

А. В. Чекмарев

УПРАВЛЕНИЕ ИТ-ПРОЕКТАМИ И ПРОЦЕССАМИ

УЧЕБНИК ДЛЯ ВУЗОВ

Рекомендовано Учебно-методическим отделом высшего образования в качестве учебника для студентов высших учебных заведений, обучающихся по инженерно-техническим и экономическим направлениям

**Книга доступна на образовательной платформе «Юрайт» urait.ru,
а также в мобильном приложении «Юрайт.Библиотека»**

Москва • Юрайт • 2022

УДК 004+65.012(075.8)

ББК 65.290:32.97я73

Ч-37

Автор:

Чекмарев Анатолий Владимирович — кандидат технических наук, преподаватель Банковского института Национального исследовательского университета «Высшая школа экономики».

Рецензенты:

Азаров В. Н. — доктор технических наук, профессор кафедры «Менеджмент качества» Российского университета транспорта;

Солодков В. М. — кандидат экономических наук, директор Банковского института Национального исследовательского университета «Высшая школа экономики», руководитель программы МБА «Финансы и Банки»;

Левандо Д. В. — кандидат экономических наук, PhD, доцент Национального исследовательского университета «Высшая школа экономики»;

Винокуров И. В. — начальник Управления проектного менеджмента в государственном секторе Аналитического центра при Правительстве Российской Федерации.

Чекмарев, А. В.

Ч-37 Управление ИТ-проектами и процессами : учебник для вузов / А. В. Чекмарев. — Москва : Издательство Юрайт, 2022. — 228 с. — (Высшее образование). — Текст : непосредственный.

ISBN 978-5-534-11191-0

В учебнике рассматриваются теоретические основы управления проектами и процессами в области информационных технологий с привязкой к теории сложных систем, теории качества, теории конфликтов и объектно-ориентированного проектирования. Приведено значительное количество практических примеров, рассмотрены основные международные стандарты и наборы рекомендаций в области проектного и процессного управления, такие как CMMI, ITIL, ITSM, PMI PMBOK, ISO 9000.

Содержание учебника соответствует последним требованиям федерального государственного образовательного стандарта высшего образования.

Для студентов высших учебных заведений технического и экономического профилей, аспирантов, слушателей курса Master of Business Administration, преподавателей, а также практических работников в области информационных технологий и управления.

УДК 004+65.012(075.8)

ББК 65.290:32.97я73

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

ISBN 978-5-534-11191-0

© Чекмарев А. В., 2018

© ООО «Издательство Юрайт», 2022

Оглавление

Предисловие	7
-------------------	---

Раздел I ВСТУПИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

Глава 1. Понятие информации. Информационные ресурсы. Управленческий отрыв. Современные тенденции развития ИТ....	12
1.1. Информация	12
1.2. Информационные ресурсы и управленческий отрыв	13
1.3. Речь — книга — интернет.....	15
1.4. Новая промышленная революция	16
1.5. Кризис информационных технологий и <i>agile</i>	18
1.6. Развитие сетевых технологий и производительности	20
1.7. Стандартизация	20
Глава 2. Сложные системы. Статическая и динамическая сложность. Подходы к проектированию сложных систем. Оценка сложности	22
2.1. Понятие системы.....	22
2.2. Граница сложности	22
2.3. Статическая и динамическая сложность.....	23
2.4. Эмергентные свойства	24
2.5. Системное мышление	24
2.6. Приемы работы со сложностью	25
Глава 3. Экономическая теория институционализма.....	38
Глава 4. Теория общего управления качеством. Стандарт <i>ISO 9000</i> — управление качеством процессов	40
4.1. Контрольные карты и цикл постоянного улучшения.....	40
4.2. Закон Парето	43
4.3. Теория общего управления качеством	44
4.4. Стандарт <i>ISO 9000</i>	44
<i>Проверочные тесты к разделу I</i>	47

Раздел II УПРАВЛЕНИЕ ПРОЕКТАМИ

Глава 5. Проектная деятельность	50
5.1. Мировая статистика успешности ИТ-проектов The Standish Group	50

5.2. Основные проблемы проектного управления	54
5.3. Отличие проектной деятельности от текущей	59
5.4. Проект как адаптационный процесс	60
5.5. Стратегия, портфель, программа, проект	66
5.6. Офис управления проектами	66
Глава 6. Руководитель проекта	68
6.1. Психологические характеристики руководителя проекта	68
6.2. Функции руководителя проекта	71
6.3. Баланс прав и ответственности руководителя проекта	80
Глава 7. Устав проекта	81
7.1. Цели и приоритеты проекта	82
7.2. Планирование и контроль проекта	85
7.3. Пример контрольного (базового) плана ИТ-проекта	94
7.4. Управление рисками проекта	99
Глава 8. Стандарт PMI PMBOK. История создания стандарта.	
Стандартные процессы	103
8.1. История стандарта <i>PMBOK</i>	103
8.2. Жизненный цикл проекта <i>PMBOK</i>	105
8.3. Области знания <i>PMI PMBOK</i>	106
8.4. Группы процессов проекта	108
8.5. Инструменты и методы <i>PMI PMBOK</i>	113
Глава 9. Основные ошибки проектного управления	114
9.1. Необоснованная замена текущей деятельности проектной формой	114
9.2. Недооценка сложности проекта	116
9.3. Ошибка определения цели проекта	117
9.4. Ошибка определения ограничений проекта	121
9.5. Необоснованная передача внешней стороне функций руководства проектом	122
9.6. Ошибка определения требований проекта	122
9.7. Некорректное определение рисков проекта	123
9.8. Ошибка определения списка участников проекта	124
9.9. Ошибка выбора руководителя проекта	124
9.10. Нарушение баланса прав и ответственности руководителя проекта	125
9.11. Усложнение организационного построения проекта (включая необоснованное наращивание численности проектной группы)	126
9.12. Передача проектной группе полномочий по утверждению проектных документов	127
9.13. Недостаток взаимодействия и коммуникации со стороны проектной группы	128
9.14. Ритуализация	129
9.15. Формальное построение контроля на основе отчета руководителя проекта	131

9.16. Отсутствие системы мотивации участников проекта.....	132
9.17. Ошибка неиспользования уроков проекта.....	133
Глава 10. Гибкие методики управления проектами (<i>agile</i>)	135
10.1. Источники <i>agile</i>	135
10.2. Манифест <i>agile</i>	142
10.3. Основные методики <i>agile</i>	144
10.4. Место <i>agile</i> и его ограничения.....	150
10.5. Минимально жизнеспособный продукт (<i>MVP</i>)	152
10.6. Определение трудоемкости	154
<i>Проверочные тесты к разделу II</i>	155

Раздел III УПРАВЛЕНИЕ ИТ-ПРОЦЕССАМИ

Глава 11. Типовые ошибки процессного управления.....	158
11.1. Ориентация исключительно на результат с недооценкой контрольных функций.....	158
11.2. Совмещение разработки и администрирования автоматизированных систем	159
Глава 12. Зрелость ИТ-процессов по модели <i>CMMI</i>	163
12.1. История и цели создания <i>CMMI</i>	163
12.2. Принципы и результативность применения <i>CMMI</i>	166
12.3. Описание <i>CMMI</i>	174
12.4. Оценка зрелости.....	186
Глава 13. Стандарт <i>ISO 20 000 (ITSM)</i> и библиотека рекомендаций <i>ITIL</i>.....	189
13.1. История стандарта <i>ISO 20 000</i> и <i>ITIL</i>	189
13.2. Состав стандарта <i>ISO 20 000</i>	190
13.4. Библиотека <i>ITIL</i>	191
13.5. Метрики ИТ-процессов.....	204
13.6. Соглашение об уровне услуг — <i>SLA</i>	207
Глава 14. Управление непрерывностью, рисками и информационной безопасностью.....	210
14.1. Аварийное планирование	211
14.2. Анализ рисков	216
<i>Проверочные тесты к разделу III</i>	223
Рекомендуемая литература	226
Новые издания по смежным дисциплинам	227
Ключи к тестам.....	228

Предисловие

Инженерная задача — поиск оптимального проектного решения при наличии ограничений.

Неизвестный автор

В эпиграф вынесена формулировка, являющаяся одной из важнейших для обучения инженера. Она состоит из двух утверждений: первое — работа инженера состоит в поиске оптимума и улучшения результата своего труда, второе — его деятельность невозможна без определенности набора ограничений, временных, финансовых, функциональных, человеческих и множества других.

Соответственно, в предлагаемое пособие вошли материалы о современных подходах к развитию и оптимизации деятельности инженерных команд и руководителей в области информационных технологий, к выбору необходимых критериев оптимизации, а также описание основных проблем и ограничений, с которыми им приходится сталкиваться в своей деятельности. Другими словами, данная книга систематизировано излагает современные представления об эффективной организации работы и развитии службы информационных технологий.

Несмотря на использование в ряде случаев специальной терминологии, задачей данной работы не является повышение уровня специальных инженерных знаний непосредственно в области информационных технологий. При разработке пособия по мере возможности учитывалось требование компактности изложения. Техническая терминология представлена в объеме, достаточном для налаживания эффективного взаимодействия со специалистами-инженерами.

Сведения о современных программных и аппаратных средствах приводятся в качестве иллюстрации применяемых организационных и бизнес-решений и подходов, для демонстрации возможностей и ограничений современных технологических продуктов.

Пособие не является рекламным материалом или справочником по применяемым программно-техническим продуктам. Наименования компаний и систем приводятся в качестве примеров, а не в качестве рекомендованного, лучшего или единственного выбора.

На основе опыта преподавания предмета слушателям, не имеющим технического образования, автор посчитал возможным изложить в рамках книги некоторые основы теории сложных систем, системного мышления и теории информации.

Некоторые материалы носят полемический характер, который отражает дискуссии, имеющие место в научно-технической среде в рамках реализации информационных проектов, а также содержат личностное отношение автора. В целях представления читателю максимально освобожденных от субъективизма сведений приводятся альтернативные точки зрения на проблемы.

В связи с очевидными требованиями ограничить распространение информации о недостатках отдельных компаний в примерах их названия не упоминаются.

Учебное пособие не является чисто теоретическим, поэтому неизбежно будет устаревать с развитием российской и международной индустрии информационных технологий. Автор постарался обеспечить книге возможно длительную актуальность за счет исключения описаний короткоживущих технических решений, концентрируя внимание на основных долговременных тенденциях.

Книга предназначена для широкого круга читателей, в том числе не являющихся инженерами, для слушателей курса по управлению предприятиями промышленного, государственного и финансового сектора различного уровня, заинтересованных в построении эффективно работающей информационно-технологической поддержки бизнеса, руководителей информационно-технологических подразделений, в качестве учебного пособия для студентов старших курсов бакалавриата и магистратуры, обучающихся по технически и экономическим направлениям, а также курса МВА. Материал книги может оказаться полезным и для сотрудников служб информационно-технологической поддержки, а также сотрудников информационной безопасности. В целом пособие предназначено для тех, кто управляет специалистами, которые, в свою очередь, управляют информационными системами.

Материал пособия подкреплен реальным опытом, полученным при реализации технических и бизнес-проектов в крупнейших организациях России, в том числе в Центральном банке РФ, на Московской межбанковской валютной бирже, Фондовой бирже РТС, в Национальном депозитарном центре, «ВТБ», компаниях «Аэрофлот» и «Газпромэкспорт». Этот опыт позволил квалифицированно изложить материалы, касающиеся специализированных бизнес-применений средств автоматизации, таких как валютная и фондовая биржа, система бронирования авиабилетов, брокерское обслуживание, казначейство банка, депозитарий и др.

Опыт взаимодействия с крупнейшими иностранными поставщиками решений по автоматизации, консалтинговых и информационных услуг позволил значительно повысить полноту представленных материалов и их соответствие международным стандартам.

Основной объем содержащейся в пособии информации использовался на протяжении длительного времени при чтении лекций и проведении семинаров в Банковском институте Высшей школы экономики для слушателей курсов повышения квалификации Банка России,

информационно-технологических компаний, студентов МВА и второго высшего образования, а также в Московском институте электроники и математики для студентов специализации «Управление информационными сервисами». Это помогло выверить доступность формы изложения, информативность и полноту предлагаемых материалов.

При разработке структуры книги автор исходил из необходимости обеспечить возможность ее использования не только в качестве последовательно изучаемого текста, но и справочника со сквозным структурированием материала. Текст разбит на 14 глав, соответствующих сложившейся последовательности лекций и имеющих самостоятельное изложение, что приводит к некоторой повторяемости материала. Каждая глава дополнительно разбита на параграфы для облегчения поиска информации по содержанию. Читатель по своему усмотрению может выбирать последовательность изучения материала и пропускать темы, не актуальные на данный момент.

Последовательность изложения материала основана на принципе «от общего к частному». Этот принцип соблюдается как в целом по тексту, так и в рамках отдельных глав.

Глава 1 включает информацию об основных долговременных тенденциях развития информационных технологий в исторической перспективе, анализ их значимости в хозяйственной жизни, высокоуровневое описание основных концепций, рассматриваемых в книге.

Главы 2—4 посвящены теоретическим вопросам, базовым для понимания более специальных тем, таким как сложные системы, экономическая теория институционализма, общая теория качества.

Главы 5—10 содержат сравнительную характеристику существующих международных стандартов и лучших практик, применяемых в настоящее время для эффективного управления ИТ-проектами. В главе 10 изложены основные особенности гибкого управления проектами (англ. *agile*).

В современной практике руководители различного уровня систематически сталкиваются с задачей квалифицированного принятия решений в области процессного управления и оценки уровня зрелости процессов и подразделений. Главы 11, 12 предлагают достаточно подробную информацию о типовых ошибках процессного управления и модели зрелости.

В главе 13 приведено выборочное описание *Information Technology Infrastructure Library* и стандарта *ISO 20000 ITSM*, которые являются основой современного построения процессного управления в информационных технологиях.

Темам анализа рисков, защиты информации и аварийного планирования, без которых современное управление информационными технологиями стало невыполнимым, посвящена глава 14.

В начале каждой главы приведен список ожидаемых знаний и умений, которые должны быть получены читателем в результате изучения. В завершение глав приведены проверочные вопросы, которые могут

помочь читателю проверить свой уровень восприятия прочитанного материала.

Автор будет благодарен за предоставление отзывов и комментариев по предложенной книге. Это позволит не только сделать ее лучше в дальнейшем, но и, возможно, откорректировать подходы, применяемые им в практической и учебной деятельности. Отзывы можно отправить по электронной почте anatolii_chekmar@mail.ru или через сайты www.anatoliichekmar.wixsite.com/personal, www.urait.ru.

Раздел I

ВСТУПИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

В результате изучения данного раздела студент должен:

знать

- основные направления развития информационных технологий;
- базовые положения экономической теории институционализма;
- основные положения общей теории качества;

уметь

- использовать инструменты теории качества;

владеть

- навыками системного мышления;
 - приемами визуализации сложных систем;
 - основными приемами работы со сложными системами.
-

Глава 1

ПОНЯТИЕ ИНФОРМАЦИИ. ИНФОРМАЦИОННЫЕ РЕСУРСЫ. УПРАВЛЕНЧЕСКИЙ ОТРЫВ. СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ИТ

1.1. Информация

Существует большое количество альтернативных определений термина «информация» применительно к различным областям знаний. Американский ученый, основоположник кибернетики и теории искусственного интеллекта Н. Винер, например, считал **информацию** основополагающим термином, не поддающимся определению в более простых терминах и, определял ее как «некую субстанцию, способную передаваться между объектами и системами, при этом не являющуюся материей или энергией».

В области информационных технологий превалирует математическое определение, введенное американским инженером и математиком К. Шенноном, основанное на вероятностной оценке.

Не приводя формулу расчета информации, на основе определения К. Шеннона скажем, что информация, получаемая о некоторой системе или объекте в результате переданного сообщения, количественно равна уменьшению неопределенности (незнания) о текущем и будущем состоянии данной системы для наблюдателя. Другими словами, информация, переносимая сообщением, равна уменьшению энтропии, под которой понимается мера неопределенности или хаоса.

Кроме того, следует упомянуть, что количество получаемой информации находится в обратной зависимости от вероятности получения определенного сообщения, т.е. чем менее вероятна новость, тем больше информации получено. Например, сообщение о том, что игральная кость выпала на определенную грань, несет больше информации, чем сообщение о том, что подброшенная монета выпала решкой, так как вероятность выпадения определенной грани на игральной кости — $1/6$, монеты — $1/2$.

Интересным следствием данного определения является утверждение о том, что менее вероятное сообщение, переносит больше инфор-

мации, оказывает большее влияние на получателя. Например, неожиданная новость экономического или политического характера должна привести к большему отклонению биржевого курса, чем ожидаемая (избрание кандидата, в ходе предвыборной кампании занимавшего менее высокую позицию в рейтинге, более информативно, чем кандидата с более высоким рейтингом и соответственно сделает валютный курс более волатильным, т. е. изменчивым). Сообщение о событии, вероятность которого оценивается в 100 %, не несет информации само по себе. В этом случае информацию может нести только период времени появления данного сообщения.

Математическое определение информации, основанное на противопоставлении данного понятия понятию энтропии (незнание, хаос, неопределенность), кроме прочего, позволяет сформулировать миссию ИТ-инженера в организации, основной задачей которого является борьба с неопределенностью, непрозрачностью, непредсказуемостью системы автоматизации и хозяйственного механизма организации в целом. Как следствие, если в результате деятельности *CIO* (от англ. *chief information officer* — высший руководитель в организации по информационным технологиям) в течение определенного периода количество сбоев автоматизированных систем (непредсказуемость автоматизированных систем) выросло или осталось постоянным, а при этом прозрачность и управляемость бизнеса для руководства не увеличилась, то в компании появился кандидат на увольнение.

Как уже указывалось, начало развитию науке об информации было положено в 1940-х гг. в рамках создания и развития новой науки, названной кибернетикой. Она зародилась в качестве науки об управлении и обратной связи в автоматизированных системах. Родоначальник кибернетики Норберт Винер в 1948 г. издал книгу «Кибернетика, или управление и связь в животном и машине», считающуюся сегодня основополагающей работой для этой дисциплины.

Кибернетика, получившая дальнейшее развитие в качестве науки об информационных технологиях, и в настоящее время не потеряла своего значения как наука об управлении.

1.2. Информационные ресурсы и управленческий отрыв

Возможно, читатель имеет опыт работы в крупной государственной или частной компании. В этом случае он с высокой вероятностью сталкивался с автоматизированной обработкой информации, а также с сообществом высококвалифицированных специалистов, обычно именуемых информационными инженерами, программистами и т. д.

В рамках такого взаимодействия у каждого современного управленца накапливаются вопросы. Основной из них — насколько большое внимание, выражающееся в финансировании и других ресурсах, тре-

буют поддержка и развитие информационных технологий в организации?

Уже к середине 1990-х гг. доля трудоспособного населения развитых стран, занятая в обработке информации, превысила 50 % (рис. 1.1), а затраты их экономик на обработку информации превысили уровень затрат на энергетику. То есть информация фактически стала основным предметом труда в развитых странах¹.

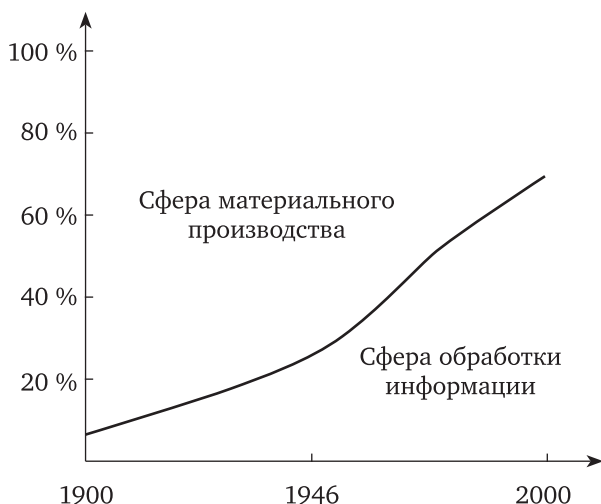


Рис. 1.1. Доля трудоспособного населения, занятого в обработке информации в развитых странах

Учитывая размер «информационной» доли экономики, можно сделать очевидный вывод — развитие технологий обработки информации играет определяющую роль в повышении производительности труда и его качества, эффективности и конкурентоспособности экономик развитых стран.

