфере, быть технически подкованным – необходимо обладать определенными личностными качествами. Быть личностью.*

- ◆ *Подтвержденные знания и опыт в предметной области.*
- ◆ *Подтвержденные знания и опыт в области управления.*
- ◆ *В целом положительные рекомендации бывших работодателей – как IT-профессионала, управленца, человека самостоятельного, инициативного, ответственного, порядочного.*
- ◆ *Соответствие уровня зарплатных ожиданий возможностям компании.*

Я думаю, что вы, читатели этой книги, достаточно профессиональны для того, чтобы самостоятельно проанализировать эту информацию и понять, что и как стоит отражать в резюме, к каким вопросам быть готовыми на собеседовании, какие личностные качества ценят работодатели.

Также рекомендую поискать в Сети и почитать, как специалисты по поиску кадров интерпретируют ответы соискателей на собеседовании. Иногда это очень интересно, небесспорно, но поможет вам понять другую сторону переговоров, увидеть, что видят они, а значит, скорректировать свою стратегию переговоров. Например, перейдя по ссылке <http://alex-levitas.livejournal.com/547454.html>, вы найдете размышления на тему: «Итак, вы задали соискателю рекомендованный мною вопрос “Какими тремя вещами, которые я могу проверить, вы больше всего гордитесь за последние 3–5 лет?” и получили ответ. Давайте посмотрим, какую информацию вы можете извлечь из ответа».

В заключение я могу посоветовать применительно к поиску работы в дополнение к паттерну «Саморазвитие и личностный бренд» еще один интересный паттерн. Сделайте самопрезентацию в PowerPoint на трех слайдах. На них должна быть представлена вся информация о вас как о личности и профессионале, о вашем опыте и о том, почему именно вы справитесь с той позицией, на которую претендуете.

У меня в свое время на создание такой презентации ушло около пяти дней. Я создал ее довольно быстро, за 2–3 часа. А затем в течение оставшегося времени как минимум через 30 минут – час что-то менял в ней. И таких итераций было две. После того как мне все понравилось, я попросил друзей критически посмотреть на результат. Причем часть друзей были не из мира IT. Их задачей было понять, чем я занимаюсь. Часть друзей – мои коллеги, их задачей было оценить, как я показал свой опыт. Я получил очень интересные ответы, после чего снова значительно все переделал.

А вы? Как быстро вы сможете сделать такую презентацию о себе?

# 10. Карта знаний серии «Путь IT-профессионала»

---

---

Вот мы и подоברались к самому концу этой книги. Теперь давайте оглянемся, где мы...

Если отобразить серию книг «Путь IT-профессионала» в виде UML-диаграммы (рис. 10.1), то основные концептуальные понятия этой книги обозначены полужирным шрифтом.

Как видите, в этой книге мы рассмотрели такие понятия, как «проект», «тип проекта», «ЖЦ проекта» и связи этих понятий с рисками проекта, требованиями к качеству продукта и проектной средой.

Немного коснулись методологии разработки программного обеспечения и показали, что WBS проекта в первую очередь ориентирована на снятие проектных рисков, главные из которых — это риски, связанные с достижением требований к качеству продукта, риски, связанные с жизненным циклом продукта, и риски, связанные с проектной средой.

Именно к этим основным понятиям и применяются паттерны. Ими же эти паттерны оперируют.



---

Я рекомендую внимательно изучить диаграмму и после этого еще раз перечитать, имея ее перед глазами, информацию по каким-то из любившихся вам паттернов. Думаю, что такое упражнение сделает для вас из обычного паттерна этакий 3D-паттерн.

---

И скорее всего, вы увидите что-то, что я упустил в описании и возможностях применения паттерна. И адаптируете эту 3D-модель по-своему. И это здорово. Значит, цель книги достигнута.



# Заключение

---

---

Мы с вами прошли длинный путь и рассмотрели хорошие практики, приемы и конкретные примеры действий на всех уровнях профессии «менеджер ИТ», от начинающего управленца в роли администратора проекта до руководителя разработки ПО.

Мы поделились с вами опытом и постарались показать самое лучшее и успешнее всего применимое на практике из того, что имеем в собственном арсенале.

Эта книга написана в надежде, что она поможет вам не наступать на грабли, на которые наступали мы и наши коллеги, не повторять наших ошибок. И мы будем очень рады, если рекомендации, которые даны в этой книге, окажутся полезными для вас и вы сможете выстроить свою профессиональную карьеру и развитие максимально эффективно.

В этой книге, как мы и собирались, получили развитие идеи и концепции, которым было тесно в формате предыдущего издания. Это вторая книга серии книг «Путь ИТ-профессионала». До встречи на страницах следующей. Ну а пока...

Вы закрываете последнюю страницу книги «Путь ИТ-менеджера. Управление проектной средой и ИТ-проектами».

Спасибо за работу.

Удачи!

И легкого вам пути!