Данный вывод справедлив для хозяйственной системы или отдельной организации любого масштаба, характеризующейся преобладанием «информационной» занятости, например, банка. В любом банке только водители, уборщицы, охрана и другой вспомогательный персонал не заняты обработкой информации. Даже кассиры относятся к категории «информационных» сотрудников — банкноты и монеты являются информационными носителями.

Превосходство США в области обработки информации стало залогом обеспечения для данного государства глобального управленческого отрыва, который выражается в превосходстве систем политического и социального управления, управления вооружениями, управления хозяйственными системами и т.д. США в настоящее время способны максимально быстро получать информацию, необходимую для при-

¹ Громов Г. Р. Очерки информационной технологии. М. : ИнфоАрт, 1993.

нятия управленческих решений, принимать наиболее оптимальные управленческие решения, предельно оперативно доводить управляющие импульсы до исполняющих элементов, а также получать информацию об успешности выполненных действий (реализовывать эффективную обратную связь).

1.3. Речь — книга — интернет

Повышение эффективности управления — не единственная цель в развитии информационных технологий. С точки зрения накопления и обмена знаниями четко прослеживается тенденция к росту доступности знаний для населения и развития техник их целенаправленной доставки для эффективного воспроизводства и умножения.

До появления сложной речи полученные навыки и знания вынужденно погибали с очередным первооткрывателем. Затем люди научились размножать знания посредством речи. Данный канал передачи не был эффективным. В процессе передачи значительная часть информации искажалась. Количество субъектов, подвергающихся информационной обработке, не могло быть большим. Передача занимала много времени. Сложные концепции в процессе передачи искажались до такой степени, что полностью теряли свою ценность. Можно говорить о том, что канал передачи накладывал достаточно жесткие ограничения на объем передаваемых знаний. Рост общей массы знаний ограничивался локальными деланками, которые при этом погибали быстрее, чем создавались в связи с очередным природным или социальным катаклизмом.

С изобретением письменности эффективность канала распространения знаний резко выросла как с точки зрения емкости, так и с точки зрения защищенности от помех и оперативности. Рассказчик получил возможность обращаться одновременно к множеству читателей. При этом переносимые знания практически не искажались. Генераторы новых знаний получили в свои руки инструмент для сева зерен знаний в глобальном масштабе. Вероятность того, что они упадут на плодородную почву, значительно выросла. Началась эпоха развития науки и широкого обучения.

В то же время рукописные книги не позволяли создавать по-настоящему массовые каналы распространения знаний. Их стоимость и уязвимость стали основным сдерживающим фактором. Кроме того, немалое значение имело творчество отдельных переписчиков, вносящих из лучших побуждений искажения в тексты, иначе говоря, помехи в канал передачи информации. Цикл воспроизводства и умножения знаний был неустойчив.

Ситуацию исправило изобретение в середине XV в. европейского книгопечатания, которое совместно с появлением дешевых технологий производства бумаги создало возможность для массового распространения знаний и их экспоненциального роста.

Радио и телевидение в качестве средств массовой информации сыграли ведущую роль по историческим меркам в рамках антракта перед основным действием. Спустя полвека после их внедрения ведущая роль перешла к Интернету.

Благодаря *World Wide Web* и родственным гипертекстовым технологиям знания начали самоорганизовываться в глобальные оперативно доступные и непрерывно обновляемые массивы. Поисковые машины позволили получать информацию по запросу, практически без затрат и в полноте, которая превосходит любые возможные потребности, а не последовательно с высокими трудозатратами и в ограниченном объеме.

Социальные сети привели к созданию пока еще неокрепших, но уже явно просматриваемых групповых интеллектов, которые обрабатывают и продуцируют информацию коллективно.

Блокчейн и *wiki*-технологии добавляют последние штрихи, гарантируя защиту информации от искажений и превращая знания в непосредственный оперативный инструмент влияния на реальный мир.

1.4. Новая промышленная революция

Следующий вопрос касается долговременных перспектив развития информационных технологий — следует ли в ближайшее время ожидать значительных революционных сдвигов в технологиях автоматизации?

Проанализируем текущее состояние информационной революции и ее перспективы по аналогии с имевшей место в XIX—XX вв. промышленной революцией (рис. 1.2).

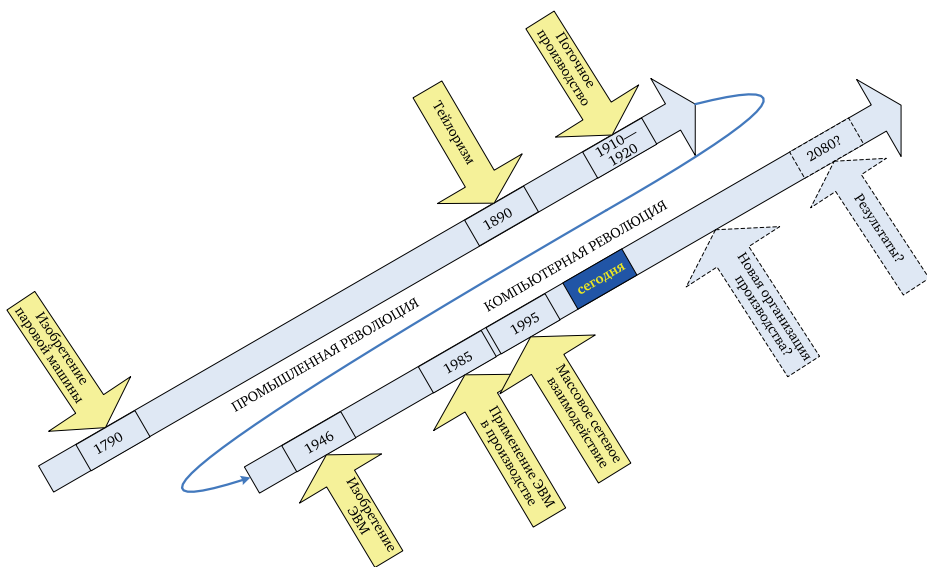


Рис. 1.2. Временные шкалы промышленной и компьютерной революций

Сравним две временные шкалы. На первой отмечен путь реализовавшего свой потенциал (паровая машина) поточного производства. На второй — находящееся в середине своего пути развитие ЭВМ.

От изобретения паровой машины до разработки новых концептуальных основ производства, основанного на энергии пара (тейлоризм), прошло около 100 лет. Спустя 30 лет человечество получило первый результаты внедрения новых подходов.

Изобретение компьютеров отстоит от нашего времени примерно на 70 лет. Несмотря на ускоряющийся прогресс научно-технологической мысли, к настоящему времени не создано концепции нового типа производства. То, что мы можем наблюдать в этой области, это отдельные предположения визионеров, далекие от целостного предложения по реорганизации производства. Наиболее вероятным направлением развития представляется переход от поточного типового производства с высоким уровнем специализации работников на малоинтеллектуальных операциях к производству изделий с высоким уровнем кастомизации и к вытеснению человека из области специализации ручных операций в область проектирования. Несмотря на то, что теоретические разработки в области новой организации труда, в основе которой лежат информационные технологии, пока далеки от завершенности, по мнению ряда исследователей, основной результат информационной революции, скорее всего, должен формулироваться как повышение управляемости производства и, в отличие от промышленной революции, переход от массового к мелкосерийному заказному производству. Этот переход должен выразиться в расширении возможностей для потребителя управлять опциями покупаемых товаров на этапе выбора.

Безусловно, разработка упомянутых теоретических основ может не занять столетие, как в случае промышленной революции. Но даже учитывая значительное превосходство нынешнего научного потенциала, завершения данной работы не следует ожидать в непосредственном будущем. То же относится и к основным результатам информационной революции для экономической системы, аналогом которых в первой промышленной революции стали общедоступные телевизоры, холодильники, автомобили и т.п.

«Информационная гонка» только начинается, а отставание отдельных участников вполне может оказаться краткосрочным в рамках общего периода изменений. Тем важнее сократить до минимума период осмысления масштаба предстоящих изменений и их значимости.

Возможно, ранее других стран данная идея собрала критическую массу сторонников в США. Начиная с 1960-х гг., Соединенные Штаты Америки реализуют программу, выводящую **информационные ресурсы**, под которыми понимается информация, средства ее обработки, хранения и передачи, в разряд основного возобновляемого ресурса и средства производства, соответственно.

Именно США к настоящему времени располагают наиболее развитыми банками данных (хранилища информации), в том числе

по финансовой статистике (например, «Блумберг»), наиболее быстродействующими компьютерами, наиболее развитыми сетями передачи данных и, что является более важным, крупнейшим научным потенциалом в области информационных технологий в мире¹.

1.5. Кризис информационных технологий и *agile*

Международная статистика реализации проектов в области информационных технологий демонстрирует удручающие цифры — подавляющее большинство таких проектов (по некоторым оценкам до 90 %) завершаются со значительным запаздыванием, требуют значительной корректировки в части целей и ресурсов или прекращаются без достижения планируемого результата. При этом эффективно применяемыми оказываются не более 11 % разрабатываемых программ². В чем причина низкой эффективности работы служб, связанных с сопровождением и развитием информационных технологий?

В литературе можно встретить удачную аналогию — сравнение области автоматизации и месторождения полезных ископаемых. На начальном этапе затраты на добычу минимальны — минеральные ресурсы добывают практически с поверхности. С течением времени добыча все более затрудняется и в какой-то момент времени становится нерентабельной до появления очередного прорыва в технологиях.

Аналогичный процесс происходит и в индустрии обработки информации. Накопленные за длительный период в рамках развития точных наук и инженерии наборы типовых простейших алгоритмов оказались автоматизированы (исчерпаны) за первые десятилетия развития электронно-вычислительных машин. В настоящее время задачи, стоящие перед программистами, подчас оказываются настолько сложными, что их автоматизация на текущем уровне технологий либо не представляется возможной, либо не является рентабельной. То есть сложность обрабатываемых процессов и объектов достигла уровня, на котором затруднено получение быстрого и гарантированного результата в части автоматизации (*time to market* значительно возрастает), а обрабатывающие автоматизированные системы в силу своей сложности демонстрируют непредсказуемость, выражающуюся в снижении надежности. Еще более усугубляет ситуацию то, что функционирование автоматизированных комплексов следует рассматривать как совместное с их пользователями и администраторами, что резко повышает сложность подобного комплекса.

Описанное явление в индустрии информационных технологий начало проявляться достаточно давно и к настоящему времени приобрело характер общеизвестного кризиса. Этот кризис привел к выде-

¹ Громов Г. Р. Очерки информационной технологии.

² Громов Г. Р. Очерки информационной технологии.

лению двух тенденций развития информационных технологий: разработка методов эффективного управления процессами в области сопровождения и развития ИТ с особым упором на их «человеческую» составляющую и разработка методов формализованного описания объектов реального мира на языках, с одной стороны, приближенных к естественному, с другой — строгих в достаточной степени для того, чтобы обеспечить облегченное «машинное» представление (программирование).

Развитие первой тенденции к настоящему времени выразилось в создании стандартов, наборов рекомендаций и моделей в части управления ИТ-службами и процессами — *ISO 20000*, *COBIT*, *Information Technology Infrastructure Library*, *Capability Maturity Model*, *SCRUM/agile*, *DevOps* и др. В общем виде данная тенденция формулируется как «лучшая практика — это следование лучшим практикам». Лучшие практики закрепляются в постоянно развивающихся стандартах, развитие которых сопровождается доработкой методик объективной оценки соответствия организаций их требованиям.

Возрастающее давление кризиса, связанного с усложнением задач автоматизации и требованием потребителей по ускорению их решения, привело к тому, что разработка программного обеспечения к настоящему времени стала флагманом в части реформирования инженерных подходов к созданию сложных систем, которые в общем виде сводятся к поэтапному каскадированию задач (*waterfall*) с их последовательной декомпозицией.

Провозглашенный сообществом разработчиков программного обеспечения в феврале 2001 г. манифест *Agile software development* явился реакцией на затяжной кризис в разработке программного обеспечения и скорее продекларировал необходимость изменений, чем указал точный путь.

К настоящему времени накоплен значительный опыт успешного и неуспешного внедрения подходов, базирующихся на *agile*. Далее более подробно будут рассмотрены конкретные методы работы в этой парадигме. Сейчас стоит отметить то, что развитие сетевого взаимодействия, усложнение автоматизированных объектов, появления средств быстрой разработки и прототипирования ПО, доступных для непрофессионалов, инструментов поддержки конвейерной разработки ПО и другое не могло не привести к глубокому пересмотру применявшихся традиционных инженерных подходов.

В рамках второй тенденции, касающейся инструментария проектирования программ, создан значительный задел в виде парадигмы объектно-ориентированного проектирования, автоматизированных систем, реализующих эту парадигму с применением языков формализованного описания, например, *UML* (от англ. *universal modelling language*) и других, а также в виде развития универсальных форматов описания документов — *XML*. Достоинством данных форматов является то, что они предусматривают в составе документов описание методов

их обработки и позволяют создавать описание все более сложных объектов путем эволюционного развития и комбинации ранее описанных.

1.6. Развитие сетевых технологий и производительности

Широко известен в различных интерпретациях сформулированный в 1965 г. эмпирический закон Мура, подтверждающийся на протяжении последних десятилетий и определяющий темп количественного роста производительности (или количества активных элементов процессора) средств обработки данных как двукратный за полтора года.

Хорошей иллюстрацией к этому закону в части средств хранения данных может послужить то, что считавшееся большим к 2000 г. дисковое хранилище размером в 1 терабайт применялось при построении центра обработки данных крупной международной авиакомпании, а в настоящее время — это небольшая медиатека для домашнего использования.

С 2000 г. прошло более 15 лет. Это значит, что полуторалетний цикл удвоения производительности вычислительных систем сработал более 10 раз. Компьютеры при той же стоимости стали примерно в тысячу раз производительнее (два в десятой степени равно 1024). Емкость их накопителей и пропускная способность каналов связи выросли также в тысячу раз.

В 1994 г. на валютных торгах Московской межбанковской валютной биржи отсутствовала локальная сеть (ее установили только в 1995 г.). Если студент московского вуза в том же году решил бы отправить запрос на анкету для поступления в аспирантуру в США, то ему бы пришлось воспользоваться услугами одного из пары доступных в Москве центров, предоставляющих услуги отправки электронной почты, или рассылать письма на бумажном носителе. Применяемые сети были крайне ограничены, количество пользователей измерялось десятками. В настоящее время сетевое взаимодействие объединяет миллиарды пользователей и устройств.

1.7. Стандартизация

При рассмотрении основных тенденций развития информационных технологий нельзя обойти вниманием нарастающую стандартизацию во всех сферах. Количество используемых сетевых протоколов консолидируется в области *IP*, форматы передаваемых и сохраняемых файлов графики после многотысячного разнообразия 1980—1990-х гг. свелись сегодня практически исключительно к *JPEG*, аппаратные платформы явно тяготеют к *Intel*-подобным конфигурациям и наборам команд.

Подобные тенденции стандартизации можно обнаружить во многих областях информационных технологий. С одной стороны, данное явление характерно для индустрии, приблизившейся к зрелости. С другой

стороны, в отсутствии подробного анализа его причин и выработки корректирующих мероприятий данная тенденция может вместо зрелости привести к застою и снижению темпов развития.

В целом же, как без изобретения стандартизированного кирпича человечество не имело возможности строить многоэтажных зданий, так и без изобретения стандартных информационных «кирпичиков» нет возможности перейти на следующие уровни сложности в построении информационно-технологических объектов.

Созвучие с технологией блокчейн не является случайным. В последнее время явно прослеживается тенденция внедрения «кирпичного» подхода не только в выстраивании технологий защиты целостности данных, но и в построении схем контролируемых цепочек сборки программ и даже бизнес-процессов. К этой категории можно отнести, например, модели поставки модулей программ с открытым кодом на основе блокчейн и смарт-контрактов.

Типизация и стандартизация программ нашла отражение также и в реализации большинства методологий гибкой разработки *agile*. Их основным атрибутом сегодня является так называемый *minimum viable product (MVP)* — минимально жизнеспособный продукт. Его реализация стандартной командой из 7—10 человек должна укладываться в единичный спринт (от 1 недели до месяца), а внедрение — создавать дополнительную распознаваемую клиентами ценность автоматизированного продукта.

Таким образом, унификации и стандартизации в настоящее время подвергается уже не «железо» или «софт», а постановка задачи и трудоемкость этапов проекта.

Для более подробного анализа основных положений, приведенных в данной главе, можно порекомендовать работу Г. Р. Громова «Очерки информационной технологии».

Глава 2

СЛОЖНЫЕ СИСТЕМЫ. СТАТИЧЕСКАЯ И ДИНАМИЧЕСКАЯ СЛОЖНОСТЬ. ПОДХОДЫ К ПРОЕКТИРОВАНИЮ СЛОЖНЫХ СИСТЕМ. ОЦЕНКА СЛОЖНОСТИ

2.1. Понятие системы

В проектной деятельности часто приходится сталкиваться с задачей оценки потребности в ресурсах и времени, необходимых для реализации проекта. Один из подходов заключается в том, чтобы сравнить количественные параметры создаваемой системы с опытом проектов, реализованных ранее. При этом плановые оценки можно вывести на основе некоторых поправочных коэффициентов.

Естественно, более сложный проект (более сложной системы) потребует больше времени и больших затрат. Что такое сложный или простой проект, сложная или простая система?

Системой является обособленная от окружающей среды совокупность особым образом связанных компонентов, которая подвергается воздействию среды и, в свою очередь, оказывает воздействие на среду.

Исходя из определения, сложной является система, имеющая большое количество входящих в нее объектов и связей между ними. При этом под большим количеством понимается количество объектов, затрудняющее их одновременный контроль или анализ, т. е. делающее систему непредсказуемой и тяжело контролируемой.

Отметим, что здесь возникает внешняя для системы роль наблюдателя или контролера, который при этом образует с наблюдаемой или управляемой системой некоторую надсистему. Например, на бытовательском уровне молоток для мастера представляется простой системой, а для дилетанта немедленно превращается в сложную непредсказуемую систему, часто с фатальными последствиями.

2.2. Граница сложности

В широко известной книге Г. Буча «Объектно-ориентированный анализ и проектирование с примерами приложений»¹ между простой

¹ Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений. М. : Вильямс, 2008.

и сложной системами проведена следующая граница: если система состоит из восьми объектов и менее, то она простая; из более восьми — сложная.

Здесь критерием выбрана способность мозга человека (наблюдателя) контролировать некоторое количество объектов одновременно. Таким образом, граница между сложным и простым проведена в мозге человека, на базе его способности воспринимать информацию о некотором количестве объектов одновременно.

Например, минимальное тактическое воинское подразделение (отделение) в количестве около 10 человек поддается эффективному контролю младшего командира в стрессовой динамичной ситуации. Увеличение количества подчиненных привело бы к недопустимому росту сложности контроля и потере управляемости, что было установлено длительной практикой военных действий.

Игральная кость — на первый взгляд простой объект, но ее будущее состояние непосредственно после броска плохо предсказуемо. Дело в том, что в этот момент времени она оказывается включена в чрезвычайно сложную систему, состоящую в том числе из бросающей руки, гравитационно воздействующей Земли, сопротивляющегося воздуха, подставленной поверхности стола, а также, по отдельным мнениям, из планет Солнечной системы и ближайших звезд. Исходя из этого подхода, граница определяется возможностью прогнозирования поведения системы. Сложная система плохо прогнозируется.

2.3. Статическая и динамическая сложность

В математической теории различают два основных вида сложности: статическую и динамическую.

Под *статической сложностью* понимают количество объектов, входящих в систему, и количество связей между ними.

Динамическая сложность учитывает изменяющиеся с течением времени процессы, происходящие в системе (между составляющими объектами) и с ее участием.

Для иллюстрации можно привести пример военного или гражданского самолета, который состоит из сотен тысяч, а иногда и миллионов комплектующих. Спроектировать такую инженерную систему, обеспечив пространственное размещение и крепление комплектующих друг к другу, — задача сама по себе почти нереализуемая. Но после ее первичного решения необходимо обеспечить взаимодействие всех подсистем в условиях полета в различных скоростных режимах в присутствии разнообразных дестабилизирующих факторов (температура, пыль, электромагнитные излучения, химический состав воздуха, боевое противодействие и т. п.).

2.4. Эмергентные свойства

Дополнительную проблему в анализе и проектировании сложных систем представляет то, что при объединении (взаимодействии) двух или нескольких систем объединенная система получает набор свойств и параметров, не являющийся суммой параметров и свойств вошедших систем. Ни электронная управляющая система, ни корпус, ни реактивный двигатель ракетного носителя по отдельности не способны выйти на околоземную орбиту. Только их корректное совмещение обеспечивает получение необходимых рабочих свойств.

Еще более красноречивую иллюстрацию предоставляет организм человека — какой из органов позволяет создавать произведения искусства?

Такие «возникающие» свойства систем называют *эмергентными*. Часто из наличия данных свойств выводят определение системы как сущности, состоящей из множества элементов, которые при корректном объединении обеспечивают проявление новых значимых «системообразующих» свойств.

Часто в инженерной практике встречается обратная ситуация, когда в результате совмещения эффективно функционирующих подсистем нарушается корректное функционирование.

Например, спроектированное и отестированное танковое орудие полностью соответствует техническому заданию. Испытания на стенде пройдены успешно. Та же ситуация имеет место для корпуса и ходовой подсистемы, которые тестировались с размерным и весовым имитатором орудия, но при установке орудия на корпус и попытке стрельбы на ходу перпендикулярно движению произошло опрокидывание.

Подобные случаи выявляются при так называемых комплексных испытаниях. Испытания подсистем по отдельности называются частными и обычно рассматриваются как предварительные.

Схожие проблемы присутствуют при создании автоматизированных систем, которые в качестве систем управления отражают на логическом уровне сложность систем внешнего физического мира и дополняются сложностью поддерживающих электронных систем. При этом многие электронные элементы к настоящему времени за счет миниатюризации и высоких частот уже работают в рамках квантовой физики, которая описывается нечеткой вероятностной логикой.

2.5. Системное мышление

С инженерной точки зрения важно то, что практически любая функционирующая система (архитектурная, механическая, электронная и т. д.) является сложной системой и, на первый взгляд, не поддается эффективному анализу, прогнозированию и контролю, поскольку нередко состоит даже не из десятков, а из миллионов элементов, дина-

мически взаимодействующих друг с другом. Однако человечество достаточно давно выработало ряд подходов, которые позволяют обходить ограничение способности мозга справляться со сложностью и в настоящее время отражаются в рамках методологии, часто называемой системным мышлением.

Эта методология явилась результатом развития таких теоретических и практических подходов, как анализ Декарта, теория хаоса, кибернетика и др. При этом теория хаоса развивает свойственный восточной философии интуитивный взгляд на системы как нечто целостное, характеризующееся состоянием (точки притяжения, аттракторы) и поведением, игнорируя особенности входящих элементов. Анализ Декарта склонен к детализации и рассмотрению элементов системы. Кибернетика систематизирует инженерную практику и распространяет ее подходы в том числе на живые и общественные системы, при этом придавая большое значение явлениям обратной связи.

Ключевым в системном мышлении является рассмотрение объектов реального мира (и себя) в качестве совокупности взаимодействующих элементов (системы). При этом необходимо совмещать первичное представление с альтернативным взглядом на себя и любые системы как на приносящие индивидуальный вклад элементы более масштабных систем. То есть каждая система является элементом более крупной системы.

Другими словами, системное мышление состоит в совмещении видения комплексности любого объекта и его включенности в другие системы в зависимости от цели рассмотрения. При этом деление реального мира на отдельные системы и уровень детализации рассмотрения определяются сознательно с пониманием ограничений такого рассмотрения, исключительно по воле наблюдателя, исходя из краткосрочных целей его деятельности, и должны меняться при изменении этих целей.

Например, при обучении ребенка живописи было бы не очень эффективно рассматривать его как совокупность органов. Но при аппендиците хирург немедленно выделяет элемент организма, подлежащий удалению. При этом ни в том, ни в другом случае человек не рассматривается как элемент сообщества (часть системы более высокого уровня), совокупности множества разнородных живых клеток или даже атомов (множества взаимодействующих элементов микроуровня). Это представление может использоваться, но в других случаях.

2.6. Приемы работы со сложностью

Абстрагирование. При разработке функциональных требований к автоматизированной системе необходимо корректно выбрать уровень рассмотрения (уровень детализации). На этом этапе проектирования система рассматривается как целостная сущность, взаимодействующая с другими автоматизированными системами и пользователями —

так называемый «черный ящик». Описание в требованиях отдельных элементов проектируемой системы было бы ошибкой, так как в этом случае игнорируется ограниченность способности автора требований и других заинтересованных лиц (заказчика, исполнителя и т. д.) эффективно с минимальным количеством ошибок проанализировать реальные потребности предметной области и согласовать проектный документ на нескольких уровнях детализации одновременно. Результатом будет затягивание согласования требований, их неполнота и множественные ошибки.

Описанный выше подход в инженерной методологии называют абстрагированием. В общем случае абстрагирование — это исключение из рассмотрения деталей системы (или ее связей), не важных с точки зрения проектировщика на текущем этапе работы.

Например, в большинстве случаев при проектировании стационарных (не перемещаемых) автоматизированных систем не учитывается гравитационное взаимодействие. При этом учет массы вычислительных элементов в требованиях становится обязательным при создании бортовых систем или при проектировании центров обработки данных в части конструкции перекрытий здания.

Частой ошибкой проектирования является попытка реализации системы без точной привязки к моменту времени или этапу жизненного цикла проекта (продукта). В этом случае проектировщик пытается реализовать свое представление о системе без фиксации целевого состояния проекта, как бы растянутое по времени, причем на неограниченную перспективу. Именно в таких случаях в функциональных требованиях появляются пункты о «возможности гибкой настройки пользователем», «универсальности», «поддержке разнородных перспективных платформ» и т. д. Таким образом, проектировщик или руководитель проекта значительно усложняют свою задачу за счет увеличения динамической сложности.

Абстрагирование помогает справиться не только со статической сложностью, но и с динамической.

В целом подход абстрагирования предусматривает сознательное и обоснованное наложение максимального количества ограничений на проектирование, систему, реализуемый проект, исходя из приоритетов оперативных целей и задач, решаемых в текущий момент времени. Наличие ограничений позволяет успешно оптимизировать выбор проектного решения из доступного пространства альтернатив. Понятно, что оптимальный выбор из бесконечного набора альтернатив затруднителен, что часто на практике приводит к неудаче реализуемых проектов.

Декомпозиция и иерархия. Следующий подход — иерархия — является противоположностью и одновременно дополнением абстрагирования. При абстрагировании не рассматриваются внутренние детали системы и проектирование или анализ не могут быть полностью выполнены. Иерархия позволяет устранить этот недостаток. Рассмо-

трех систему в целом, мы можем выполнить ее декомпозицию на ограниченное количество подсистем, которые будут подвергнуты индивидуальному анализу в качестве «черных ящиков» на следующем этапе. Учитывая ограничения мыслительных способностей человека, не рекомендуется превышение количества этих элементов свыше 8—10. Это позволит сохранить представление об их взаимодействии в процессе работы.

Подобную декомпозицию представления о проектируемой или анализируемой системе проводят неоднократно до достижения необходимого уровня детализации, создавая, таким образом, иерархию представлений, которые на каждом из уровней доступны для восприятия аналитиком или соответствующего уровня. Этот аналитик может быть в свою очередь представителем иерархической организационной структуры, в которой на верхнем уровне детализации оказывается, например, генеральный конструктор, или выполнять свою работу последовательно, перемещаясь между уровнями представления сверху вниз или элементами декомпозиции в рамках одного уровня детализации.

На практике часто встречается ошибка нарушения иерархии, когда в рамках одного проектного документа или раздела документа необоснованно делается попытка описания системы сразу на нескольких или не соответствующих уровнях детализации для выделенных подсистем, что приводит к нарушению методологии, обеспечивающей успешность проектирования, в особенности при совместной работе крупных коллективов.

Если поручить декомпозицию одного и того же объекта специалистам в различных областях или располагающим различными командами, то полученные результаты в виде иерархий, скорее всего, будут различаться.

Проведение декомпозиции на различных этапах и уровнях детализации может базироваться на различных принципах. Наиболее часто встречается «традиционная» декомпозиция. В этом случае сложную систему разбивают на элементы в соответствии с ранее сложившейся практикой, например, автомобиль принято рассматривать как комплекс, состоящий из двигателя, трансмиссии, ходовой части, системы управления, кузова, электрооборудования. Соответственно, большинство учебников для начинающих водителей, организационная структура проектирующих и производящих организаций, разделы проектных и эксплуатационных документов и т. д. структурированы в соответствии с этим делением.

В случае создания новаторской технической системы может использоваться принцип декомпозиции на основе особенностей имеющейся в распоряжении организационной структуры («организационный»), которая должна выполнить работу. Таким образом, первые экземпляры индивидуальных воздушных транспортных средств, скорее всего, повторяют структуру, характерную для автомобилей, если их разработкой займутся автоконцерны.

С инженерной точки зрения более обоснован принцип декомпозиции, основанный на приоритете повышения самоценности элементов с точки зрения выполняемых функций и обеспечения возможности их индивидуального развития и эволюции, т. е. на основе индивидуального жизненного цикла элементов как продуктов. Например, при разработке сложного программного комплекса при проектировании могут быть выделены элементы, которые могут в дальнейшем продаваться и эволюционировать в качестве отдельных продуктов.

Допустимо также проводить декомпозицию, основываясь на поддержании равного уровня сложности разделяемых элементов или минимизации взаимосвязей между ними. Эти подходы позволяют применять автоматизированные алгоритмы декомпозиции, например, при распределении электронных элементов на слоях печатных плат или микросхем.

Унификация. Предел детализации при анализе или проектировании выбирается исходя из достижения уровня, на котором составляющие подсистемы могут быть представлены типовыми унифицированными сущностями. Здесь проявляется третий важнейший подход к созданию или анализу сложных систем — унификация.

Сколько человеческих, финансовых и временных ресурсов потребовало бы проектирование такой сложной системы, как самолет, если бы каждую используемую в конструкции гайку проектировали индивидуально, а не использовали промышленно производимую стандартную из выбранной для данной конструкции ограниченной номенклатуры? А при выпуске очередного поколения, обычно усложняющего конструкцию на порядок? Если мы проектируем банковскую систему, пользователями которой являются тысячи человек, использующих различные компьютеры и каналы связи? Если проектируем жилое здание, не используя типовых элементов (например, кирпичей)?

Унификация архитектурных элементов (стандарт кирпича) в свое время обеспечила возможность создания высотных зданий.

В случае разработки автоматизированной банковской системы многотысячные пользователи приводятся к небольшому набору групп, что значительно упрощает проектирование.

Для случая механических систем строго отслеживается соблюдение стандартов на номенклатуру используемых типовых элементов.

В информационных технологиях применение унификации является не только обязательным, но и, возможно, наиболее ярко выраженным. В настоящее время большинство компьютеров взаимодействуют между собой на основе стандартного *IP*-протокола, предпринимаются постоянные усилия по приведению всех вычислительных систем в соответствие стандартизированной модели *Open System Interconnection*, идет систематическая работа по стандартизации форматов электронных документов и файлов хранения мультимедийных данных и т. д.

Визуализация. Эффективным инструментом анализа и проектирования сложных систем является визуализация — прием, который

позволяет более активно подключить к работе правую половину мозга проектировщика-аналитика, отвечающую за интуитивное и образное мышление. При этом необходимо принимать во внимание то, что потенциально этот тип мышления является гораздо более мощным и выносливым по сравнению с логическим. Некоторые авторы в области психологии и когнитивных способностей сравнивают логический тип мышления и образно-интуитивный с мальчиком — наездником на слоне. Наездник-логик — маленький и слабый. Его задача — направлять, выстраивать последовательности и решать сложные задачи. Образно-интуитивный слон — вынослив и силен. С его помощью легко решаются массовые однотипные задачи, происходит поиск аналогий и распознавание образов.

Реализуется прием визуализации не только путем использования графического изображения материальных объектов различного уровня детализации, но также за счет разработки схем различного типа. Причем анализ разрабатываемых абстрактных схем может производиться и формальными математическими методами в рамках теории графов; таким образом обеспечивается совмещение образного и логического подхода.

Графы. Под графом понимают множество так называемых «вершин» или «узлов», соединенных «дугами», или «ребрами» (рис. 2.1). Количество вершин графа определяет его «порядок», количество ребер — «размер» графа. Вершина графа, не имеющая ребер, называется «висящей», или «изолированной». Если ребра имеют направление, соединяют «начальную» и «конечную» вершины, то такой граф называют «ориентированным». В ориентированном графе существуют «маршруты» — последовательности вершин и соединяющих ребер с учетом направления и «пути» — маршруты без повторяющихся ребер. Выделяют «сильно связанные графы», в которых вершины взаимно достижимы с помощью путей. Математика графов широко применяется в алгоритмах анализа, разработки и оптимизации сложных систем, например, в задачах проектирования печатных плат и микросхем, оптимального построения бизнес- и технологических процессов.

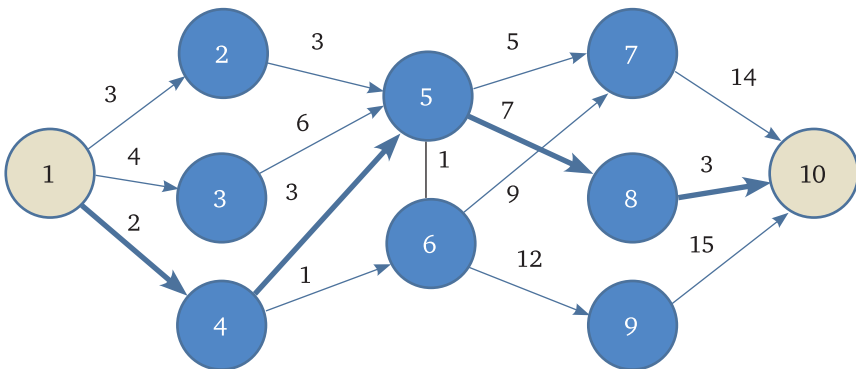


Рис. 2.1. Пример изображения графа

С помощью теории графов разрабатываемое визуальное представление подвергается логическому анализу. Именно этой задачей определяется используемая визуальная нотация, которая является наиболее усеченной и стандартизированной среди всех схематичных изображений. Остальные схематичные представления с помощью простого преобразования могут быть приведены к виду графа или им являются непосредственно.

Схемы. В практике проектирования применяют большое количество различных типов схем, визуально более приближенных к описываемым системам или процессам, чем абстрактные графы. Если ставится задача их дальнейшей алгоритмической обработки, они в большинстве случаев могут быть приведены к формализованному граф-описанию, но часто это не является обязательным.

К основным в системном анализе и проектировании относят функциональные и структурные схемы.

Структурные схемы содержат информацию об элементах системы и наличии между ними взаимосвязи без указания функций связей и элементов. Задача этих схем — дать информацию о структуре системы на верхнем уровне детализации и послужить основой для первичного включения в работу интуитивно-визуального мышления проектировщика.

Функциональные схемы несут более детализированную информацию. Их задача — дать информацию о функциях связей и элементов. Они в большей степени, чем структурные, являются носителями справочной и ссылочной информации о системе.

Диаграммы процессов IDEF. На практике используется большое количество стандартных схем, иллюстрирующих протекание процессов. К таким стандартам относятся, например, *IDEFO* — диаграммы, которые позволяют на стандартном широко применяемом графическом языке описывать последовательность выполнения операций (англ. *activity box*) с дополнительными информационными элементами (дугами), обозначающими обрабатываемые объекты и ориентированными по отношению к функциональному блоку следующим образом (рис. 2.2):

- верхняя сторона содержит информацию о значимом «Управлении» (англ. *Control*), в роли которого может выступать, например, технологическая инструкция;
- левая сторона имеет значение «Вход» (англ. *Input*); например, при описании изготовления какой-либо детали токарем будет содержать упоминание заготовки;
- правая сторона имеет значение «Выход» (англ. *Output*), т. е. результат операции;
- нижняя сторона имеет значение «Механизм» (англ. *Mechanism*), который в вышеприведенном случае должен будет именоваться как «токарь».

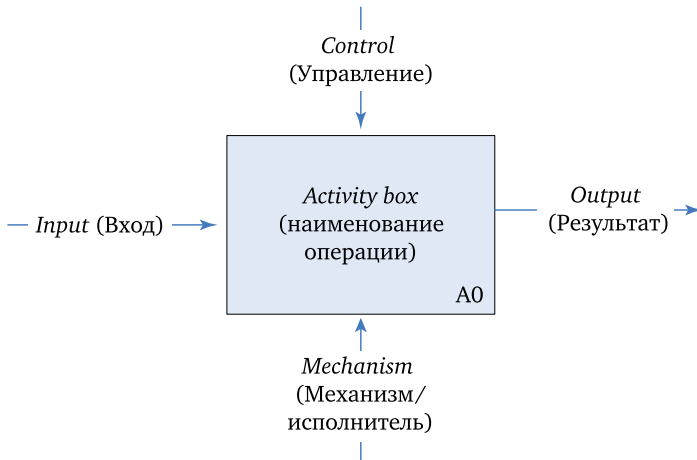


Рис. 2.2. Изображение элемента IDEF

Стандарт описания процессов *IDEFO* является развитием описания систем *SADT* (от англ. *structured analysis and design technique*). Первая версия вышла в 1981 г. в рамках инициированной ВВС США программы автоматизации промышленных предприятий *ICAM* (от англ. *integrated computer aided manufacturing*).

Расшифровка стандартов *IDEF* — *ICAM DEFinition*. Обновленные редакции стандарта выпускаются Национальным институтом по стандартам и технологиям США (*NIST*).

Нотация *IDEFO* предназначена для функционального описания процессов. При этом существуют дополнительные нотации: для описания внутренних информационных потоков в системе (*IDEF1*), для детализации реляционной структуры данных (*IDEF1X*), процессов развития систем (*IDEF2*), для документирования техпроцессов (*IDEF3*), для объектно-ориентированного проектирования (*IDEF4*), для описания состава и функционирования систем (*IDEF5*) и т.д.

Несмотря на широкое применение, стандарты *IDEF* являются далеко не единственными.

Диаграммы Use Case. Для проектирования и анализа автоматизированных систем на этапе первичного описания пользователем для передачи специалисту-программисту эффективным является применение use-case диаграмм (переводят как «диаграмма прецедентов» или «диаграмма вариантов использования»).

Этот стандарт описания является частью универсального языка моделирования (*UML*).