# Глоссарий

---

---

IT	Информационные технологии	Широкий класс дисциплин и областей деятельности, относящихся к технологиям управления и обработки данных, а также создания данных, в том числе с применением вычислительной техники. <a href="http://ru.wikipedia.org/wiki/Информационные_технологии">http://ru.wikipedia.org/wiki/Информационные_технологии</a>
ИС	Информационная система	Система хранения, обработки и передачи информации, представленной в определенной форме. В широком смысле информационная система есть совокупность технического, программного и организационного обеспечения, а также персонала, предназначенная для того, чтобы своевременно обеспечивать надлежащих людей надлежащей информацией. <a href="http://ru.wikipedia.org/wiki/Информационная_система">http://ru.wikipedia.org/wiki/Информационная_система</a>
ООП	Объектно-ориентированное проектирование	Подход к проектированию информационных систем на основании определения объектов информационных систем, которые являются в определенном смысле проекциями объектов реального мира/предметной области, для автоматизации которой разрабатывается информационная система. ООП — это часть объектно-ориентированной методологии, которая предоставляет программистам возможность оперировать понятием «объект», а не понятием «процедура» при разработке кода. Объекты содержат инкапсулированные данные и процедуры, сгруппированные вместе, отображая таким образом сущность объекта. Интерфейс объекта описывает взаимодействие с объектом, то, как он определен. Программа, полученная при реализации объектно-ориентированного исходного кода, описывает взаимодействие этих объектов. <a href="http://ru.wikipedia.org/wiki/Объектно-ориентированное_проектирование">http://ru.wikipedia.org/wiki/Объектно-ориентированное_проектирование</a>
ТЗ	Техническое задание	Исходный документ для проектирования сооружения или промышленного комплекса, конструирования технического устройства (прибора, машины, системы управления и т. д.), разработки информационных систем, стандартов либо проведения научно-исследовательских работ (НИР). ТЗ содержит основные технические требования, предъявляемые к сооружению, изделию или услуге, и исходные данные для разработки; в ТЗ указываются назначение объекта, область его применения, стадии разработки конструкторской (проектной, технологической, программной и т. п.)

		документации, ее состав, сроки исполнения и т. д., а также особые требования, обусловленные спецификой самого объекта либо условиями его эксплуатации. Как правило, ТЗ составляют на основе анализа результатов предварительных исследований, расчетов и моделирования. <a href="http://ru.wikipedia.org/wiki/Техническое_задание">http://ru.wikipedia.org/wiki/Техническое_задание</a>
ПУТ	План управления требованиями	Описание подходов к разработке требований и управлению ими, которые будут применяться в конкретном проекте; какие требования будут создаваться, кем и когда, как они будут связаны между собой и с другими проектными артефактами; как будет производиться управление состояниями требований, какие полномочия в управлении у каких ролей и т. д. ПУТ входит в состав плана управления проектом (ПУП) и должен соответствовать плану управления документами (ПУД) проекта
ПУП	План управления проектом	Основной документ для управления проектом, описывает подходы, которые будут применяться для управления конкретным проектом. Включает в себя несколько планов управления для разных областей: требований, коммуникаций, рисков, качества и т. д.
ПУД	План управления документами	Описывает, какие проектные артефакты на каком этапе проекта в каком состоянии должны находиться, какие роли являются согласующими лицами для каждого артефакта, какие роли входят в команду утверждения, какая роль является ответственной за артефакт, даты готовности артефактов и т. д.
БД	База данных	Совокупность данных, хранимых в соответствии со схемой данных, манипулируют которыми в соответствии с правилами средств моделирования данных. <a href="http://ru.wikipedia.org/wiki/База_данных">http://ru.wikipedia.org/wiki/База_данных</a>
ПО	Программное обеспечение	Совокупность программ, процедур и правил, а также документации, относящихся к функционированию системы обработки данных (СТ ИСО 2382/1–84). <a href="http://ru.wikipedia.org/wiki/Программное_обеспечение">http://ru.wikipedia.org/wiki/Программное_обеспечение</a>
ОС	Операционная система	Комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами, а с другой — предназначены для управления устройствами, вычислительными процессами, эффективного распределения вычислительных ресурсов между вычислительными процессами и организации надежных вычислений. Это определение применимо к большинству современных ОС общего назначения. <a href="http://ru.wikipedia.org/wiki/Операционная_система">http://ru.wikipedia.org/wiki/Операционная_система</a>
	Спонсор проекта	Сотрудник (как правило, руководитель высшего звена) организации, реализующей проект, который курирует проект со стороны организации — владельца проекта, обеспечивает общий контроль и поддержку проекта (финансовые, материальные, человеческие и другие ресурсы). Спонсор (куратор) проекта отвечает за достижение проектом конечных целей и реализацию выгод для организации. Спонсор проекта несет ответственность перед генеральным директором/президентом или управляющим советом.

## (Продолжение)

		Спонсор проекта назначает менеджера проекта и обеспечивает ему необходимую поддержку. <a href="http://www.pmpractice.ru/knowledgebase/management/keypoints/participants/">http://www.pmpractice.ru/knowledgebase/management/keypoints/participants/</a>
ЗЛ	Заинтересованное лицо	Человек или организация, которые могут повлиять или на которых может повлиять результат определенных действий (в контексте книги — разработка ИС, процесс выполнения проекта, результат выполнения проекта) в целом
ЖЦ	Жизненный цикл	В книге под ЖЦ понимается ЖЦ ИС. Жизненный цикл информационной системы — период времени, который начинается с момента принятия решения о необходимости создания информационной системы и заканчивается в момент ее полного вывода из эксплуатации. <a href="http://ru.wikipedia.org/wiki/Жизненный_цикл">http://ru.wikipedia.org/wiki/Жизненный_цикл</a>
МП	Менеджер проекта	Специалист в области управления проектами, который несет ответственность за планирование, подготовку и исполнение конкретного проекта. Обычно такая должность и деятельность имеют место в отраслях, склонных к проектным работам, например: IT, строительство и др. <a href="http://ru.wikipedia.org/wiki/Менеджер_проектов">http://ru.wikipedia.org/wiki/Менеджер_проектов</a>
ООАП	Объектно-ориентированный анализ и проектирование	См. ООП
МПр	Менеджер продукта	Менеджер продукта — это специалист в области управления продуктами, который несет ответственность за планирование, подготовку и развитие конкретного продукта. В IT менеджер продукта довольно часто отвечает не за распространение и продажи продукта, а скорее за стратегию его развития, находясь на стыке коммуникаций с менеджерами по продажам/обслуживанию клиентов, с одной стороны, и менеджерами проектов по разработке продукта — с другой
	Проект требований	Часть общей технологической инфраструктуры проекта, предназначенная для хранения требований и управления ими в соответствии с утвержденным планом управления требованиями. См. ПУТ
	Спецификация требований	Спецификация требований программного обеспечения (Software Requirements Specification) — законченное описание поведения системы, которую требуется разработать. Включает ряд пользовательских сценариев (use cases), которые описывают все варианты взаимодействия между пользователями и программным обеспечением. Пользовательские сценарии являются средством представления функциональных требований. В дополнение к пользовательским сценариям спецификация содержит также нефункциональные требования (такие как требования производительности, стандарты качества или проектные ограничения), которые налагают ограничения на дизайн или реализацию. <a href="http://ru.wikipedia.org/wiki/Спецификация_программного_обеспечения">http://ru.wikipedia.org/wiki/Спецификация_программного_обеспечения</a>