В общем виде диаграмма задает типы пользователей автоматизированной системы в виде так называемых акторов¹ (англ. *actor*), и связываемых с ними вариантов использования системы — *use case* (рис. 2.3).

¹ Часто встречающийся перевод с английского как «актер» не является удачным. Скорее по смыслу подходит «деятель».

В качестве актора по отношению к описываемой может выступать сторонняя автоматизированная система.

Под вариантом использования понимается то, что система делает для актора без приведения описания, как она это делает. Данное описание служит для отображения информации о вариантах диалога пользователей с автоматизированной системой и в данном случае применяется прием, который выше был описан как «черный ящик».

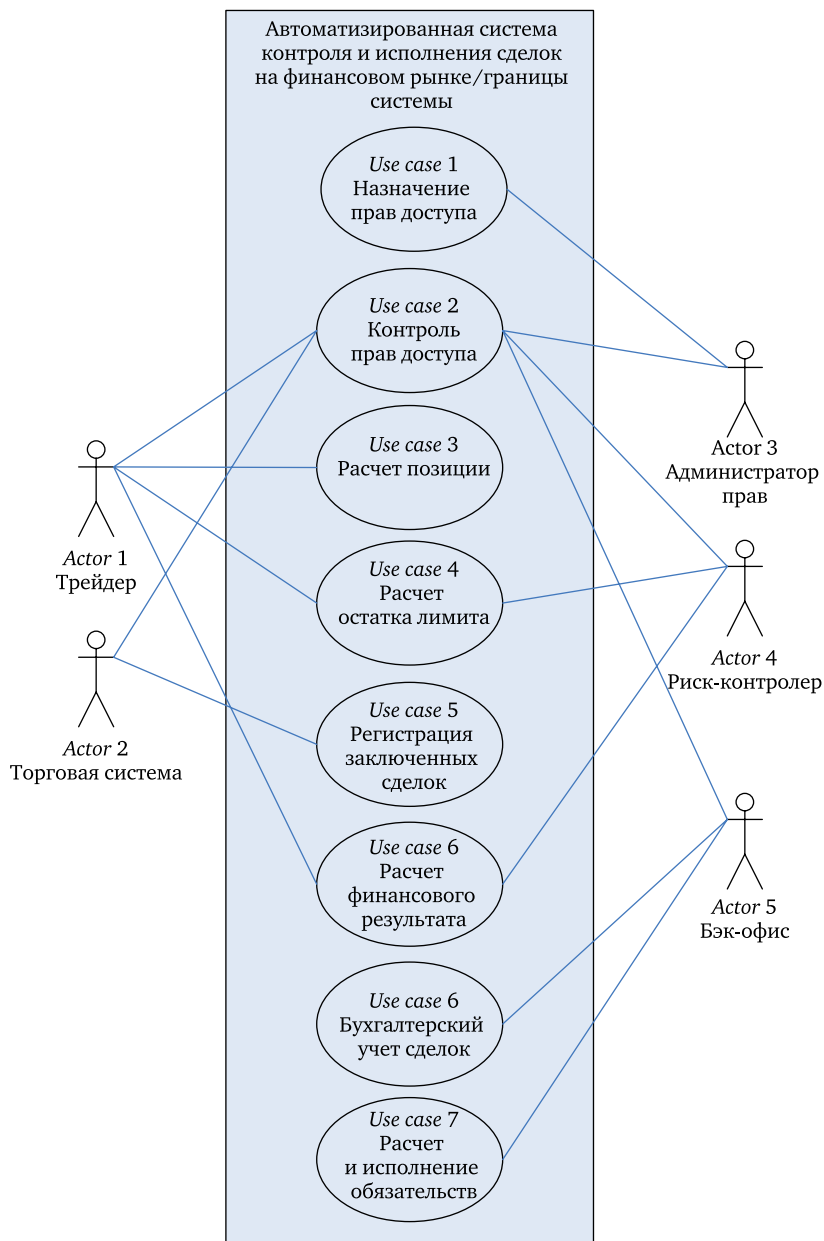


Рис. 2.3. Пример изображения схемы use case

Use case-представление обладает рядом очевидных преимуществ:

- устанавливается четкая граница моделируемой системы. Это позволяет защититься на начальном этапе проектирования от включения в требования излишних функций;

- описание производится на языке пользователя с выделением конкретных типов пользователей. В результате значительно снижается риск ошибочной формулировки требований пользователя. Согласование объемных проектных документов может производиться с отдельными подразделениями в контексте выделенного типа пользователя, что значительно облегчает данную процедуру и подстраховывает от ошибок;

- создание классификатора пользователей и задействованных сервисов системы позволяет на ранних этапах спроектировать подсистему разграничения прав доступа и проверить полноту требований с учетом списочного состава физических лиц, входящих в каждый тип актора, который в данном контексте представляет собой группу доступа;

- визуализация диаграммы позволяет выявить дублирующиеся типы пользователей и изолированные элементы, которые, очевидно, могут быть безболезненно удалены из проектных документов, либо демонстрируют недостаток описания — присутствие не связанного *use case* говорит или о том, что задана излишняя функция, либо о том, что потерян тип пользователя, к которому он должен быть прикреплен;

- являясь частью последовательной технологии описания *UML*, диаграмма позволяет обеспечивать последовательную многоэтапную детализацию описания с поддержкой целостности отражаемой графической информации на различных этапах: *use case*, диаграмма переходов (при реализации *use-case* внутри системы), диаграмма классов и т. д.¹ Другими словами, данный вид графического описания поддерживает проектирование сверху вниз и обеспечивает сохранение уровня детализации на соответствующем этапе проектирования.

Диаграммы BPM (от англ. *business process modelling*). Часто при проектировании автоматизированных систем используется процессный подход. Этот подход предполагает, что процесс — это заданная последовательность операций, выполняемых процессными ролями, имеющая вход, результат, временные и стоимостные характеристики.

Например, процессное описание используется для построения так называемых банковских кредитных конвейеров. Основными задачами при построении подобных человеко-машинных систем является сокращение до минимума времени ответа по запрошенному физическим лицом кредиту. При этом рассмотрение кредитной заявки осуществляется частично автоматически на основе заполненной клиентом анкеты, а частично соответствующими специалистами, представляющими ряд разнородных подразделений банка: безопасность, юристы, риск-менеджмент и др.

¹ Полное описание языка *UML* и стандартов *IDEF* не является предметом данной книги.

Для построения эффективно работающего конвейера в этом случае необходимо не только нормировать время обработки кредитной заявки всеми участниками процесса, распараллелить все доступные для этой операции, но и обеспечить оценку нагрузки на всех участников, а также оценку стоимости всех этапов работы — процесс не должен стоить дороже получаемого дохода, суммарное время обработки не должно превышать целевого, которое часто оказывается заданным со стороны конкурентного рынка.

Для решения подобных задач международный некоммерческий консорциум *Object Management Group*, организованный в 1989 г. крупнейшими производителями программного обеспечения, принял в 2006 г. нотацию описания бизнес-процессов *Business Process Model and Notation (BPMN)*.

Нотация включает основные элементы, представленные на рис. 2.4.

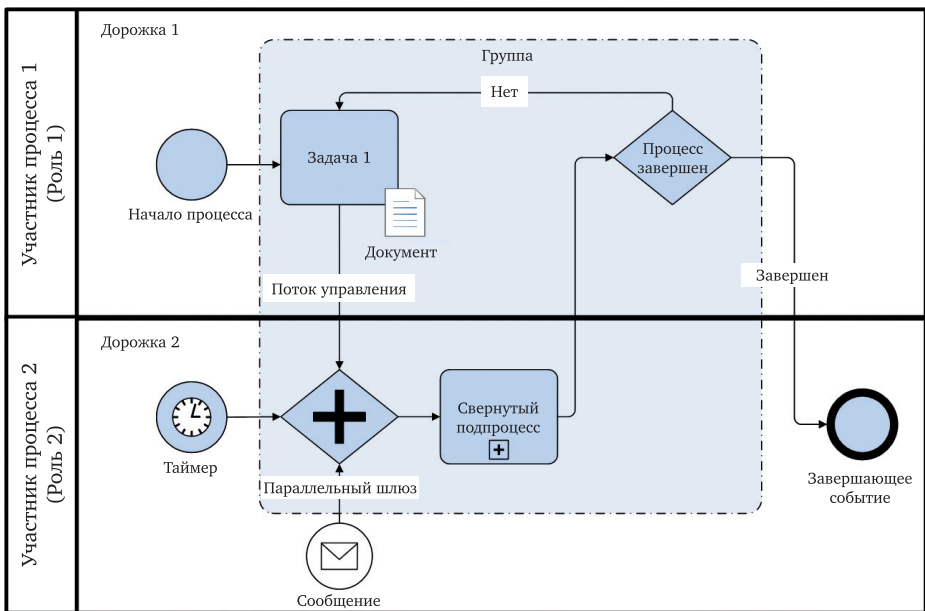


Рис. 2.4. Пример изображения схемы BPM

События изображаются на схеме окружностью и означают какое-либо происшествие в мире. События инициируют действия или являются их результатами. Основные типы событий:

- простые события — начало процесса, окончание процесса, промежуточное событие и др.;
- события-таймеры — элементы, позволяющие указать на схеме регулярные события, периоды времени и др.;
- события-сообщения — элементы, позволяющие указать на входящее для процесса сообщение.

Действия изображаются на схеме прямоугольниками со скругленными углами. Могут содержать информацию о дополнительных харак-

теристиках, например, по времени и стоимости выполнения. Среди действий различают задания и подпроцессы. Графическое изображение свернутого подпроцесса снабжено знаком плюс у нижней границы прямоугольника.

Задание (англ. *task*) — это единица работы, элементарное действие в процессе.

Свернутый подпроцесс (англ. *collapsed subprocess*) является составным действием, но скрывает детали реализации процесса.

Логические операторы изображаются ромбами и представляют точки принятия решений в процессе. С помощью логических операторов организуется ветвление и синхронизация потоков управления в модели процесса:

- оператор исключающего «или», управляемый данными (англ. *data-based exclusive gateway*). Если оператор используется для ветвления, то поток управления направляется лишь по одной исходящей ветви. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток;

- оператор исключающего «или», управляемый событиями (англ. *event-based exclusive gateway*), направляет поток управления лишь по той исходящей ветви, на которой первой произошло событие. После оператора данного типа могут следовать только события или действия-обработчики сообщений;

- оператор включающего «или» (англ. *inclusive gateway*) активирует одну или более исходящих ветвей, в случае, когда осуществляется ветвление. Если оператор используется для синхронизации, то он ожидает завершения выполнения одной входящей ветви и активирует выходной поток;

- оператор «и» (англ. *parallel gateway*), использующийся для ветвления, разделяет один поток управления на несколько параллельных. При этом все исходящие ветви активируются одновременно. Если оператор используется для синхронизации, то он ожидает завершения выполнения всех входящих ветвей и лишь затем активирует выходной поток.

Соединяющие объекты служат для связи между собой объектов управления. Отображаются на схеме линиями различного типа со стрелками на одном конце:

- поток управления изображается сплошной линией. Поток управления задает порядок выполнения действий;

- поток сообщений — изображается штриховой линией, показывает, какими сообщениями обмениваются участники процесса;

- ассоциации изображаются пунктирной линией. Используются для отображения связи артефактов, данных или текстовых аннотаций с объектами потока управления.

Роли — являются основными идентифицирующими элементами дорожек. Служат для указания участников процесса, за которыми закреплены наборы действий. Дорожки, в свою очередь, служат для визуаль-

ной группировки последовательности действий, наборов артефактов и других объектов, закрепленных за ролями. Дорожки могут группироваться в пулы.

Артефакты служат для отображения в схеме дополнительной информации:

- данные показывают, например, какие документы необходимы действиям для выполнения и какие документы производятся. Могут отображаться также в виде элемента базы данных;
- группа изображается в виде штрихового прямоугольника с закругленными углами;
- группа позволяет визуально объединять действия, но не влияет на поток управления в диаграмме;
- текстовые аннотации используются для уточнения значения элементов диаграммы и повышения удобства ее использования.

Кроме того, *ВРМН* включает ряд вспомогательных элементов, позволяющих отобразить на схеме дополнительную ценную информацию.

Общие правила графического языка. При использовании визуального языка необходимо принимать во внимание два основных фактора:

1) с образами работает половинка мозга, предназначенная для типовых задач, и ее работу достаточно просто затруднить, применяя нестандартные или усложненные элементы. Именно поэтому на любую образную нотацию накладывают жесткие ограничения по размерам, типам элементов, цвету и т. д. Любая схема, заимствующая сторонние элементы разнородных нотаций или элементов, непривычных для участников процессов анализа (проектирования), является принципиально некорректной;

2) каждый вид схем и используемых визуальных нотаций имеет свое назначение и решает свои задачи. Не существует универсального визуального языка, решающего все задачи инженера. Совмещение разнонаправленных по задачам отображений на единой схеме часто приводит к ошибкам и исключает эффективное использование.

Графическое схематичное изображение разрабатывается не только с целью повышения эффективности деятельности аналитика-проектировщика или математической обработки, но также для передачи значимой информации взаимодействующему специалисту или коллективу.

Учитывая то, что реальные сложные системы практически никогда не разрабатываются и не анализируются отдельными специалистами, последнюю из приведенных целей следует считать приоритетной. Схемы разрабатываются в основном для их прочтения взаимодействующими специалистами. Это накладывает жесткие требования к используемому графическому языку и соглашениям изображений.

В данном случае с полным основанием применим принцип «глупец — не тот, кто не понимает, а тот, кого не понимают». Другими словами, ответственность за обеспечение однозначного и эффективного прочтения схем, так же как и другой документации, лежит на их разработчике.

При подготовке схем следует обращать внимание, например, даже на то, что в европейской культуре принято прочтение документов слева направо и сверху вниз. Это значит, что для облегчения прочтения потоки данных, направление исполнения процессов и т. д. желательно располагать слева направо и сверху вниз. Иное расположение затруднит работу взаимодействующих аналитиков.

Также затрудняет прочтение приведение на одной схеме более 8—10 объектов, использование мелкого шрифта, перекрестных ссылок между страницами документов и т. д.

Глава 3

ЭКОНОМИЧЕСКАЯ ТЕОРИЯ ИНСТИТУЦИОНАЛИЗМА

Современное состояние российского хозяйства характеризуется недостаточным уровнем обучения инженерных специалистов экономике и управлению. Это приводит к многочисленным попыткам передать ведущие управленческие позиции в высокотехнологичных областях так называемым «профессиональным менеджерам», большинство из которых не обладают необходимыми инженерными знаниями и системным мышлением, что отрицательно сказывается на эффективности деятельности сложных технологичных производств.

Между тем в развитых экономиках достаточно давно сформирован ряд направлений развития экономической и инженерной мысли, стирающих грань между управлением технологиями и экономикой на основе процессного подхода и понятии сложных систем. Изучение этих подходов позволяет повысить эффективность работы инженеров и неинженерных специалистов в качестве руководителей производства.

Одним из наиболее прогрессивных подходов в современной экономической теории является **институционализм** (и неоинституционализм). Первичные положения данного направления были сформулированы в начале XX в. американскими экономистами Т. Вебленом (1857—1929), Д. Коммонсом (1862—1945), У. Митчеллом (1884—1948).

В своей основе институционализм имеет следующие понятия: «институты» — нормы, обычаи поведения в обществе, «институты» — закрепление норм и обычаев в виде законов, организаций, государственных учреждений, союзов и т. д.

Основным отличием институционализма от других направлений является попытка внедрения системного процессного подхода в экономической теории, что представляет ценность для специалистов в области информационных технологий, участвующих в укрупненном анализе хозяйственных процессов и структур или рассматривающих возможность заимствования инструментария экономической теории для применения в проектной или научной деятельности.

В еще большей степени обоснованным является обратное заимствование наработок информационно-технологической сферы и теории качества в анализе и развитии хозяйственных структур.

Институционализм как теоретический подход имеет следующие отличительные черты:

- экономические категории — цена, прибыль, спрос — рассматриваются в качестве характеристик взаимодействия систем, образующих хозяйство;

- экономика исследуется в качестве социальной подсистемы, наряду с политикой, наукой и культурой;

- хозяйствующие субъекты рассматриваются как сложные системы, реализующие набор внутренних процессов, процессов взаимодействия и характеризующиеся изменяющимися состояниями, а не абстрактные модели, стремящиеся к максимизации какого-либо показателя эффективности, чаще финансового;

- предпочтение отдается традиционному для технической инженерии подходу к анализу — сверху вниз. Эффективность деятельности хозяйственных субъектов определяется эффективностью их взаимодействия в рамках надсистемы (хозяйства) и эффективностью этой системы в целом (в классической экономической теории считают, что цели субъекта являются источником направления деятельности хозяйства в целом);

- государство рассматривается в качестве необходимой управляющей системы;

- индивид является сложной системой, с изменяющимися целями и поведением, не оптимальным и непредсказуемым в связи со сложностью анализа влияющих факторов;

- экономика декларируется как эволюционирующая система, управляемая процессами и институтами, находящимися в постоянной взаимной адаптации;

- задача экономической науки определяется не столько в том, чтобы составить прогноз, а в том, чтобы сформировать исчерпывающий и эффективный набор мер по развитию экономики, политики и сознания, образующих единую систему;

- уровень развития экономической системы связывается в первую очередь с уровнем развития (эффективности) институтов, с размером транзакционных издержек и уровнем оппортунистического (необоснованно конфликтного) поведения.

Последний тезис представляет особый интерес, так как перекликается с подходами, получившими широкое распространение в настоящее время в области информационных технологий и проектного управления, которые базируются на теории качества и модели зрелости процессов и организаций¹.

¹ Подробнее о теории институционализма см.: Аузан А. Экономика всего. Как институты определяют нашу жизнь. М. : Манн, Иванов и Фербер, 2014.

Глава 4

ТЕОРИЯ ОБЩЕГО УПРАВЛЕНИЯ КАЧЕСТВОМ. СТАНДАРТ ISO 9000 — УПРАВЛЕНИЕ КАЧЕСТВОМ ПРОЦЕССОВ

4.1. Контрольные карты и цикл постоянного улучшения

Продукты деятельности ИТ-индустрии, например, программное обеспечение, автоматизированные комплексы, микропроцессоры, системы искусственного интеллекта, справедливо относят к категории не только самых высокотехнологичных, но и наиболее сложных систем. Их качество и стабильность работы во многом определяют деятельность производственных предприятий, государственных и финансовых институтов. Это стало основанием для выхода ИТ-индустрии на передовые позиции в разработке подходов по управлению качеством и их практическом применении. Уровень проникновения передовых подходов по управлению качеством в производство автоматизированных систем в настоящее время создает блокирующий барьер для выхода на рынок производителей, не обладающих необходимой компетенцией в данных вопросах.

К началу XX в. поточное производство, основанное на тейлоризме и являющееся основным следствием промышленной революции, прочно укоренилось в экономике развитых стран. Качество продукции перестало рассматриваться как результат персональных усилий или умений. На первое место вышел анализ производственных систем, процессов и их отклонений, влияющих на результаты производства.

Первооткрывателями и основными идеологами в данной области технологических знаний по настоящее время считаются Уолтер Эндрю Шухарт (*Walter Andrew Shewhart*, 1891—1967) и Уильям Эдвардс Деминг (*William Edwards Deming*, 1900—1993).

Начиная с 1931 до 1939 г. Шухарт издал ряд статей и книг (например, «Статистический метод с точки зрения контроля качества») по вопросам управления качеством промышленной продукции. В этих работах он ввел понятие контрольных карт, которые теперь называют контрольными картами Шухарта (иногда — контрольными картами Деминга), а также понятие цикла постоянного улучшения, обосновал системный

и статистический подходы к управлению производством, которые легли в основу разработки концепции «Шесть сигма» (6σ).

В основе концепции контрольных карт лежит понимание технологического процесса как случайного стационарного процесса (имеющего устойчивое среднее или среднестатистическое состояние со случайными предсказуемыми отклонениями), подверженного воздействию дестабилизирующих факторов, которые должны быть исключены в результате деятельности технолога.

Таким образом, задача сводится к тому, чтобы привести технологический процесс в стабильное состояние (добиться управляемого состояния и изолировать от внешнего негативного воздействия), а затем путем дополнительной подстройки параметров процесса обеспечить статистически количественно допустимый уровень качества результата его работы.

Например, для случая токарной обработки некоторой детали параметры токарного станка и погрешности в работе квалифицированного мастера являются внутренними для процесса факторами, а брак в поступившей заготовке или изменения в освещенности рабочего поля, приводящие к браку, — внешним дестабилизирующим фактором. Интересно, что таким же внешним дестабилизирующим фактором будет в этом случае и недостаточная квалификация или болезненное состояние токаря.

Для случая ИТ-технологий, например, процесса сопровождения автоматизированной системы, таким стационарным случайным процессом может быть количество сбоев автоматизированных систем в заданный период времени (день, неделя, месяц и т. д.). Резкое изменение количества сбоев за период в этом случае потребует анализа с целью выявления внешнего фактора, влияющего на статистику. Чаще всего это может быть изменение персонального состава сопровождающего подразделения или включение вычислительных элементов с низкой надежностью.

Сами по себе карты Шухарта при этом являются средством графического мониторинга и документирования интересующих характеристик производственного процесса с привязкой к временной шкале, что позволяет проводить постконтроль процесса для выявления периода воздействия дестабилизирующих факторов и их анализа.

На картах отмечаются «*CL* — центральная линия», «*LCL* — нижняя контрольная граница», «*UCL* — верхняя контрольная граница», которые по результатам накопленной статистики работы процесса принимаются в качестве характеристик стабильного процесса с допустимым уровнем отклонений (рис. 4.1).

Выход количественной оценки за границы считается сигналом о необходимости анализа дестабилизирующих факторов, которые к этому привели. В ряде случаев таким сигналом считается также наличие нескольких последовательных точек над или под центральной линией.

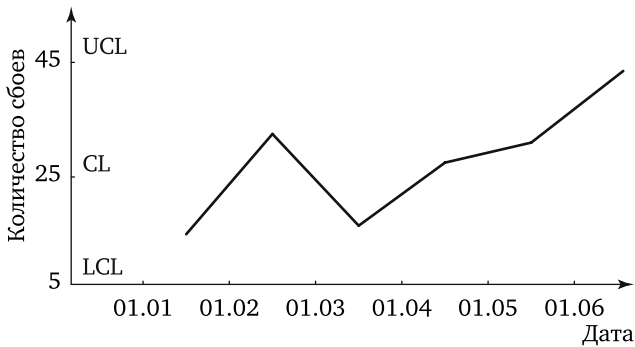


Рис. 4.1. Пример контрольной карты Шухарта для случая процесса сопровождения автоматизированной системы

С именами Деминга и Шухарта связана одна из важнейших концепций в теории качества — цикл постоянного улучшения. Оптимальное качество сложного продукта недостижимо в результате единовременного проекта.

На практике невозможно разработать идеальные требования к продукту, а затем реализовать их: вмешиваются факторы, связанные с невозможностью полностью контролировать сложные системы. Поэтому качественный продукт является результатом непрерывного развития.

Работа по развитию качества не должна прерываться. В интерпретации Деминга, окончательно сформулировавшего цикл развития, он выглядит следующим образом: *plan, do, check, act* (PDCA) или «планируй, выполняй, проверяй, действуй» (рис. 4.2).

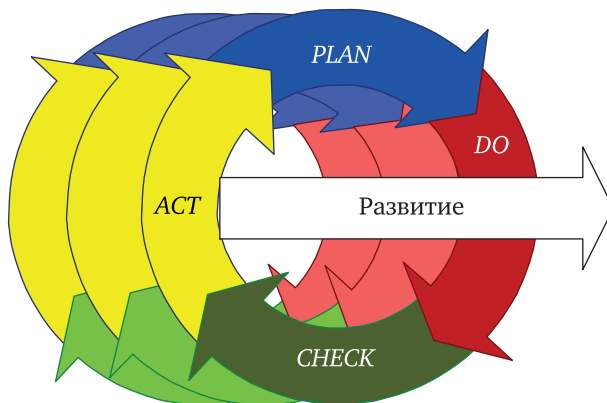


Рис. 4.2. Цикл Шухарта — Деминга

На первом этапе — «планируй» — планируются изменения и возможные меры, направленные на улучшение и рост качества, включая их испытания.

На втором этапе — «выполняй» — выполняются запланированные мероприятия улучшения в масштабе, достаточном для получения

полезной информации, но не больше, чем потребуется в том случае, если дела пойдут неудачно.

На третьем шаге — «проверяй» — проверяются и анализируются результаты, полученные на этапе «выполни».

На четвертом этапе — «воздействуй» — в зависимости от результатов анализа, выполненного на предыдущем этапе, внедряются проверенные ранее мероприятия в укрупненном масштабе.

После четвертого этапа — «воздействуй» — следует повтор цикла с использованием полученного опыта и в связи с изменившимися условиями реального мира.

Управление организацией и качеством является непрерывным процессом, повторяющим фазы задания измеряемых целей, планирования действий по улучшению, их выполнения и анализа результатов с последующей корректировкой.

4.2. Закон Парето

Еще одна ключевая концепция теории качества связывается с именем Вильфредо Парето. Закон Парето, имеющий скорее эмпирический характер, чем теоретический, звучит следующим образом: 20 % труда реализуют 80 % результата, остальные 20 % результата требуют 80 % общих затрат. С точки зрения теории качества закон говорит о том, что с помощью 20 % усилий можно устранить до 80 % дефектов.

Для визуализации анализа строят так называемую диаграмму Парето (рис. 4.3), которая позволяет сформулировать максимально выигрышную стратегию устранения дефектов.

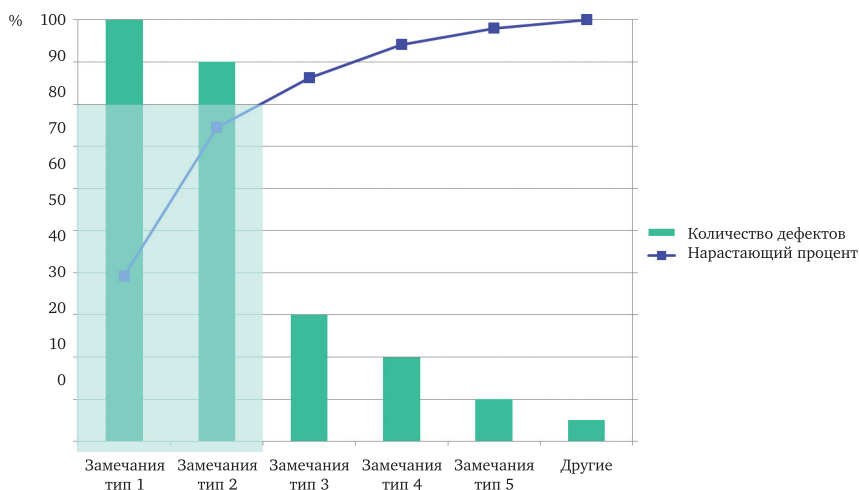


Рис. 4.3. Пример диаграммы Парето

Из диаграммы видно, что типы дефектов сгруппированы по количеству от большего к меньшему. Сопровождающая кривая демонстри-

рует накопленный суммарный процент. Точка пересечения этой кривой с отметкой 80 %, таким образом, ограничивает справа набор первоочередных мер по устранению дефектов. Исходя из закона Парето, предполагается, что этот результат будет достигнут за счет 20 % от общих затрат на устранение всего массива дефектов.

4.3. Теория общего управления качеством

Вышеприведенные подходы постоянного улучшения и статистического контроля процессов Шухарта — Деминга нашли отражение в целостной концепции общего управления качеством (англ. *total quality management, TQM*). Данная концепция обычно сводится к нескольким основным принципам:

- ориентация на потребителя;
- лидерство руководителей;
- вовлечение работников;
- процессный подход;
- системный подход к менеджменту;
- постоянное улучшение;
- принятие решений, основанное на фактах;
- взаимовыгодные отношения с поставщиками.

Исторически концепция *TQM* стала преемником раннего тейлоризма с его механистическим подходом к человеку и значительно повысила значение человеческой составляющей компании в решении вопросов качества. Обученный коллектив, эффективно реализующий процессы производства, был объявлен основной ценностью.

Системность концепции обеспечивается включением в решение задачи оптимизации качества всех заинтересованных участников, включая акционеров, руководство, потребителей, поставщиков и сотрудников.

Процессный подход получает преимущество перед продуктовым. Приоритет смещается от качества продукции в сторону управления качеством процессов, что является более обоснованным в условиях удлинения технологических цепочек и затруднительности в ряде случаев контроля промежуточного продукта.

4.4. Стандарт *ISO 9000*

Дополнительный толчок расширению применимости модели общего управления качеством и ее институционализации¹ был придан в результате анализа взаимодействия стран-союзников в ходе Второй мировой войны. Проведенный по итогам войны анализ выявил

¹ **Институционализация** — внедрение, полезное применение на практике, регламентация, регулирование.

значительные недостатки и несоответствия в организации военного производства и подходах к обеспечению качества продукции, которые снизили эффективность совместных боевых действий и взаимного снабжения союзников.

Итогом работы по переосмыслению полученного опыта стало создание в 1946 г. Международной организации по стандартизации (англ. *International Organization for Standardization, ISO*).

Создание *ISO* было инициировано на встрече делегатов 25 стран в Институте гражданских инженеров (англ. *Institute of Civil Engineers*) в Лондоне, где было решено создать новую международную организацию по индустриальной кооперации и стандартизации.

Новая организация начала работу в феврале 1947 г. Представители СССР принимали участие в иницилирующей встрече. За период работы организации *ISO* и членства в ней СССР руководитель Госстандарта СССР дважды становился ее председателем.

К настоящему времени *ISO* опубликовала 19 500 международных стандартов, охватывающих практически все разделы технологий и производства. В организацию входит 165 стран. Центральный секретариат *ISO* расположен в Женеве.

Интересен факт, касающийся наименования организации. Первоначальное название *IOS* (*International Organization for Standardization*) было изменено на *ISO* за счет изменения порядка следования слов для того, чтобы дополнительно указать на равенство участников и цели организации за счет аналогии с греческим словом *isos* — «равный».

В соответствии с уставом *ISO* целью этой организации является «содействие развитию стандартизации в мировом масштабе для облегчения международного товарообмена и взаимопомощи, а также для расширения сотрудничества в области интеллектуальной, научной, технической и экономической деятельности».

Работа *ISO* обеспечивается 238 техническими комитетами по различным отраслям промышленности и другим направлениям. Нумерация комитетов соответствует последовательности их создания.

Наибольшее количество стандартов (2766) опубликовано комитетом «Информационные технологии — *ISO/IEC JTC1*» — совместным комитетом *ISO* и Международной электротехнической комиссии. Следующим по количеству опубликованных стандартов является комитет «Пищевые продукты» (*ISO/TC34*) — 819 стандартов.

Комитет *ISO/TC176* «Управление качеством и обеспечение качества» опубликовал 24 стандарта. Несмотря на небольшой объем деятельности этого комитета, именно она в итоге привлекла к *ISO* широкое международное внимание. В настоящее время стандарты *ISO* являются наиболее распространенными в международной практике.

Членами комитетов *ISO* являются национальные организации по стандартизации. Проекты согласовываются участниками и утверждаются, если набирают не менее 75 % голосов.

Большинство развитых стран мира имеют свои национальные институты и системы стандартизации: *BSI* (англ. *British Standards Institution*, создан в 1901 г.) — в Великобритании, *NIST* (англ. *The National Institute of Standards and Technology*, 1901 г.) — в США, *DIN* (нем. *Deutsches Institut für Normung*, 1917 г.) — в Германии, *AFNOR* (*Association Francaise de Normalisation*, 1926 г.) — во Франции, ГОСТ (Государственные стандарты, 1925 г.) — в России. При этом в середине XX в. было опубликовано множество отраслевых стандартов, однотипная продукция изготавливалась для различных заказчиков по различным стандартам, аудиты проводились по противоречащим друг другу требованиям. Эта ситуация была общей для технологически быстро развивающихся стран.

Вторая мировая война выявила недостатки сложившейся системы и Министерством обороны США в европейских странах была опубликована серия документов НАТО по обеспечению и стандартизации качества — *AQAP* (англ. *NATO Allied Quality Assurance Publications*). Эти стандарты основываются на том, что обороноспособность зависит в значительной степени от качества оборонной продукции (систем вооружения), включая комплекс технических средств, программного обеспечения, средств обслуживания, персонала и процессов, а качество продукции наилучшим образом может быть достигнуто путем применения комплексного системного подхода к качеству на всех стадиях ее жизненного цикла.

В 1972 г. Британский институт стандартов сделал попытку стандартизировать технические условия по обеспечению качества в различных отраслях промышленности и разработал стандарт «Руководство по обеспечению качества» *BS4891* (англ. *A guide to quality assurance*).

Публикация данного стандарта инициировала продолжение работ по детализации. В 1975 г. был издан стандарт *BS5179*, который позволил создавать общепринятые системы выбора поставщиков. В 1979 г. был опубликован государственный стандарт Великобритании в области обеспечения качества — *BS5750*. Этот стандарт в 1987 г. стал основой для разработки Техническим комитетом *ISO TC176* группы стандартов *ISO* серии 9000.

Необходимо отметить большой вклад в работу по стандартизации и сертификации в Великобритании, который внесла Маргарет Тэтчер. Практически все эти работы велись по ее прямому указанию и под ее кураторством.

Основой стандартов *ISO 9000* стал комплексный подход к управлению качеством — всеобщее управление качеством, *TQM*.

Серия стандартов *ISO 9000:87* включала в себя три основных блока:

- *ISO 9001:87* — Модель для обеспечения качества при проектировании и (или) разработке, производстве, монтаже и обслуживании;
- *ISO 9002:87* — Модель для обеспечения качества при производстве, монтаже и обслуживании;

- *ISO 9003:87* — Модель для обеспечения качества при контроле и испытаниях готовой продукции.

В настоящее время действует обновленная версия стандартов *ISO 9000*, которая включает следующие основные документы:

- *ISO 9000:2005* — Системы менеджмента качества. Основные положения и словарь;

- *ISO 9001:2008* — Системы менеджмента качества. Требования (содержит минимальный набор требований для получения сертификата соответствия стандарту *ISO 9000*);

- *ISO 9004:2009* — Менеджмент с целью достижения устойчивого успеха организации. Подход с позиции менеджмента качества (содержит расширенные рекомендации по постоянному улучшению деятельности организации).

С 2001 г. в России действует серия отечественных стандартов ГОСТ Р ИСО 900х—2001, повторяющая стандарты *ISO 900х:2000*.

Необходимо отметить, что данный набор стандартов закрепляет термины «менеджмент», «менеджмент качества», «система менеджмента качества» и «всеобщий менеджмент качества».

Несмотря на то, что ГОСТ в этой части является переводом *ISO* и не имеет принципиальных отличий, отечественные сертифицирующие организации проводят свои сертификационные аудиты на соответствие ГОСТ Р ИСО 9001:2001, а зарубежные организации — свои, по *ISO 9001:2000*.

Различия заключаются в следующем. В России имеются законодательные решения о необходимости сертификации на соответствие ГОСТ Р ИСО 9001:2001 (в частности, в соответствии с Постановлением Правительства № 13 от 02.02.1998, наличие данного сертификата является одним из условий получения Госзаказа). При этом российский сертификат не действует за пределами России.

Для признания мировым сообществом необходимо сертифицироваться по *ISO 9001:2000* у соответствующего органа, например, *Bureau Veritas Quality International* (Франция), *Lloyd's Register Quality Assurance* (Великобритания), *Det Norske Veritas (DNV)*, Норвегия), *ABS Quality Evaluations* (США) и др.

В США и Западной Европе сертифицировано более 80 % компаний. В Китае число таких предприятий уже превысило отметку 40 %. Во всем мире (включая Россию) стандарт *ISO* серии 9000 применило более миллиона предприятий.

Проверочные тесты к разделу I

1. В соответствии с математическим определением информация — это:

- а) новости и архивные документы;
- б) данные для проведения анализа, электронные документы, обрабатываемые ЭВМ;
- в) мера уменьшения энтропии (непредсказуемости системы) при получении сообщения.

2. Основная миссия инженера по информационным технологиям:

- а) закупать вычислительные средства, обеспечивать их установку и эксплуатацию;
- б) упорядочивать и развивать системы управления;
- в) обеспечивать обработку информации.

3. В настоящее время в развитых странах относительно количество работников (от общего числа), занятых в сфере обработки информации:

- а) в связи с ростом производительности вычислительных машин — снижается;
- б) в связи с ростом потоков данных — растет;
- в) остается постоянным.

4. Какое из направлений не входит в список основных современных тенденций развития ИТ:

- а) развитие организации управления информационными технологиями в рамках предприятия;
- б) унификация вычислительных средств;
- в) рост разнообразия используемых форматов хранения информации?

5. Сложная система — это:

- а) набор связанных компонент с предсказуемым поведением;
- б) большое количество не связанных компонент с плохо предсказуемым поведением;
- в) набор связанных компонент, который может быть обособлен от окружения, обладает свойствами, возникающими при корректных связях, с плохо прогнозируемым поведением.

6. Закон Парето:

- а) позволяет выбирать приоритеты устранения дефектов;
- б) связывает спрос и предложение;
- в) позволяет проектировать сложные системы.

7. Институционализм:

- а) позволяет прогнозировать инфляцию;
- б) ориентируется на уровень потребления домашних хозяйств;
- в) связывает теорию сложных систем и экономическую теорию.

Раздел II

УПРАВЛЕНИЕ ПРОЕКТАМИ

В результате изучения данного раздела студент должен:

знать

- теоретические основы проектной деятельности как процесса взаимной адаптации;

- типовые ошибки проектного управления;

- основные положения базовых стандартов в области проектного управления;

уметь

- разрабатывать и согласовывать основные документы проектного управления;

- минимизировать риски проекта;

владеть

- гибкими и традиционными методиками управления проектами.

Глава 5

ПРОЕКТНАЯ ДЕЯТЕЛЬНОСТЬ

Человечество с давних времен занимается реализацией проектов в различных областях. Примерами могут служить строительство египетских пирамид, строительство Собора Василия Блаженного, советский или американский атомный проект, запуск первого космического корабля и многие другие.

5.1. Мировая статистика успешности ИТ-проектов *The Standish Group*

*The Standish Group*¹ определяет следующие три группы проектов по успешности (табл. 5.1):

- успешный проект (англ. *successful*) — проект реализован без превышения планового срока и бюджета. Требования к результатам проекта достигнуты;
- провальный проект (англ. *failed*) — проект не реализован, требования к его результатам не достигнуты;
- спорный проект (англ. *challenged*) — проект реализован частично, с превышением сроков, или бюджета.

Таблица 5.1

Статистика успешности по отдельным годам²

Проекты	Годы				
	2011	2012	2013	2014	2015
Успешные	29 %	27 %	31 %	28 %	29 %
Спорные	49 %	56 %	50 %	55 %	52 %
Провальные	22 %	17 %	19 %	17 %	19 %

По оценкам *The Standish Group* в 2013 г. во всем мире на ИТ-проекты было потрачено 750 млрд долл. Это около трети ВВП России, или 11 % от мирового валового продукта. На Соединенные Штаты Америки пришлось около 300 млрд долл., на Европу — 200 млрд долл., на Азию — 100 млрд долл., на остальную часть мира — 150 млрд долл.