		В более широком смысле спецификация требований – документ, в котором содержатся требования или модели, который используется в процессе разработки требований и управления ими в проекте
	Драфт документа	Версия документа, нуждающаяся в ревью (рецензировании), согласовании или утверждении
	Статус-митинг	Совещание, на котором проектная команда определяет состояние работ проекта, анализирует проблемы и пути их решения, выявляет новые риски и пути их минимизации и предотвращения
АИС	Автоматизированная информационная система	Совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации. АИС являются, с одной стороны, разновидностью информационных систем (ИС), с другой – автоматизированных систем (АС), вследствие чего их часто называют ИС или АС
СУ	Система управления	Совокупность процедур, рабочих инструкций, методологий, инструментов, АИС и навыков сотрудников компании, работающих как единое целое и позволяющих за счет синергетического эффекта от их совместного использования эффективно выполнять производственную деятельность в какой-либо области знаний (дисциплине). Проще говоря, СУ – это совокупность организационной структуры, методик, процессов и ресурсов, необходимых для достижения целей управления
СМК	Система менеджмента качества	Система управления качеством производимой продукции в какой-либо организации. Это задокументированный «образ» предприятия как организма, то есть саморегулирующегося механизма, приспособленного к жизни в конкретной экономической среде. <a href="http://ru.wikipedia.org/wiki/Система_менеджмента_качества">http://ru.wikipedia.org/wiki/Система_менеджмента_качества</a>
СУПр	Система управления производством	СУ производства компании объединяет в себе все остальные СУ по областям деятельности компании и является составной частью системы менеджмента качества компании
СУАР	Система управления архитектурой	СУ, позволяющая эффективно выполнять производственную деятельность в области проектирования и разработки архитектуры
СУТ	Система управления требованиями	СУ, позволяющая эффективно выполнять производственную деятельность в области разработки требований и управления ими
СУП	Система управления проектами	СУ, позволяющая эффективно выполнять производственную деятельность в области управления проектами
СККП	Система контроля качества продуктов	СУ, позволяющая эффективно выполнять производственную деятельность в области контроля качества продуктов
СУИК	Система управления изменениями	СУ, позволяющая эффективно выполнять производственную деятельность в области управления изменениями
РИ	Рабочая инструкция	Рабочая инструкция – пошаговая инструкция для одной или нескольких ролей компании с целью выполнения какой-либо деятельности. Чаще всего рабочие инструкции стараются писать для одной роли/должности, объединяя

## (Продолжение)

		несколько РИ в одну процедуру. Целостный процесс, в свою очередь, описывается через совокупность процедур. Все эти описания являются частью СМК компании
ERP	Enterprise Resource Planning System – система планирования ресурсов предприятия	ERP-система (англ. Enterprise Resource Planning System – система планирования ресурсов предприятия) – это интегрированная система на базе ИТ для управления внутренними и внешними ресурсами предприятия (значимые физические активы, финансовые, материально-технические и человеческие ресурсы). Цель системы – содействие движению потоков информации между всеми хозяйственными подразделениями (бизнес-функциями) внутри предприятия и информационная поддержка связей с другими предприятиями. Построенная, как правило, на централизованной базе данных, ERP-система формирует стандартизованное единое информационное пространство предприятия. <a href="http://ru.wikipedia.org/wiki/ERP">http://ru.wikipedia.org/wiki/ERP</a>
CRM	Customer Relationship Management-System – система управления взаимодействием с клиентами	Корпоративная информационная система, предназначенная для автоматизации CRM-стратегии компании, в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путем сохранения информации о клиентах (контрагентах) и истории взаимоотношений с ними, установления и улучшения бизнес-процедур и последующего анализа результатов. Под термином «CRM-система» понимается программный продукт (ПО), направленный на реализацию концепции CRM. <a href="http://ru.wikipedia.org/wiki/CRM">http://ru.wikipedia.org/wiki/CRM</a>
Enterprise Architecture	Архитектура предприятия/ корпоративная архитектура	Высокоуровневая архитектура всего предприятия, покрывающая бизнес-потребности ИТ-способностями. Корпоративная архитектура фокусируется на определении потоков и бизнес-процессов, действий, функций, информации, данных и технологий предприятия, и на вызовах, стоящих перед ИТ, необходимых для того, чтобы эффективно применить технологию в ответ на изменение бизнес-потребностей
Business Architecture	Бизнес-архитектура	Описывает все бизнес-процессы, бизнес-акторы, бизнес-сущности и бизнес-правила с точки зрения бизнеса. Бизнес-архитектура не зависит от применяемых в разработке технологий
Information Architecture	Информационная архитектура	Определяет структуры данных и описывает все потоки данных, которые используются для поддержки бизнес-архитектуры. Такие операции, как идентификация, систематизация, категоризация, хранение данных, относятся к информационной архитектуре. Может представляться в виде модели данных
Solution (System/ Application) Architecture	Архитектура решения	Архитектура программного обеспечения, которое реализует функции бизнес-архитектуры
Technology Architecture	Технологическая архитектура	Описывает архитектуру ИТ-окружения, которое используется для поддержки информационной архитектуры и архитектуры решения