¹ URL: <http://www.standishgroup.com>.

² URL: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>.

Таким образом, на полностью провальные проекты в 2013 г. пришлось около 150 млрд долл., а с учетом частично провальных (спорных) — еще дополнительно около 350 млрд долл. Итого цена вопроса с успешностью ИТ-проектов в мире составляет около 500 млрд долл. в год без учета утраченных возможностей, которые должны были принести дополнительный доход в результате реализации неуспешных проектов.

В рамках исследования¹ *The Standish Group* представила также несколько интересных дополнительных наблюдений. Основное то, что успешность сильно отличается для проектов различного масштаба, а также в зависимости от применяемого подхода (табл. 5.2).

Таблица 5.2

Успешность проекта в зависимости от их масштаба и применяемого подхода

Масштаб проекта (доход компании), млн долл.	Метод	Успешные	Спорные	Провальные
Все	<i>Agile</i>	39 %	52 %	9 %
	<i>Waterfall</i>	11 %	60 %	29 %
≈ 2,5 (от 0,5)	<i>Agile</i>	18 %	59 %	23 %
	<i>Waterfall</i>	3 %	55 %	42 %
≈ 1,3 (от 0,2 до 0,5)	<i>Agile</i>	27 %	62 %	11 %
	<i>Waterfall</i>	7 %	68 %	25 %
≈ 0,4 (до 0,2)	<i>Agile</i>	58 %	38 %	4 %
	<i>Waterfall</i>	44 %	45 %	11 %

Представленные данные позволяют сделать следующие выводы: крупные проекты (с объемом финансирования от 1 млн долл.) бывают успешными с вероятностью в несколько раз меньше, чем мелкие, бюджет которых не превышает полмиллиона долларов. Сложные проекты связаны с большим риском.

Применение *agile* резко повышает шансы любого проекта на успех и позволяет успешно справляться со сложностью проектов, выравнивая шансы на успех масштабных и мелких проектов.

Кроме статистики успешности, *The Standish Group* проанализировала основные факторы, повлиявшие на успешность или провал проектов (табл. 5.3). Данные собирались на основе экспертных оценок руководителей более чем 50 тыс. компаний (табл. 5.3).

В ходе анализа эффектов от применения *agile* подхода при сложных проектах авторами исследования отмечено, что в таких проектах быстрее обнаруживаются принципиальные ошибки, допущенные в ходе реализации, и они могут быть оперативно откорректированы.

¹ URL: <https://www.projectsmart.co.uk/white-papers/chaos-report.pdf>.

Факторы успеха проектов

Факторы успеха	Описание	Вклад в успешность проекта
Поддержка руководства (англ. <i>executive support</i>)	Финансовая и эмоциональная поддержка руководства	15 %
Зрелость поведения (англ. <i>emotional maturity</i>)	Высокий уровень взаимодействия в коллективе	15 %
Вовлеченность пользователей (англ. <i>user involvement</i>)	Пользователи принимают активное участие в принятии решений, сборе информации, создании эффективной обратной связи при согласовании требований, прототипировании; настроены на поиск компромиссов	15 %
Оптимизация (англ. <i>optimization</i>)	Организация реализует структурированный набор мер для повышения эффективности деятельности и взаимной согласованности отдельных проектов на основе бизнес-показателей	15 %
Компетентность персонала (англ. <i>skilled staff</i>)	Компетентность персонала в бизнес-области и в информационных технологиях, его способность создавать высококачественный продукт	10 %
Стандартизированная инфраструктура управления (англ. <i>standard architectural management environment</i>)	Развитая современная инфраструктура интегрированных процессов, технологий, сервисов для разработки, внедрения и эксплуатации программного обеспечения	8 %
Профессиональность <i>agile</i> (англ. <i>agile proficiency</i>)	Команда проекта и владелец продукта овладели методологией <i>agile</i> на высоком уровне и способны отличать хорошие и плохие результаты <i>agile</i>	7 %
Сдержанность исполнения (англ. <i>modest execution</i>)	При реализации проекта команда избегает излишней нерезультативной активности и чрезмерного использования средств автоматизации проектной деятельности	6 %
Профессиональный проектный менеджмент (англ. <i>project management expertise</i>)	Уровень владения методологией проектного управления и его инструментами обеспечивает выполнение и перевыполнение требований заинтересованных в проекте сторон и организации в целом	5 %
Прозрачность бизнес-целей (англ. <i>clear business objectives</i>)	Цели проекта соответствуют бизнес-целям организации, ее стратегии и понятны всем участникам проекта	4 %

Среди ключевых факторов успеха (15 % вклада в успех) приводится такой, как зрелость поведения (англ. *emotional maturity*). Под набором данных факторов понимают более 50 навыков участников проекта, например: управление ожиданиями, достижение согласия по спорным вопросам, сотрудничество и др.

Дополнительный выявленный проблемный фактор — недостаточный уровень умений и знаний членов рабочих групп проектов. Компании в основном ориентированы не на долговременное обучение персонала, занятого в проектах, а на быстрые внедрения инструментов проектного управления, которые не приносят успеха при дефиците обучения.

Большое значение имеет то, что большинство компаний игнорируют необходимость взвешивания требований проекта относительно возврата инвестиций (англ. *return on investment, ROI*). Это достаточно сложная методология, но на практике может применяться упрощенный вариант взвешивания значения отдельных требований относительно *ROI* по пятибалльной шкале.

Среди результатов анализа приведены, на первый взгляд, противоречивые данные относительно точности целеуказания при проектном управлении. Проекты, где были точно заданы цели, оказались менее успешны, чем те, где цели были более размыты. В некотором смысле результаты подтвердили выводы научной работы Такеши и Нонака (*Takeushi & Nonaka*) о необходимости общего целеуказания на высоком уровне. Авторы исследования выдвигают радикальную идею: только то, что неизвестно заранее, способно произвести реальный результат. Обычно в ИТ-проектах невозможно представить результат и ход реализации проекта в деталях. Задача дополнительно усложняется, когда ИТ-руководство пытается синхронизировать ИТ-проекты с целями организации в целом. Часто это приводит к тому, что либо организация, либо ИТ начинают тормозить общее движение. Данные выводы противоречат традиционному подходу при реализации проектов, который требует обеспечить точную формулировку целей по модели *SMART* (от англ. *specific, measurable, achievable, relevant, time-bound*).

Интересно, что авторы исследования отдельно указывают на то, что с их точки зрения проектная организация работ не является наиболее эффективной, и они настойчиво рекомендуют методики, базирующиеся на непрерывной поставке программных продуктов.

В последнее время критерии успешности проектов подвергаются пересмотру. Институт проектного управления (англ. *The Project Management Institute, PMI*) определяет успех проекта по трем основным координатам: вовремя (англ. *on time*), в рамках бюджета (англ. *on budget*), точно в цель (англ. *on target*). Такой набор координат также называют «железным треугольником», или тройным ограничением.

Вероятно, под влиянием вывода о низком значении точного целеуказания *The Standish Group* определяет успех проекта как вовремя, в рамках бюджета и с удовлетворительным результатом.

Успех как цель крайне трудно определяется. Большое количество проектов, на первый взгляд достигнувших успеха в рамках «железного треугольника», не принесли ожидаемой результативности для компаний-заказчиков.

5.2. Основные проблемы проектного управления

The Standish Group в рамках исследования сделала вывод о том, что уровень успешности ИТ-проектов в международной практике на протяжении последних лет не превосходит 30 %. При этом существуют оценки, которые делают данную картину еще более удручающей. В соответствии с ними не более 11 % от «успешных» проектов оказываются действительно полезными, т. е. соответствующими бизнес-целям организации¹.

Основной проблемой проектного управления является то, что любой проект направлен на создание уникального нового продукта услуги, системы и т. п. То есть, он должен привести к тому, что организация, реализующая проект, должна измениться в большей или меньшей степени.

Любая функционирующая сложная система стремится к сохранению стабильного состояния, и ее нормальное поведение заключается в сопротивлении изменению. Таким образом, инициатор проекта и проектная группа неизбежно сталкиваются с сопротивлением взаимодействующих подразделений, сотрудников, внешних компаний (партнеров) тем более заметным, чем более значимые изменения предполагаются проектом.

При реализации проекта приходится учитывать, что проектная группа располагает ограниченными ресурсами, а у организации, реализующей проект, присутствует не только ограничение сопротивляемости изменениям, но и предел ее возможности адаптироваться к этим изменениям. Превышение предела возможности адаптироваться к изменениям может привести не только к прекращению проекта, но и разрушению реализующей проект системы, что может выражаться в различной форме вплоть до прекращения деятельности компании.

Широко известна полшутливая примета 1990-х гг., которая связывала прекращение работы очередного коммерческого банка с этапом завершения подготовки котлована под фундамент нового здания. То есть банки, переоценивая свои ресурсы и недооценивая сложность проекта постройки здания, массово прекращали свое существование.

Отсюда, например, в проектном управлении находит применение подход так называемых «дельфинов», в рамках которого любой крупный проект («кит») предлагается разбивать на серию мелких («дельфины»). Реализуя данный подход, руководитель проекта проводит организацию через серию мелких изменений с получением полезного

¹ Громов Г. Р. Очерки информационной технологии.

результата на каждом этапе, что позволяет снизить ее сопротивление и экономит ресурсы проектной группы.

Дополнительный выигрыш в эффективности получают в рамках реализации стандартных подходов к управлению проектами, активизируя взаимодействие со всеми участниками проекта в течение всего периода работы за счет разработки и выполнения детального плана коммуникаций. Это позволяет не только снизить уровень имеющих место конфликтов, но и активизировать адаптационные процессы участников.

Упоминание темы конфликтов является обязательным при рассмотрении проблем проектного управления. Раздел по управлению конфликтами включен, в том числе, в ведущий стандарт *PMI PMBOK*¹.

Источником конфликтов является различие в целях участников проекта, взаимодействующих с ними подразделений, сторонних компаний. Руководитель проекта стремится реализовать проект, подрядчик — получить оплату, затрагиваемые проектом подразделения в основном хотят сэкономить ресурсы и сохранить существующее положение вещей и т. д. При этом стороны конфликта практически никогда не имеют достоверной оценки причин его возникновения.

Результатом конфликтов становится взаимная адаптация и успех в реализации проекта либо отказ от адаптации и провал проекта. Уровень конфликта может быть измерен методами теории экспертного оценивания. Научные работы последнего времени в области проектного управления и анализа конфликтов (например, Барки²) показали связь высоких уровней конфликта при реализации проектов с падением качества получаемого результата проекта. В организациях, не располагающих развитой системой проектного управления (низкого уровня зрелости в терминах модели *СММ*), уровень конфликтов резко изменяется в ходе реализации проекта и обычно имеет седловидную форму (рис. 5.1).

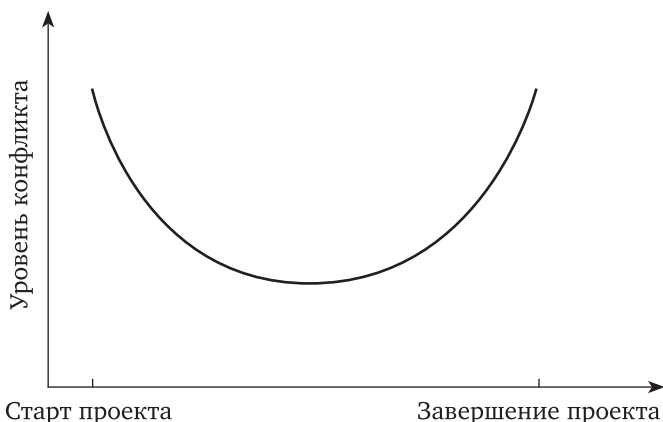


Рис. 5.1. Типовая кривая конфликта

¹ Project Management Institute Project Management Body of Knowledge.

² *Barki H.* Interpersonal Conflict and Its Management in Information System Development // *MIS Quarterly*. — 2001. — P. 195—228.

То есть весь конфликт, который должен в идеале создавать адаптационное давление на всем протяжении проекта, локализуется с чрезмерными уровнями в этапах инициации проекта (согласования плана, бюджета, состава проектной группы) и сдачи работ (приемка и оплата).

Следующей значимой проблемой проектного управления является то, что сложность проектов как в области информационных технологий, так и в других областях деятельности неуклонно возрастает. Проектным группам приходится обеспечивать проектирование и создание все более сложных систем с привлечением к работам по проекту все более сложно структурированных коллективов, что при отсутствии новых приемов управления и проектирования приводит к аварийному прекращению проектов или к недопустимому снижению качества их результата.

При этом для случая автоматизированных систем ситуация резко утяжеляется. Включая в себя все большее количество компонент, которые должны отслеживать изменения внешнего мира, они уже не располагают временем на «отдых» и должны подвергаться практически непрерывным изменениям. То есть все большее значение приобретает динамическая сложность, связанная с темпом изменений требований к функциональности автоматизированных комплексов.

В цифрах это выражается следующим образом: в конце 1970-х гг. срок создания новой промышленной программной системы составлял около одного года при сроке функционирования без потребности в значимых изменениях около семи лет. В конце 1980-х гг. средний срок разработки сравнялся со сроком функционирования без изменений — на уровне 2—3 лет¹. В настоящее время срок создания новой крупной системы в разы превышает срок ее сохранения в первоначальном виде.

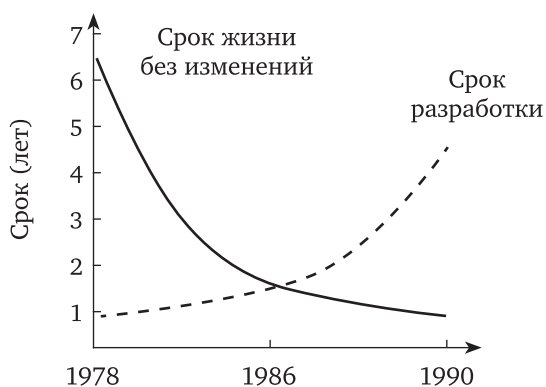


Рис. 5.2. Время первичной разработки и срок жизни без изменений промышленных прикладных систем

Таким образом, при попытке создания новой системы с подготовкой полных детальных функциональных требований проектная группа

¹ Громов Г. Р. Очерки информационной технологии.

рискует не добраться до этапа разработки, не говоря уже о завершении проекта. Требования к программе неизбежно начинают меняться под влиянием внешних условий до завершения их согласования, и это не является результатом ошибок, допущенных при инициации проекта.

Однако несмотря на вышеуказанную проблему, крупные программные системы продолжают создаваться и эволюционировать. Ключевое слово — «эволюционировать». В успешных проектах применяется подход, при котором на начальном этапе формируется минимальный набор требований, реализуемых с полезным эффектом максимально быстро. Затем система переводится в состояние эксплуатации с обеспечением ускоренного развития по изменяющимся требованиям.

Результат такого подхода иллюстрируется, например, показателями перераспределения относительных затрат между этапами жизненного цикла сложного программного обеспечения, например, за период с 1975-го по 1985-й г. (табл. 5.4).

Таблица 5.4

Относительные трудозатраты по этапам жизненного цикла проекта¹

Этап жизненного цикла автоматизированной системы	1975 г.	1985 г.
Постановка задачи и проектирование	28 %	40 %
Кодирование	14 %	3 %
Отладка и приемо-сдаточные испытания	28 %	7 %
Сопровождение	30 %	50 %

Из таблицы можно сделать несколько выводов. Первый — за счет внедрения средств быстрой разработки и языков высокого уровня со временем снижаются затраты на непосредственное написание текстов программ и их отладку, т. е. на работу программиста. Второй — усложнение программных систем приводит к росту затрат на проектирование и перемещению значительного объема работы с этапа создания программы на этап ее развития и сопровождения.

Осознание данного факта привело, в том числе, к созданию радикально новых подходов к разработке программного обеспечения, таких как *agile*. В рамках данного метода разработки практически не предусматривается первичное проектирование программ, и делается попытка перевести разрабатываемый продукт в этап сопровождения и развития с первых шагов работы.

Среди проблем проектного управления в области информационных технологий нельзя не отметить отдельно нерешенную проблему налаживания взаимопонимания между заказчиком (пользователем) автоматизированной системы и ее разработчиком. Пользователь не понимает языка технического специалиста. Разработчик программного обеспечения не владеет знаниями, необходимыми для однозначного понимания требований бизнес-специалиста.

¹ Громов Г. Р. Очерки информационной технологии.

Несмотря на развитие различных языков формального описания бизнес-процессов, таких как универсальный язык моделирования (*UML*) и соответствующих систем проектирования (например, *Rational Rose*), данная проблема не может считаться разрешенной. Ее существование приводит к неизбежности ошибок при составлении требований на разработку программного обеспечения и частично компенсируется в основном за счет специализации коллективов разработчиков и их обучения в рамках отдельной предметной области или за счет привлечения бизнес-специалистов к программированию, что не может быть отнесено к промышленному решению проблемы.

Еще одну проблему проектного управления представляет затруднительность промежуточного контроля реализации проекта в части объема выполненных работ и их качества.

Периодическая отчетность руководителя проекта перед заказчиками проекта и его спонсором не решает проблему не только в связи с тем, что руководитель проекта заинтересован в оптимистичности оценки промежуточного результата, но и с тем, что промежуточный этап не поддается оценке из-за отсутствия объективно распознаваемого и измеряемого результата. Именно поэтому большое значение придается выделению контрольных точек в календарных планах проектов, а ошибки в их планировании приводят к утере внешнего контроля и зачастую — к провалу проекта.

Смежной проблемой является отсутствие надежных методик планирования сложных проектов. План проекта является необходимым элементом договора руководителя проекта с другими заинтересованными сторонами. При этом план составляется на этапе инициации проекта до выполнения основного объема работ по системному проектированию. Затруднительность планирования и очевидная необходимость внесения изменений в план в ходе выполнения работ часто провоцирует организации на полный отказ от общего планирования, что не является корректным, приводит к рассинхронизации деятельности взаимодействующих подразделений, утрате внешнего контроля над проектом, снижению качества результата, необоснованному перерасходу бюджета и превышению сроков.

Проект определяется как деятельность, направленная на достижение уникального результата. Уникальность каждого проекта порождает повышенную персональную зависимость от руководителя проекта, которую затруднительно скомпенсировать даже внедрением развитых технологий проектного управления. Это усложняет подбор соответствующих сотрудников, выдвигает дополнительные требования к их знаниям, опыту и психологическим характеристикам. Поэтому в настоящее время проектное управление в значительной степени является ремеслом и искусством, а не технологией, обеспечивающей гарантированный качественный результат при выполнении неких технологических требований.

5.3. Отличие проектной деятельности от текущей

Реализация проектов направлена на создание нового уникального результата, продукта или услуги. В этом ее основное отличие от текущей операционной деятельности (рис. 5.3).



Рис. 5.3. Отличие проектной и текущей деятельности

К середине XX в. в связи с высокой интенсивностью инноваций и крупных проектов, реализуемых в основном в оборонной и космической сфере, произошла стандартизация проектной деятельности и ее выделение в отдельную специализацию. «Гонка вооружений» и «космическая гонка» привели к рождению проектного управления в качестве отдельной области деятельности.

Хорошим примером проекта является первый или очередной ребенок в молодой семье. «Новый результат» для молодых родителей, «проект» ограничен по времени, поддается планированию и бюджетированию. При этом рост количества подобных «проектов» неизбежно приводит к необходимости создания специализированных родильных домов, для которых они превращаются в процессную текущую деятельность с выделением соответствующей медицинской специальности.

Так непрекращающийся поток инновационных проектов XX в. привел к «рождению» новой специальности «руководитель проекта», которая располагает своим специфическим набором знаний и инструментов.