System Architecture	Системная архитектура	Это представление системы, которое показывает реализацию функциональных возможностей системы аппаратными средствами и компонентами программного обеспечения, устанавливает связь архитектуры программного обеспечения и архитектуры аппаратных средств, а также регламентирует взаимодействие пользователя с этими компонентами. Существуют и другие определения системной архитектуры, например: ряд взаимосвязанных шаблонов (паттернов), которые структурируют модули и данные и обеспечивают требуемое поведение системы (См. Data Architecture). Системная архитектура является составной частью архитектуры решения
Software Architecture	Архитектура программного обеспечения	Является составной частью системной архитектуры. Описывает организацию системы с точки зрения программных компонентов, из которых она состоит, и связи между компонентами
Data Architecture	Архитектура данных	Является составной частью системной архитектуры. Описывает структуры данных и логические связи между данными
Hierarchical model	Иерархическая модель	Объекты этой модели организованы через отношения «родитель — ребенок» и составляют иерархию объектов
Network model	Сетевая модель	Объекты этой модели чаще всего связаны в виде графа. Все связи обычно имеют одинаковые характеристики с разными значениями. Хороший пример — сетевая диаграмма проекта в проектном управлении
Relational model	Реляционная (взаимосвязанная) модель	Наиболее распространенная в настоящее время для проектирования реляционных баз данных модель. Объектами модели являются структуры данных, которые связаны между собой разными типами связей
Object model	Объектная модель	См. ООП
CMMI	Capability Maturity Model Integration — набор моделей (методологий) совершенствования процессов в организациях	CMMI содержит набор рекомендаций в виде практик, реализация которых, по мнению разработчиков модели, позволяет реализовать цели, необходимые для полной реализации определенных областей деятельности. <a href="http://www.sei.cmu.edu/cmmi/">http://www.sei.cmu.edu/cmmi/</a>
CMMI Process Improvement	Улучшение процессов в соответствии с CMMI	См. CMMI. Про метод оценки зрелости процессов по CMMI можно прочитать здесь: <a href="http://en.wikipedia.org/wiki/Standard_CMMI_Appraisal_Method_for_Process_Improvement">http://en.wikipedia.org/wiki/Standard_CMMI_Appraisal_Method_for_Process_Improvement</a>
SWEBoK	Software Engineering Body of Knowledge	Документ, подготавливаемый комитетом Software Engineering Coordinating Committee, в который вовлечено сообщество IEEE Computer Society. Назначение SWEBoK — объединение знаний по инженерии программного обеспечения (разработке программного обеспечения). <a href="http://ru.wikipedia.org/wiki/SWEBoK">http://ru.wikipedia.org/wiki/SWEBoK</a>
Rational Unified Process (RUP)	Унифицированный процесс разработки ПО компании Rational	Унифицированный процесс разработки ПО компании Rational с однозначно выраженными рекомендациями по разработке ПО, включающими в себя перечень всех необходимых деятельности, выполняемых проектными ролями на каждой итерации, и шаблонов артефактов. <a href="http://ru.wikipedia.org/wiki/RUP">http://ru.wikipedia.org/wiki/RUP</a>

## (Продолжение)

Microsoft Solution Framework (MSF)	Руководство по созданию решений от Microsoft	MSF не является методологией разработки ПО. MSF предлагает подходы, основанные на определенной совокупности принципов, моделей, дисциплин, руководств и методик для проектов различной степени сложности, ориентированных на поставку решений. <a href="http://ru.wikipedia.org/wiki/Microsoft_Solutions_Framework">http://ru.wikipedia.org/wiki/Microsoft_Solutions_Framework</a>
ICONIX	Хотя название процесса пишется прописными буквами, оно не является аббревиатурой по разъяснению ICONIX Software	Процесс разработки программного обеспечения с использованием ограниченного количества UML моделей и диаграмм, состоящий из небольших шагов, ведущих к цели. <a href="http://iconixprocess.com/iconix-process/">http://iconixprocess.com/iconix-process/</a>
Спиральная разработка Barry W. Boehm	Спиральная разработка	Спиральная модель (англ. Spiral model) была разработана в середине 1980-х годов Барри Бозмом. Она основана на классическом цикле Деминга PDCA (plan-do-check-act). При использовании этой модели ПО создается в несколько итераций (витков спирали) методом прототипирования. <a href="http://en.wikipedia.org/wiki/Barry_Boehm">http://en.wikipedia.org/wiki/Barry_Boehm</a>
	Agile	Гибкая методология разработки программного обеспечения <a href="http://en.wikipedia.org/wiki/Agile_software_development">http://en.wikipedia.org/wiki/Agile_software_development</a>
Extreme Programming (XP)	Экстремальное программирование	Одна из гибких методологий разработки программного обеспечения. Авторы методологии — Кент Бек, Уорд Каннингем, Мартин Фаулер и др. <a href="http://ru.wikipedia.org/wiki/Extreme_programming">http://ru.wikipedia.org/wiki/Extreme_programming</a>
Robustness diagram	Диаграмма устойчивости	Или диаграмма пригодности. Используется в ICONIX. <a href="http://iconixprocess.com/iconix-process/analysis-and-preliminary-design/robustness-analysis/">http://iconixprocess.com/iconix-process/analysis-and-preliminary-design/robustness-analysis/</a>
Sequence diagram	Диаграмма последовательности	Показывает взаимодействие объектов во времени. Используется в языке UML. <a href="http://en.wikipedia.org/wiki/Sequence_diagram">http://en.wikipedia.org/wiki/Sequence_diagram</a>
Analysis model	Аналитическая модель	Объектная модель, описывающая реализацию вариантов использования. Создается на фазе уточнения (Elaboration) в RUP
	Rational Requisite PRO	Инструмент для разработки требований и управления ими. <a href="http://www-01.ibm.com/software/awdtools/reqpro/">http://www-01.ibm.com/software/awdtools/reqpro/</a>
Requirement	Требование	1. Условие или способность, необходимые пользователю, чтобы решить проблему или достигнуть цели. 2. Условие или способность, которыми должны обладать система или компонент системы для удовлетворения контракта, стандарта, спецификации или другого формально установленного документа. 3. Документированное представление условия или способности, показанных в определении 1 или 2. © (IEEE Std 610.12-1990)
Unified Modeling Language (UML)	Унифицированный язык моделирования	Язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации,

		проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация. <a href="http://ru.wikipedia.org/wiki/UML">http://ru.wikipedia.org/wiki/UML</a>
Use case model	Модель вариантов использования системы/модель прецедентов системы	Описывает поведение разрабатываемой системы (как пользователи взаимодействуют с системой, и что система делает в ответ на эти взаимодействия). Эта модель описывает функциональные требования в терминах вариантов использования (use cases). Используется в RUP
Domain model	Модель предметной области	Модель, которая описывает основные сущности предметной области, их взаимосвязи друг с другом и атрибуты сущностей. Используется для выявления, классификации и формализации сведений обо всех аспектах предметной области, определяющих свойства разрабатываемой системы. <a href="http://en.wikipedia.org/wiki/Domain_model">http://en.wikipedia.org/wiki/Domain_model</a>
Business Vision	Бизнес-видение	Концепция создания и развития продукта. См. BVISION
Technical Vision	Техническое видение	Концепция системы. См. TVISION
Scrum team	Команда «регби»	Подход впервые описали Хиротака Takeuti и Икудзиро Нонака [1] в статье The New Product Development Game (Гарвардский деловой обзор [2], январь-февраль 1986). Они отметили, что проекты, над которыми работают небольшие кросс-функциональные команды, обычно систематически производят лучшие результаты, и объяснили это как «подход регби». В 1991 году Де Грейс и Шталь в книге «Злые проблемы, справедливые решения» [3] сослались на этот подход, как на Scrum (толкотня; схватка вокруг мяча (в регби)), спортивный термин, приведенный в статье Takeuti и Нонакой
Workflow	Поток работ	Состоит из последовательности взаимосвязанных действий. Используется для моделирования процессов. <a href="http://en.wikipedia.org/wiki/Workflow">http://en.wikipedia.org/wiki/Workflow</a>
Business modeling	Бизнес-моделирование	Совокупность методов, практик и шаблонов для моделирования бизнеса
GUI Storyboard	Прототип пользовательского интерфейса	Раскадровка графических форм пользовательского интерфейса
Dynamic model	Динамическая модель	Описывает поведение разрабатываемой системы. Включает в себя модель вариантов использования (См. Use case model), диаграммы устойчивости (См. Robustness diagram), диаграммы последовательности (См. Sequence diagram)
Static model	Статическая модель	Описывает сущности системы. Включает в себя модель предметной области (См. Domain model), диаграмму классов
Test Plan	Тест-план	План тестирования разрабатываемого ПО
	Updated Domain Model	Обновленная модель предметной области
Class Model	Модель классов	Статическая модель, элементами которой являются классы, типы, структуры данных и взаимосвязи между ними. Используется в RUP

## (Продолжение)

Unit Testing	Модульное тестирование	Процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. <a href="http://en.wikipedia.org/wiki/Unit_testing">http://en.wikipedia.org/wiki/Unit_testing</a>
WBS	Work Breakdown Structure	Иерархическая структура работ проекта
Use case	Вариант использования системы/ прецедент	Составная часть моделирования ИС – функция системы, которая приносит ощутимый и значимый результат актору. См. Вариант использования (UC)
Rational Rose	Имя собственное инструмента	Инструмент для моделирования ИС на языке моделирования UML. <a href="http://www-01.ibm.com/software/awdtools/developer/rose/">http://www-01.ibm.com/software/awdtools/developer/rose/</a>
DOORS Telelogic	Имя собственное инструмента	Инструмент для разработки и управления требованиями. <a href="http://www-01.ibm.com/software/awdtools/doors/">http://www-01.ibm.com/software/awdtools/doors/</a>
Sparx Enterprise Architect	Имя собственное инструмента	Инструмент для моделирования ИС. <a href="http://www.sparxsystems.com/products/ea/index.html">http://www.sparxsystems.com/products/ea/index.html</a>
Drug & drop	«Бери и тащи»	Общеиспользуемое обозначение действий мышью по перемещению элементов, например на рабочем столе ОС Windows при удерживаемой левой клавишей мыши
SSL	Secure Sockets Layer – уровень защищенных сокетов	Криптографический протокол, который обеспечивает установление безопасного соединения между клиентом и сервером. SSL изначально разработан компанией Netscape Communications. Впоследствии на основании протокола SSL 3.0 был разработан и принят стандарт RFC, получивший имя TLS. Протокол обеспечивает конфиденциальность обмена данными между клиентом и сервером, использующими TCP/IP, причем для шифрования применяется асимметричный алгоритм с открытым ключом. При шифровании с открытым ключом применяются два ключа, причем любой из них может использоваться для шифрования сообщения. Тем самым, если для шифрования использован один ключ, то для расшифровки нужен другой ключ. В такой ситуации можно получать защищенные сообщения, публикуя открытый ключ и храня в тайне секретный ключ. <a href="http://ru.wikipedia.org/wiki/SSL">http://ru.wikipedia.org/wiki/SSL</a>
	Supplementary requirements	Дополнительные требования
Microsoft Visio	Имя собственное инструмента	Продукт Microsoft для построения визуальных схем и диаграмм
Rational SoDA for Word	Имя собственное инструмента	Инструмент для создания отчетов из инструмента управления требованиями Rational Requisite Pro
Use case realization	Реализация варианта использования системы/ прецедента	Описывает реализацию варианта использования в терминах объектов. Применяется в RUP
API	Application programming interface – интерфейс прикладного программирования	Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах. Применяется программистами для написания всевозможных приложений