Для его реализации создаются объединенные команды из специалистов различных подразделений и внешних компаний. Специалисты — участники проекта оказываются в двойном подчинении. Они

подчиняются руководителю проекта на временной основе и своему постоянному линейному руководителю. Таким образом, проект является «организационной аномалией»¹. Проект игнорирует границы между подразделениями и функциональными обязанностями сотрудников в рамках текущей деятельности.

5.4. Проект как адаптационный процесс

Организация является сложной развивающейся системой и подчиняется в своем развитии теории сложных систем. Реализуемые проекты — двигатели этого развития, источник нового состояния организации, продукта или услуги.

Закон взаимной адаптации для сложных систем звучит следующим образом: «Динамика развития любой системы есть процесс взаимной адаптации компонентов системы между собой и системы с внешней средой». При этом «любая сложная система может быть реализована посредством одной из дискретного ряда ее возможных структур»², и переход между ними не всегда возможен в ограниченный период времени даже с высокими затратами ресурсов. На рис. 5.4 представлена диаграмма перехода состояний сложной системы — организации из состояния «до проекта» в состояние «в результате проекта».

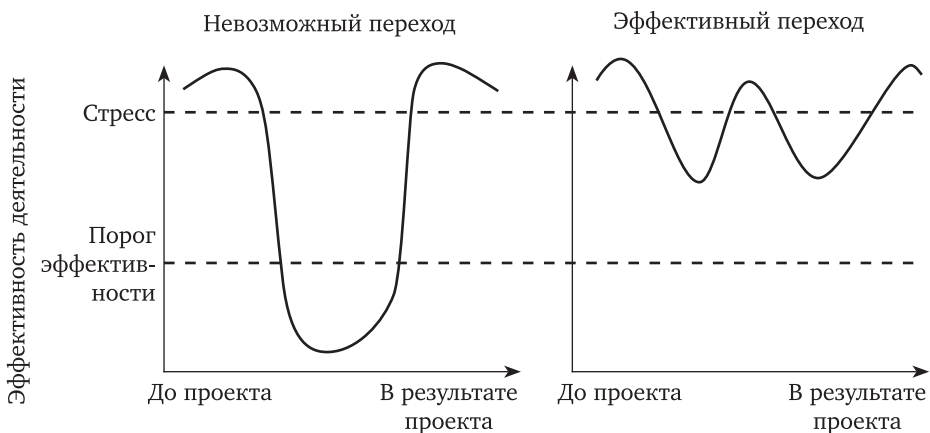


Рис. 5.4. Диаграмма переходов состояния системы при реализации проекта

Важно понимать, что иногда состояние «в результате проекта» может не совпадать с набором возможных для системы состояний-структур, а иногда это состояние не является непосредственно достижимым за один шаг.

¹ См.: Верзух Э. Управление проектами: ускоренный курс по программе МВА. М. : Вильямс, 2017.

² Венда В. Ф. Системы гибридного интеллекта. М. : Машиностроение, 1990.

Это утверждение хорошо иллюстрируется присутствующими в документации *СММИ* рекомендациями не пытаться наращивать зрелость путем реализации проектов, с «перепрыгиваниями» промежуточных уровней зрелости, например с первого уровня зрелости на третий или четвертый. Допустимыми переходами являются с 1 на 2, со 2 на 3 и т. д.

На рис. 5.4 кроме кривых перехода отмечены уровни «стресса» и «порога эффективности». Для любой живой системы существует пороговое значение эффективности деятельности. Если ее эффективность оказывается ниже порогового значения, то система разрушается.

Уровень «стресса» отмечает уровень падения эффективности, при котором система начинает «чувствовать» свою неадаптированность к внешней среде, т. е. испытывать стресс. Переходы между состояниями всегда связаны с некоторым падением эффективности и стрессом. Главное, чтобы это падение эффективности и стресс не стали слишком большими.

Сложной системе «не нравится» испытывать стресс, поэтому любая система первично сопротивляется тому, что ее выталкивает в это состояние. Соответственно и организации сопротивляются проектам. Цель системы — сохранять комфортное для нее состояние покоя и быть вне стрессового диапазона. Цель проекта — перевести организацию в новое состояние. Разница целей становится источником конфликта. Конфликт становится для наблюдателя ценным источником информации.

PMI PMBOK и *СММИ* уделяют достаточно большое внимание вопросам управления конфликтами при реализации проектов. Но в них не поясняются реальные источники возникновения конфликтов. Конфликт ошибочно представляется как результат «плохого поведения» участников команды.

В действительности проектные конфликты возникают неизбежно и являются следствием того, что одна часть организации активно продвигая проект, заставляет адаптироваться к его будущим результатам остальную часть организации. Чем активнее происходит реализация проекта, тем более активное сопротивление ему оказывается. Адаптация ресурсозатратна. Любая сложная система сопротивляется потере ресурсов. Это является основной причиной возникновения конфликтов между подсистемой — командой проекта и всеми остальными подсистемами, входящими в организационную систему.

Конфликт в рамках проекта — процесс взаимодействия подсистем, сопровождающийся их взаимной адаптацией, снижением эффективности и стрессом.

Критерии эффективности и цели взаимодействующих подсистем в организации всегда частично не совпадают между собой и не совпадают с соответствующими параметрами надсистемы. То есть цели подсистем и надсистемы находятся в конфликте, а рост эффективности надсистемы (организации) сопровождается снижением эффективности подсистем.

При возрастании уровня зрелости (по модели *СММТ*) происходит переход от управления проектами к управлению процессами. Это означает, помимо прочего, что организация переводится в состояние постоянного обновления, модернизации и адаптации к изменениям внешней среды, что резко снижает ее сопротивление изменениям.

В отличие от высших, низкие уровни зрелости (1—2) характеризуются повышенным сопротивлением реализуемым проектам.

В рамках данного подхода конфликт рассматривается как естественное явление, вызванное любым взаимодействием сложных систем, которое в случае присутствия человеческих компонент может сопровождаться негативными эмоциями различного уровня (рис. 5.5). При этом уровень зрелости организации характеризуется не только уровнем негативных эмоций, сопровождающих конфликты, но и преобладанием одного из пяти стандартных подходов к разрешению конфликтов: отстаивание своих интересов (англ. *asserting*), приспособление (англ. *accommodating*), компромисс (англ. *compromising*), решение проблемы (англ. *problem-solving*) и избегание (англ. *avoiding*).

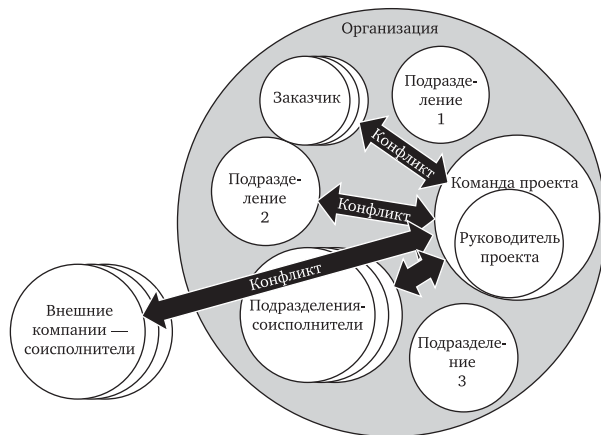


Рис. 5.5. Конфликты

Понятно, что преимущественное отстаивание своих интересов, приспособление и избегание характеризует низкий уровень зрелости организации, так как приводит к снижению качества достигаемых проектами результатов. С точки зрения модели зрелости *СММТ*, организации, находящиеся на более высоком уровне зрелости, не только характеризуются более высоким уровнем взаимной адаптации входящих в них подсистем (подразделений, сотрудников и процессов), но и располагают дополнительным резервом адаптальности, что обеспечивает им возможность реализации более сложных проектов. При этом *резерв адаптальности* — это расстояние между пиком достигнутой эффективности деятельности и пороговым уровнем.

Характеристики конфликта в ходе реализации проекта в значительной степени определяются корпоративной культурой и клима-

том в организации, т. е. показателями зрелости организации, а также характеристиками процессов в команде проекта (рис. 5.6). Чем ниже уровень зрелости, тем выше уровень конфликта для проектов равной сложности.

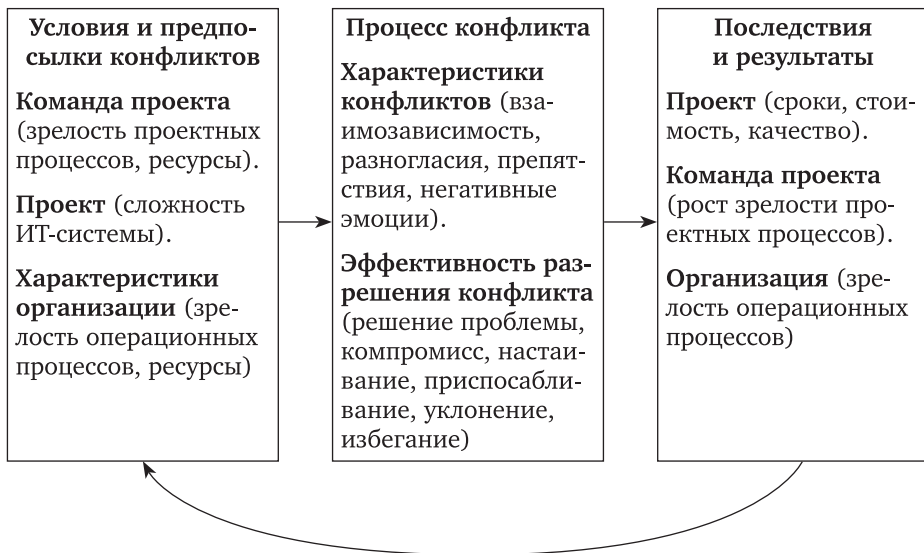


Рис. 5.6. Процесс реализации проекта

Кроме того, на уровень конфликта влияет сложность проекта. Чем ниже уровень зрелости организации, процессов проекта и выше сложность проекта, тем выше уровень конфликта и ниже эффективность его разрешения. Чем выше уровень конфликта, тем хуже качество результата проекта¹.

Следует обратить внимание на то, что процесс конфликта зависит от ресурсов организации (и ее подсистем, участвующих в конфликте) и ресурсов проектной команды. Если ресурсов команды не хватает, то она не сможет «вытолкнуть» организацию из состояния «комфортного покоя» в состояние «проект реализован». Если у проектной команды ресурсов слишком много, то она может довести организацию до состояния критической потери эффективности.

Оценивая уровень конфликта, можно получить количественные показатели зрелости организации и эффективности проектных процессов, а оперативно влияя на процесс конфликта — управлять качеством результата проекта и обеспечивать возрастание уровня зрелости организации.

Положительная роль конфликта является следствием реализации процессов взаимной адаптации взаимодействующих систем, которые активизируются при реализации проектов. При этом неправильно

¹ Barki H. Interpersonal Conflict and Its Management in Information System Development // MIS Quarterly. — 2001. — P. 195—228.

управляемые конфликты приводят к растрате ресурсов. Если мы правильно используем информацию, поставляемую конфликтами, то можем точно и сфокусированно влиять на неэффективно взаимодействующие процессы, развивая их (рис. 5.7). Из рисунка видно, что основными претендентами на повышение эффективности являются процессы контроля результата и сдачи-приемки. Учитывая то, что уровень может оцениваться на различных этапах, мы можем еще более точно находить «больные места» в организации и проектной работе.

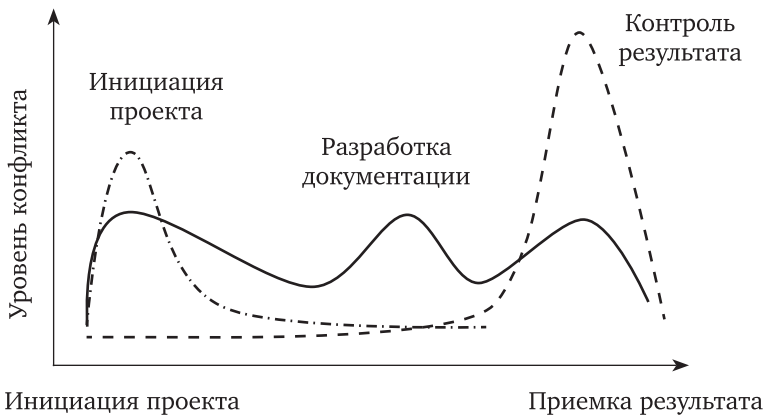


Рис. 5.7. Пример оценки уровня конфликта по ходу реализации проекта для различных процессов

Таким образом, проектные конфликты могут служить эффективным двигателем развития. Например, вспышка конфликта при оформлении проектной документации может служить источником информации о том, что отсутствует достаточная регламентация процесса разработки и согласования документов. Если мы эту информацию не используем, то в следующем проекте или на следующем этапе проекта этот процесс в очередной раз сработает неэффективно и конфликт, «съедающий» ресурсы, возникнет повторно, поскольку развивающий адаптационный процесс не был выполнен корректно.

Конфликты, реализуемые с участием людей, поддаются количественной оценке по набору параметров через оценку их восприятия конфликтующими сторонами (уровень разногласий, вмешательства и негативных эмоций, взвешенных относительно количественной оценки основного условия возникновения конфликта — взаимозависимости вовлеченных в конфликт сторон).

Человек — это универсальный датчик. В данном случае он поставляет информацию об уровне неэффективности взаимодействующих процессов. Информация, получаемая от конфликтующих сторон, при «усреднении» обеспечивает общую целостность и суммарную объективность.

Оперативно получаемая оценка уровня конфликта в качестве меры дезадаптации (необходимости адаптироваться) позволяет получить

представление о готовности участвующей в проекте организации к реализации текущего этапа работ.

Оценка проявляющегося подхода к разрешению конфликта в рамках этапа работ связана с потенциальным качеством результата и позволяет прогнозировать данное качество с опережением. «Содействие» (компромисс, решение проблем) демонстрирует готовность к адаптации и обеспечивает рост качества, «противодействие» (уклонение, настаивание, приспособливание) затрудняет взаимную адаптацию и снижает качество.

Эффективное управление конфликтом может быть построено только на основе повышения эффективности обмена информацией между сторонами. Оно невозможно без повышения информированности сторон о целях и функциях друг друга, а также без повышения уровня собственной информированности о параметрах своей деятельности¹.

Подробный анализ конфликтов приводит к общему повышению эффективности деятельности взаимодействующих сторон за счет повышения общей информированности.

Являясь с точки зрения теории взаимной адаптации необходимым условием поступательного развития организации, конфликты при чрезмерном уровне приводят к дополнительным непроизводительным потерям, остановке развития и даже создают угрозу существованию отдельных подсистем организации.

В рамках проектной деятельности типовые процессы реализуются поэтапно. Следовательно, оценка конфликта на каждом этапе позволяет получить оценку эффективности взаимодействующих стандартизированных процессов и выполняющих их систем. Другими словами, обнаружив вспышку конфликта при реализации проекта и проанализировав ее, мы можем выявить неэффективно выполняемые процессы и неэффективно работающие подразделения.

В связи с наличием предела адаптативности систем можно сделать вывод о наличии предела реализуемых в рамках проектов изменений и, соответственно, уровня допустимой сложности реализуемых проектов.

Оценка и управление конфликтами позволяет контролировать качество проекта, а также обеспечивать поэтапное развитие операционных и проектных процессов организации на основе цикла постоянного улучшения качества.

В заключение раздела предлагаем дополнительно прочитать книгу Нассема Талеба «Антихрупкость»². В ней в популярной форме изложены идеи, касающиеся адаптации, сложных систем и конфликтов. Способность адаптироваться (адаптивность) при этом названа антихрупкостью.

Кроме того, может быть рекомендована к изучению книга Гарольда Керцнера «Стратегическое управление в компании. Модель зрелого

¹ Дружинин В. Введение в теорию конфликта. М. : Радио и связь, 1989.

² Талеб Н. Антихрупкость. М. : Колибри, 2015.

управления проектами»¹. В ней подробно изложена модель зрелости управления проектами *Project Management Maturity Model (PM3)*. Особую ценность представляет вопросник, позволяющий самостоятельно оценить текущий уровень зрелости проектного управления в организации.

5.5. Стратегия, портфель, программа, проект

Необходимо различать три базовых понятия проектной деятельности: проект, программа, портфель проектов (рис. 5.8).

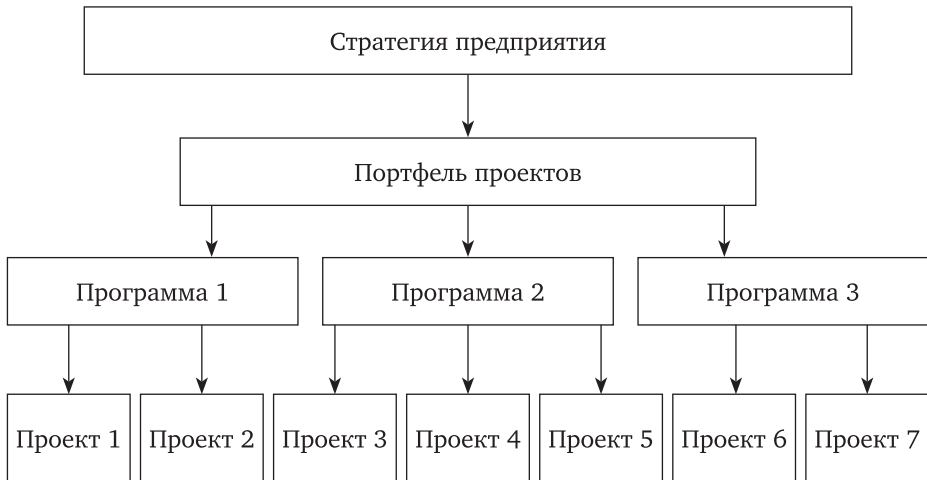


Рис. 5.8. Соотношение понятий «проект», «программа», «портфель проектов»

Вся совокупность проектов организации образует **портфель проектов**. Он подлежит единому управлению с точки зрения контроля, координации по ресурсам, срокам и задачам отдельных проектов. Проекты сгруппированы в **программы** исходя из бизнес-целей, закрепленных в **стратегии** развития предприятия. Таким образом обеспечивается декомпозиция целей от укрупненных бизнес-целей к частным проектам с измеряемым точным результатом и иерархическое управление структурой проектов.

5.6. Офис управления проектами

Для управления портфелем проектов обычно создают **офис управления проектами**. Его функции и полномочия могут варьироваться от оказания поддержки в управлении проектами до непосредственного

¹ Керцнер Г. Стратегическое управление в компании. Модель зрелого управления проектами. М. : ДМК-Пресс, 2010.

управления проектами. В любом случае офис управления проектами обеспечивает методологическую поддержку, развитие инструментов управления проектами и централизованный мониторинг исполнения.

На практике решение о форме работы офиса управления проектами и его полномочиях (поддерживающий, контролирующий или управляющий) принимается, исходя из потребностей и особенностей конкретной организации.

Например, если уровень зрелости проектного управления в организации является низким (руководители проектов недостаточно квалифицированы, инструменты не развиты, проектные процессы не институционализированы, линейные подразделения не располагают достаточным штатом для организации проектной работы), то, безусловно, обоснованным является создание офиса проектной работы в формате управляющего — от него ожидается высокая вовлеченность управляющего ресурса в каждый реализуемый проект. За счет концентрации компетенции проектного управления и ее опережающего развития может быть достигнута повышенная результативность реализации проектов.

Если в такой ситуации офис управления проектами ограничит свою деятельность контрольными функциями, то при отсутствии единой методологии и, например, требований к документированию, отдельные проектные команды не будут справляться даже с отдельными типовыми проектными задачами, например, с разработкой устава проекта. При значительном количестве заинтересованных сторон проектная команда не сможет самостоятельно согласовать документ. Подобные примеры на практике встречаются достаточно часто.