IDEF0	DEF – методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования сложных систем (Icam DEFinition – IDEF, или другой вариант – Integrated DEFinition)	Методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов
Диаграммы eEPC	Extended Event Process Chain	Составляющая моделирования ИС
Target Release	Целевой релиз	Атрибут требования. Номер версии продукта, где требование будет реализовано
Phase	Фаза	Атрибут требования. Фаза проекта, в которой реализуется требование
Cause	Причина	Атрибут требования. Текстовое поле для указания запросов на изменение (Enhancement Requests), номеров запросов на разработку функциональности (Feature Request), номеров протоколов совещаний или просто для указания комментариев по разъяснению перевода требования из одного состояния в другое
Enhancement Requests	Запрос на изменение	Запрос на изменение
OMT	Object Modeling Technique	В 1994 году Гради Буч и Джеймс Рамбо, работавшие в компании Rational Software, объединили свои усилия для создания нового языка объектно-ориентированного моделирования. За основу языка были взяты разработанные ими методы моделирования (Object-Modeling Technique, OMT). OMT был ориентирован на анализ, а Booch – на проектирование программных систем. В октябре 1995 года была выпущена предварительная версия 0.8 унифицированного метода (англ. Unified Method). Осенью 1995 года к компании Rational присоединился Айвар Якобсон, автор метода Object-Oriented Software Engineering – OOSE. OOSE обеспечивал превосходные возможности для спецификации бизнес-процессов и анализа требований при помощи сценариев использования. OOSE также был интегрирован в унифицированный метод. <a href="http://ru.wikipedia.org/wiki/Unified_Modeling_Language">http://ru.wikipedia.org/wiki/Unified_Modeling_Language</a>
OOSE	Object-Oriented Software Engineering	
CCB	Change Control Board	Комиссия по управлению изменениями

# Словарь терминов

---

---

Computation-independent model	Вычислительно-независимая модель
Platform-independent model	Платформонезависимая модель
Platform-specific model	Платформоориентированная модель
More abstract	Более абстрактная
More specific	Более специфичная
MDA	Model-Driven Architecture – моделирование, управляемое архитектурой
VOPC	View Of Participated Classes – представление классов, участвующих в реализации поведения
WCF	Windows Communication Foundation
SDLC	Software Development Life Circle – жизненный цикл разработки программного обеспечения
Requirement Management Planning	Планирование управления требованиями. <i>См. ПУТ</i>
Business Analysis	Анализ текущего состояния и описание бизнеса объекта автоматизации
Conception Design	Концепция автоматизации
Enhancement Requests	Запрос на расширение функциональности
Business Use Case Model	Модель бизнес-кейсов
Use Case Modeling (UCM)	Описание желаемого поведения системы
Use Case Analysis (UCA)	Анализ ожиданий и поведения системы
System Analysis	Системный анализ
Functional Design	Иерархия функциональных требований
Legacy SRS Analysis and Requirements Reengineering	Восстановление требований из унаследованных спецификаций требований
Legacy Requirements Reverse Engineering	Восстановление наследованных требований
Additional Business Analysis	Дополнительный бизнес-анализ
Current Application Reverse Engineering	Восстановление требований из текущей реализации системы
Preliminary Analysis (PA)	Предварительный анализ
Business Analysis (BA)	Анализ текущего состояния и описание бизнеса объекта автоматизации
Functional Design (FD)	Функциональный дизайн
System Design (SD)	Системный дизайн
Unit Testing (Unit_T)	Юнит-тестирование

Functional Testing (FT)	Функциональное тестирование
Integrational Testing (Integr_T)	Интеграционное тестирование
Statement of Work (SOW)	Заявление о нужных работах
Stakeholder (STKR)	Запросы заинтересованных лиц
Project Management Plan (PMP)	План управления проектом. См. ПУП
Conceptual Design (CD)	Дизайн концепции
Architectural Design (AD)	Архитектурный дизайн
Development (DEV)	Разработка
Stress Testing (Stress_T)	Нагрузочное (стрессовое) тестирование
Customer Acceptance Test (CAT)	Приемочные испытания
Test Management (TM)	Управление тестированием
Risk Management (Risk Mng)	Управление рисками
Quality Atributes (Qatr)	Атрибуты качества
Conception Creation	Создание концепции
Architecture Design	Проектирование архитектуры
Functionality Realization	Реализация функциональности
Production	Производство
Features	Функциональность
Spiral Architecture Driven Development	Спиральная архитектурно ориентированная разработка ПО
Initiation	Инициация
Happy Path Definition	Определение удачных сценариев выполнения прецедентов
Draft	Версия
Release	Релиз
TD	Test Design – проектирование и планирование тестирования
SA	Системный анализ
Conceptual Model	Концептуальная модель
Product Group Manager	Менеджер группы продуктов. См. Менеджер продуктов
Lead Architect	Главный архитектор
Balanced Score Card (BSC)	Система сбалансированных показателей
Key Goals Indicators (KGI)	Ключевые показатели достижения целей
Client Service	Клиент-сервисная технология
EBITDA	Сумма операционной прибыли
Critical Success Factors (CSF)	Критические факторы успеха выполнения процессов
Project Management Office (PMO)	Офис управления проектами
Lessons Learned	Выученные уроки
Next Release SOW	Задание на следующий релиз
Deployment Requirements	Требования к развертыванию
SOW feasibility analysis and estimations effort	Анализ реализуемости задания на следующий релиз и оценка трудозатрат
Roles and Responsibilities Analysis	Анализ ролей и ответственностей
Business Processes	Бизнес-процессы
Functional Roles	Функциональные роли

(Продолжение)

Decisions	Решения
Functions	Функции
Activities	Деятельности
Responsible	Ответственный
Accountable	Орган/лицо отчетности
Inform	Орган/лицо для информирования
Consulting	Консультант – владеет информацией/ожиданиями/возможностями, необходимыми для выполнения и завершения работы
Specific Goal (SG)	Специфическая цель
General Goal (GG)	Общая цель
General Practice (GP)	Общая практика
Specific Practice (SP)	Специфическая практика
Key Process Indicators (KPI)	Ключевые показатели хода выполнения процессов
Steering Committee	Исполнительный комитет
Work Authorization System (WAS)	Система авторизации работ
Project Management System (PMS)	Система управления проектами
Application Life Time System (ALM)	Система управления жизненным циклом производства продукта
Engineering Process Areas	Области процесса разработки
Technical Solutions (TS)	Область технических решений
Project Iteration (PI)	Интеграционные процессы проекта
Requirement Development (RD)	Разработка требований
Requirement Management (REQM)	Управление требованиями
Decision Analysis and Resolutions (DAR)	Методика анализа и принятия решений
Verification (VER)	Верификация
Risk Management (RSKM)	Управление рисками
Project Planning (PP)	Планирование проекта
Project Monitoring and Control (PMC)	Мониторинг и контроль проекта
Validation (VAL)	Приемка продукта
Project Management Process Areas	Области процесса управления проектом
Basic Project Management Process Area	Взаимодействие процессных областей
Advanced Project Management Process Areas	Области расширенного процесса управления проектом
Integrated Product Management (IPM)	Интегрированное управление продуктом
Project Management Body of Knowledge (PMBOK)	Свод знаний по управлению проектом
Peer Review	Одноранговое ревью
Steering Committee	Управляющий комитет
SEPG	Группа совершенствования процессов инженерии ПО
Institution	Институция – процесс формального описания процессов в виде процедур, рабочих инструкций и регламентов
Workflow	Поток работ

Customer Relationship Management System	Система управления взаимоотношениями с клиентами
Enterprise Management System	Система управления предприятием
Capability	Способность, возможность
Backbone	Каркас (кость скелета)
Story	История
System	Система
Object Constraint Management	Управление ограничениями на объекты и связи объектов
Employee	Работник
Work For	Работает на
Company	Компания
Technology	Технология

# Дополнительные материалы

---

---

На сайте книги вы найдете дополнительные материалы, перечисленные в порядке упоминания в ней:

- ◆ процессы РМВОК: [http://bit.ly/pmway\\_pmbok](http://bit.ly/pmway_pmbok);
- ◆ жизненный цикл разработки: [http://bit.ly/pmway\\_sdlc](http://bit.ly/pmway_sdlc);
- ◆ пример базовых правил работы подразделения: [http://bit.ly/pmway\\_baserules](http://bit.ly/pmway_baserules);
- ◆ проект «Ковчег» как иллюстрация экстремального проекта: [http://bit.ly/pmway\\_projectark](http://bit.ly/pmway_projectark);
- ◆ описание подхода «экстремальное управление проектами»: [http://bit.ly/pmway\\_extprm](http://bit.ly/pmway_extprm);
- ◆ пример отчета о статусе проекта для всех ЗЛ: [http://bit.ly/pmway\\_statereport](http://bit.ly/pmway_statereport);
- ◆ управление неопределенностью: [http://bit.ly/pmway\\_mnguncert](http://bit.ly/pmway_mnguncert);
- ◆ шаблон для подготовки к переговорам: [http://bit.ly/pmway\\_negotiation](http://bit.ly/pmway_negotiation);
- ◆ рекомендации по проведению презентации: [http://bit.ly/pmway\\_presentation](http://bit.ly/pmway_presentation);
- ◆ рекомендации по выбору жизненного цикла: [http://bit.ly/pmway\\_choiceoflc](http://bit.ly/pmway_choiceoflc);
- ◆ пример жизненного цикла производства: [http://bit.ly/pmway\\_lcexample](http://bit.ly/pmway_lcexample);
- ◆ презентация по работе с пакетами требований: [http://bit.ly/pmway\\_reqpackage](http://bit.ly/pmway_reqpackage);
- ◆ пример метрик проекта: [http://bit.ly/pmway\\_projectmetrics](http://bit.ly/pmway_projectmetrics);
- ◆ WIBAS CMMI Browser: [http://bit.ly/pmway\\_cmmi](http://bit.ly/pmway_cmmi) или <http://www.cmmi.de/#el=CMMI-DEV/0/HEAD/folder/Processcollection.00012>;
- ◆ пример устава проекта: [http://bit.ly/pmway\\_projectcharterex](http://bit.ly/pmway_projectcharterex);
- ◆ архитектура СУП в РМВОК: [http://bit.ly/pmway\\_pmspmbok](http://bit.ly/pmway_pmspmbok);
- ◆ пример реестра ошибок: [http://bit.ly/pmway\\_buglist](http://bit.ly/pmway_buglist);
- ◆ пример программы и методики испытаний: [http://bit.ly/pmway\\_aceptancetesting](http://bit.ly/pmway_aceptancetesting);
- ◆ опыт построения QMS: [http://bit.ly/pmway\\_qms](http://bit.ly/pmway_qms);
- ◆ шаблон расчета коэффициента трудового участия: [http://bit.ly/pmway\\_ild](http://bit.ly/pmway_ild).

Также напоминаем, что на сайте книги «Путь аналитика», <http://bp4you.ru/index.php/materialy-knigi>, серии «Путь IT-профессионала» вы найдете дополнительные материалы:

- ◆ шаблон «Протокол совещания»;
- ◆ шаблон «Запросы заинтересованного лица»;
- ◆ шаблон «Концепция системы»;
- ◆ специальный скрипт для Visio, который позволяет специфицировать GUI-элементы непосредственно в Visio, сохраняя комментарий/требование в базе данных MS Access;
- ◆ конспект «UML и Rational Unified Process»;
- ◆ презентация «Возможности Rational SoDA и полезные шаблоны»;
- ◆ шаблоны SoDA для генерации спецификаций требований;
- ◆ шаблон ПУТ для водопадного жизненного цикла разработки;
- ◆ рекомендации по проведению презентаций;
- ◆ выбор ЖЦ проектов в зависимости от характеристик системы и требований к качеству продукта;
- ◆ шаблон ПУД.

# Список использованной литературы

---

---

1. *Перерва А. Д.* Путь аналитика. Практическое руководство IT-специалиста. — М.: Питер, 2011.
2. *Чернов А. В.* Управление портфелями проектов (IBM Best Practices). Курс МВА. — М.: ВШЭ ГУУ, 2007.
3. *Ebert C.* Understanding the Product Life Cycle: Four Key Requirements Engineering Techniques. IEEE Software. — 2006. Vol. 23. Issue 3. — P. 19–25.
4. A Guide to the Project Management Body of Knowledge (PMBOK® Guide): <http://www.pmi.org/>.
5. *Minoli D.* Enterprise architecture A to Z: frameworks, business process modeling, SOA, and infrastructure technology. — Auerbach Publications Taylor & Francis Group, 2008.
6. CHAOS Chronicles v. 3.0, The Standish Group Int'l, 2003: <http://www.standishgroup.com/chaos/toc.php>.
7. *De Carlo D.* eXtreme project management: using leadership, principles, and tools to deliver value in the face of volatility. Published by Jossey-Bass. — A Wiley Imprint, 2004.
8. *Карли Ф.* Трудный выбор. Уроки бескомпромиссного лидерства в сложных ситуациях от экс-главы Hewlett-Packard / Пер. с англ. Э. В. Кондуковой. — М.: Эксмо, 2009.
9. *Минцберг Г.* Структура в кулаке: создание эффективной организации. — СПб.: Питер, 2001.
10. *Листер Т., ДеМарко Т.* Peopleware. — М.: Символ-плюс, 2011.
11. *Херси П., Бланиар К.* Менеджмент организационного поведения (Management of Organizational Behavior). 8-е изд. — Prentice Hall, 2000.
12. *Шварц Д.* Искусство мыслить масштабно. — М.: Поппури, 2008.

13. Вебер Б. Энциклопедия относительного и абсолютного знания. — М.: Рипол-классик, 2007.
14. SEI. Capability Maturity Model for Software. Version 1.1, Document No. CMU/SEI-93-TR-25, ESC-TR-93-178. — Pittsburgh, PA: Carnegie-Mellon University Software Engineering Institute, 1993.
15. Chrissis M., Konrad M., Shrum C. CMMI: Guidelines for Process Integration and Product Improvement. — Addison-Wesley, 2003.
16. Katzenbach J. R., Smith D. K. The wisdom of team. — 1993.

# Использованные интернет-ресурсы

---

---

- ◆ [http://en.wikipedia.org/wiki/ISO\\_21500](http://en.wikipedia.org/wiki/ISO_21500).
- ◆ <http://ru.wikipedia.org/wiki/Проект>.
- ◆ [http://ru.wikipedia.org/wiki/Портфель\\_проектов](http://ru.wikipedia.org/wiki/Портфель_проектов).
- ◆ [https://ru.wikipedia.org/wiki/Программа\\_проектов](https://ru.wikipedia.org/wiki/Программа_проектов).
- ◆ [http://www.iteam.ru/publications/project/section\\_39/article\\_2209/](http://www.iteam.ru/publications/project/section_39/article_2209/).
- ◆ <http://www.sovnet.ru/specialists/ipma-certification/>.
- ◆ [http://www.cecsi.ru/coach/leadership\\_attributes.html](http://www.cecsi.ru/coach/leadership_attributes.html).
- ◆ <http://www.pmi.org/>.
- ◆ <http://www.standishgroup.com/chaos/toc.php>.
- ◆ [http://www.agileproductdesign.com/downloads/patton\\_user\\_story\\_mapping.ppt](http://www.agileproductdesign.com/downloads/patton_user_story_mapping.ppt).
- ◆ <http://www.youtube.com/watch?v=XWyBkmvNGrk>.
- ◆ <http://alex-levitas.livejournal.com/547454.html>.

# Послесловие

---

---

Современный рынок информационных технологий, требующий немедленных и точных ответов о результатах, качестве и сроках выполнения сложных задач, влияющих на судьбу как отдельных продуктов, так и целых транснациональных корпораций, огромную роль отводит Его Величеству проектному управлению. Будь то разработка небольшого мобильного приложения или системы управления бюджетным процессом целого региона — это комплексный проект, реализация которого требует соответствующих знаний, умений и навыков. Причем знания, умения и навыки должны концентрироваться не только в области непосредственной разработки программного обеспечения, но и в других немаловажных сферах, определяющих конечный эффект от реализации проекта. Как рассчитать сроки реализации проекта? Как подобрать нужных людей, где их найти, сколько и каких специалистов потребуется? Как управлять качеством? Как привести проектную команду к успеху?



Перед вами лежит книга, которая поможет выстроить систему ориентиров в управлении IT-проектами. Компания «Кейсистемс» выражает надежду, что с помощью этого издания будет реализован не один успешный IT-проект, особенно в такой важной и социально значимой сфере, как управление бюджетным процессом и общественными финансами.

*С уважением,  
Генеральный директор компании «Кейсистемс»  
Алексей Матросов*

*А. Перерва, С. Еранов, В. Иванова, С. Сергеев*

**Путь IT-менеджера.  
Управление проектной средой и IT-проектами**

Заведующая редакцией  
Ведущий редактор  
Литературный редактор  
Художник  
Корректоры  
Верстка

*И. Сальникова  
Н. Гринчик  
Н. Роцина  
С. Маликова  
Т. Курьянович, Е. Павлович  
А. Барцевич*

ООО «Питер Пресс», 192102, Санкт-Петербург, ул. Андреевская (д. Волкова), 3, литер А, пом. 7Н.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014, 58.11.12 —

Книги печатные профессиональные, технические и научные.

Подписано в печать 24.07.15. Формат 70×100/16. Усл. п. л. 25,800. Тираж 1550. Заказ 0000.

Отпечатано в ОАО «Первая Образцовая типография». Филиал «Чеховский Печатный Двор».  
142300, Московская область, г. Чехов, ул. Полиграфистов, 1.