Глава 6

РУКОВОДИТЕЛЬ ПРОЕКТА

6.1. Психологические характеристики руководителя проекта

Роль руководителя проекта является центральной в проектном управлении. К нему предъявляются очень высокие профессиональные и личностные требования. Поскольку реализация проекта связана с созданием уникального, нового для организации результата, то руководителю проекта приходится преодолевать естественное сопротивление взаимодействующих подразделений и сотрудников, преимущественно стремящихся сохранить привычную для текущей деятельности систему отношений, функций и процессов. Реализация проекта вынуждает сложную систему изменяться. Любая сложная система сопротивляется изменениям.

Формальное выполнение функций, являющееся достаточным для текущей деятельности, практически наверняка приведет к неуспешному завершению проекта.

В различных методиках оценки личностных психологических характеристик встречается классификация, связываемая либо со способностью удержания цели¹, либо с параноидальным мышлением (в хорошем смысле).

С этой точки зрения приведем показательную цитату: «Параноик склонен к “застреванию”. Но... параноику хватает сил застревать не на поверхностных, формальных, а на содержательных сторонах жизни и деятельности. И, застревая, переделывать, преобразовывать их. Ощущая в себе большой заряд энергии, параноик, как правило, ставит перед собой задачи, сложность и масштабы которых объективно превышают возможности индивидуума. При этом он переживает необходимость решения этих задач как свою собственную, лично значимую проблему... С одной стороны, он не хочет отказываться от задуманного, но с другой — реально не в состоянии воплотить в жизнь свои намерения в одиночку. Вот из чего рождается настоящее лидерство. Параноик — истинный лидер. Он твердо знает, чего хочет. Он преисполнен целеустремленности и энергии. Он воспринимает окружающих как подспорье в реализации своего масштабного замысла, по сути — как

¹ Ben-Hur S. Talent Intelligence: What You Need to Know to Identify and Measure Talent. San Francisco : Jossey-Bass, 2013.

часть самого себя...»¹. Здесь приведено описание одного из «радикалов» — ярко выраженных черт личности, которые присутствуют в нас наряду с другими, в большем или меньшем объеме. Понятно, что описанная характеристика является ключевой для руководителя проекта. Для лучшего понимания вопроса может быть рекомендовано в полном объеме изучить приведенный источник и попытаться промоделировать ситуацию, при которой руководителем проекта является пусть даже хорошо обученная личность с превалированием радикала «истероид» или «гипертимный».

Если говорить о способности «удержания цели», то люди с низким уровнем данной характеристики не способны помнить смысл задачи даже на протяжении дня. Естественно, что из такого человека, несмотря на все его таланты и компенсационные механизмы, хорошего руководителя проекта не получится.

Здесь мы обсуждаем сложные информационно-технологические проекты, срок которых нередко превышает 1—2 года, а количество вовлекаемых специалистов редко бывает менее десятка. При этом необходимо обеспечить вовлечение в проект широкого круга затрагиваемых взаимодействующих компаний или подразделений. Не обладая необходимыми для организации такой работы качествами, руководитель проекта не обеспечит его успех. По ходу реализации он будет постоянно терять цель проекта или сдаваться перед очередными трудностями. Именно поэтому такое большое внимание уделено личностным качествам руководителя проекта в отдельной главе *PMI PMBOK*. Теме лидерства, личностных характеристик руководителя проекта посвящено шесть страниц этого руководства². При этом четко отделяется понятие «управления» и непосредственно «лидерства».

К ключевым компонентам лидерства авторы руководства, кроме умения работать с людьми, справедливо относят «видение перспективы», т. е. способность создать образ будущего и передать его другим, а также способность к постоянной фокусировке на приоритетах проекта.

Авторы *PMI PMBOK* считают, что к необходимым «талантам» руководителя проекта относятся три основные категории (рис. 6.1): техническое руководство проектом; лидерство; стратегическое управление.

Под техническим руководством проектом понимается набор знаний и навыков по выполнению проектов в соответствии с соответствующей методологией и технологией.

Под стратегическим управлением — способность привязывать цели, задачи и приоритеты проекта к стратегическим целям организации в целом.

Здесь особое внимание следует обратить на то, что авторы называют эти категории «талантами». Это подчеркивает их экстраординарность.

¹ Пономаренко В. В. Практическая характерология с элементами прогнозирования и управления поведением (методика «семь радикалов»). Ростов н/Д. : Феникс, 2006.

² Руководство *PMBOK*. Project Management Institute, 2017.

Хорошего руководителя проекта нельзя создать за счет обучения. Эта профессия требует особых врожденных характеристик и способностей.



Рис. 6.1. Треугольник талантов руководителя проектов по PMI PMBOK

Кроме личностных психологических характеристик, руководитель проекта должен обладать достаточной технической квалификацией. Без квалификации в информационных технологиях он станет заложником мнений экспертов, которые, учитывая наличие естественного конфликта, связанного с сопротивлением среды и конфликтов интересов сторон в любом крупном проекте, всегда будут противоречивыми. В этой ситуации технически некомпетентный руководитель проекта будет вынужден отдать роль технического арбитра другому участнику проекта, и расстанется, таким образом, с одним из своих ключевых властных полномочий.

Берясь за сложный масштабный проект с высоким бюджетом, руководитель предварительно должен подтвердить успешную реализацию ранее выполненного проекта, масштаб которого не должен быть намного меньше иницируемого.

Формально обученным новичкам не место в руководстве сложными проектами. С этой работой часто не справляются и опытные профессионалы, которые, накапливая реальный опыт, никогда не перестают учиться новому — это первичное требование успешного проектного управления. Успешно реализованный в одной организации проект не гарантирует успеха однотипного проекта в другой. Уровень зрелости организации может оказаться ниже. Это потребует более детального планирования и разбиения результата на более мелкие фрагменты (MVP) — «разбиение кита на дельфинов».

Если руководитель проекта не располагает необходимым опытом, то перегрузит исполнителей несвойственными им задачами, переложив всю работу по планированию на них, а если не научится, то будет допускать ошибки, необоснованно полностью полагаясь на предыдущий опыт.

Карьера руководителя проекта с этой точки зрения сходна с карьерой инженера, которая заключается в постепенном наращивании сложности решаемых задач проектирования и управления. При этом параллельно с ростом сложности технических задач инженер вынужден наращивать способности управления коллективом — сложная система требует сложной организации для своего создания.

Следует дополнительно обратить внимание на то, что руководитель проекта должен обладать вышеприведенными «талантами» лично. Это один человек. Это лидер, обладающий необходимыми способностями, знаниями и опытом. Попытки создать «синтетического» руководителя проекта за счет реализации его функций коллективно (например, технически некомпетентный стратегический лидер плюс технический эксперт) обычно завершаются неуспешно. Такое взаимодействие может быть эффективным в рамках отработанных процессов и высокого уровня зрелости. Однако, во-первых, проектное управление практически всегда выходит за рамки текущей деятельности, а во-вторых, организаций с высоким уровнем зрелости процессов (3—5 уровень *СММ*) очень мало. Так что роль руководителя проекта — роль одиночная и персонально зависимая, в отличие от ролей процессов текущей деятельности.

6.2. Функции руководителя проекта

1. Основная функция руководителя проекта — достижение целей проекта. Уровень требований к технической квалификации и другим «талантам» руководителя проекта определяется его *базовыми функциями*.

Что входит в этот список? Первое и основное — он должен добиться достижения целей проекта. Причем не формально закрепленных, а постоянно уточняемых, часто скрывааемых или подменяемых заинтересованными сторонами, релевантных стратегическим целям организации.

Всем проектам мешают внешние и внутренние факторы. Профессиональный руководитель проекта их знает, адекватно оценивает и эффективно противостоит негативному воздействию. Если он взялся за проект и подписался в уставе проекта, он заранее знает, как будет добиваться успеха. Он достигнет результата. Оправданий для неуспешности он внутренне не приемлет.

У профессионализма есть и другая сторона — если руководитель проекта вынужден был взяться за потенциально рискованный проект под давлением, то, скорее всего, данный проект будет закрыт без достижения успеха достаточно быстро (в этом есть также значительная положительная сторона — сокращение потерь). При этом ответственность за потери будет переложена на исполнителей или заказчика. Профессиональный руководитель проекта четко понимает зоны ответственности и потенциальные риски. Он ими уверенно управляет. Главное,

чтобы его профессионализм не ограничивался успешным для него персонально закрытием неуспешных проектов.

2. Декомпозиция целей и задач проекта. Следующей по значимости функцией руководителя проекта является декомпозиция задач и целей проекта. Под этим понимается в случае информационно-технологического проекта разбиение создаваемой или модернизируемой системы на блоки и модули, а также разбиение общих задач на блоки соответствующих работ с привлечением необходимых процессов. Понятно, что данная функция не может выполняться без необходимой инженерной квалификации — этого требует, как минимум, первая часть функции. Результатом этой части приведенной функции является ориентировочная архитектура системы.

Разбиение проекта на подзадачи (рис. 6.2), по сути, является высокоуровневым планированием, в результате которого проект в первую очередь получает оценку ресурсов и *контрольные точки*, привязанные к временным отметкам (англ. *milestones*). Безусловно, рисунок схематичен и не представляет разнообразия декомпозиции систем и высокоуровневых контрольных точек, но дает представление об общей функции.

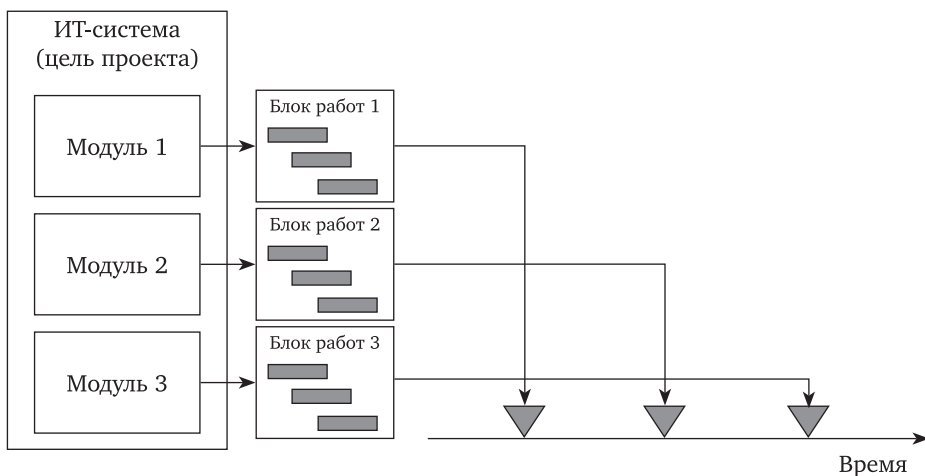


Рис. 6.2. Декомпозиция проекта

Основные требования к выделению контрольных точек выдвигаются сходным образом, с определением целей проекта. Отметим, что эти точки должны быть привязаны к материальному результату, распознаваемому независимым наблюдателем и подтверждающему соответствующий прогресс проекта в направлении достижения его целей. Безусловно, отчет руководителя проекта или не согласованный с заказчиком акт приемки очередного модуля системы не соответствуют требованиям к контрольным точкам. При этом запись о выполнении блока задач в соответствующей промышленно эксплуатируемой автоматизированной системе контроля исполнения, не смотря на свою «нематериальность», может вполне соответствовать этим требованиям.

Здесь мы говорим о том, что руководитель проекта определяет верхний уровень *иерархии структуры работ* (ИСР), или *WBS* (от англ. *work breakdown structure*). Дальнейшую детализацию выполняет при его участии команда проекта.

На практике часто приходится сталкиваться с ошибочным подходом, когда функция декомпозиции проекта частично или полностью переключается на других исполнителей. Однако, во-первых, руководитель проекта по основным требованиям к «талантам» и опыту является самым приемлемым кандидатом для данной функции — иначе ему нельзя доверять руководство проектом, а во-вторых если у него эту функцию забрать, то он лишается необходимых ему прав на принятие самых важных проектных решений. Таким образом, с него снимается и ответственность за проект.

Следует отметить, что *PMBOK* данная функция описывается в основном как интеграция проекта — поддержание его целостности и взаимосвязи временных, ресурсных и технических компонент в ходе всей реализации. Однако до того как поддерживать компоненты проекта в необходимом взаимодействии, их сначала необходимо выделить.

В рамках данного пособия под *декомпозицией* проекта будем понимать не только выделение, но и поддержание целостности компонент и их взаимодействия на протяжении жизненного цикла проекта.

Кроме того, следует отметить, что в данном случае мы говорим именно о высокоуровневом планировании. Такие планы называются *контрольными*, или *базовыми*. Они являются элементами «договора» руководителя проекта с заинтересованными сторонами (участниками, стейкхолдерами, от англ. *stakeholders*) и могут изменяться только по согласованию всех заинтересованных сторон. Ясно, что руководитель проекта, который не имеет необходимого опыта и квалификации, не сможет заключить недискриминирующий его «договор».

Детальные планы выполнения работ, называемые *рабочими*, могут составляться не руководителем проекта, а экспертами и исполнителями в ходе реализации проекта. Они согласовываются только исполнителями и рабочей группой и могут меняться без согласования с заинтересованными сторонами. В совокупности базовый план и рабочие планы образуют общий план проекта. Подробнее планирование проекта будет описано в отдельном разделе. Здесь отметим, что на практике оптимальным периодом между контрольными точками является месяц, а длительность работ в рабочих планах — две недели¹.

3. Достижение контрольных точек проекта. Руководитель проекта, согласовывая декомпозицию проекта, предлагает заинтересованным сторонам инструмент стороннего контроля реализации проекта в виде контрольных точек. Это необходимый элемент его договоренности с заинтересованными сторонами. Достижение контрольных точек проекта входит в его базовые функции и набор обязательств. Не выпол-

¹ Верзух Э. Управление проектами: ускоренный курс по программе MBA.

няя данную функцию, руководитель проекта не только увеличивает риски проекта, но и лишает заинтересованные стороны единственного эффективного механизма контроля проекта. Он, таким образом, отключает внешний контроль, что не является допустимым даже при условии его высокой приверженности целям проекта и подтвержденной квалификации.

4. Управление командой проекта. Команда проекта или иначе рабочая группа проекта — основной концентрированный исполнительский ресурс проекта. Современные проекты не могут быть реализованы в одиночку. Реальные ИТ-системы давно перешагнули в своей сложности возможности малых коллективов. Именно поэтому функция управления командой проекта является центральной для руководителя проекта. Без команды единомышленников-соисполнителей руководитель проекта не справится, скорее всего, даже с планированием задач. Их исполнение точно окажется за пределами его возможностей.

Под *управлением командой проекта* понимается ее подбор, планирование задач отдельных участников, контроль исполнения, поддержание целостности команды, обеспечение регулярной коммуникации, донесение до команды представления о видении целей проекта, мотивирование участников, обучение и развитие команды проекта, разрешение конфликтных ситуаций.

Важно понимать, что команда проекта, несмотря на то, что часто является временной и сборной, должна быть единой и эффективно взаимодействующей. Как уже говорилось, проекту противодействует инерционность организации, которая, как и любая живая сложная система, сопротивляется изменениям. Поэтому неэффективно работающая, «рыхлая» команда проекта просто не справится с сопротивлением среды и не сделает проект успешным.

Успех проекта более чем наполовину зависит от эффективно работающей команды проекта. Эффективные команды не складываются сами собой. И совершенно точно команда проекта — это не тот вопрос, который может быть отпущен на волю случая и к которому можно подходить формально.

Проблеме формирования высокоэффективных рабочих коллективов посвящено достаточно много специальной литературы. Исходя из практики реализации крупных проектов в российских организациях, выделим основные моменты, на которые следует обратить внимание при организации работы команды проекта.

На практике часто встречается ситуация, при которой в команду проекта формально включают представителей всех заинтересованных сторон, подразделений-исполнителей и даже подразделений, которые могут быть только потенциально заинтересованы в проекте. При этом данные представители выполняют скорее наблюдательные и контрольные функции, а не исполнительские. Таким образом, основной исполнительский ресурс, подчиненный руководителю проекта и содействующий ему, подменяется на противодействующий. В данном случае

проект, скорее всего, окажется неуспешным, а руководителю проекта еще и поставят в вину неэффективное управление большой по количеству «выделенной ему в подчинение» командой проекта, которой в реальности он не имел возможности руководить и которая просто поглотила весь его личный рабочий ресурс.

В подобных случаях руководитель проекта должен задействовать свои навыки и знания для «зачистки» команды проекта. Эффективным инструментом для этого является первоочередное распределение ответственности за проектные задачи — каждый участник команды проекта должен получить релевантную (служашую целям проекта) задачу под свою ответственность. Если появляется участник команды, все функции которого сводятся к участию в согласовании, содействии и т. д., то данный участник должен быть исключен. Эти функции будут выполнены в рамках текущей деятельности взаимодействующими подразделениями организации.

Еще один важный момент, на который следует обратить внимание, — это необходимость исключения из команды проекта «токсичных» участников. «Паршивая овца» способна испортить работу всей команды. Часто, несмотря на наличие высокой специальной квалификации, такой участник в силу особенностей личности систематически снижает эффективность работы, «торпедируя» принятие любых коллективных решений и оказывая негативное воздействие на участников проекта.

Для анализа причин подобной «токсичности» может быть повторно предложено обратиться к классификации психологических радикалов, приведенных в работе В. В. Пономаренко¹. Представьте, что в вашем подчинении как руководителя проекта обнаруживается ярко выраженный «эпилептоид» или «истероид». В первом случае каждое совещание или другая коммуникация с его участием будет превращаться в выяснение уровня подчиненности, а во втором — в «театр одного актера».

Никакие навыки и знания по управлению людьми в этом случае руководителю проекта не помогут. У него нет избытка временного и личного ресурса на корректировку поведения участников рабочей группы, он не специалист по психологическим отклонениям. Самый правильный подход в этом случае — воспользоваться своим правом на определение состава команды проекта и заменить «токсичного» участника.

При создании команды проекта и организации ее работы важным вопросом является вопрос о ее численности. На практике встречаются формально создаваемые команды численностью несколько десятков и даже сотен человек. Командой такое образование назвать достаточно сложно. Часто путают команду проекта и список сотрудников, которые должны быть дополнительно премированы по результатам проекта.

¹ Пономаренко В. В. Практическая характерология с элементами прогнозирования и управления поведением (методика «семь радикалов»).

Никто не мешает организовать мероприятия по дополнительному поощрению сотрудников, принявших участие в реализации успешного проекта, после его завершения, но их список не образует команду проекта.

Следует отличать команду проекта от списка людей, принимающих участие в выполнении отдельных работ в интересах проекта. Команда проекта — это эффективно взаимодействующий коллектив специалистов-единомышленников, вовлеченных в проект в виде своей основной деятельности. Их уровень вовлечения приближается к 100 %. В то время как уровень вовлеченности в отдельный проект остальных исполнителей колеблется вокруг единиц процентов годового рабочего времени.

На практике в большинстве случаев следует поддерживать численность команды проекта в 8—10 человек. Такая численность диктуется ограничениями человеческого мозга и его способностью эффективного прямого управления коллективом (см. гл. 2).

Безусловно, проекты различаются масштабом и сложностью. Встречаются проекты, где достаточной является численность команды в 3—5 человек. Существуют проекты, где недостаточной численностью команды будет и 100 человек. Во втором случае необходимо переходить к иерархической организации работы, но иерархическая организация подразумевает и смещение от термина «команда». Уже на втором уровне иерархии деятельность участников будет скорее не проектной, а текущей. Таким образом, команда проекта срачивается с иерархическим построением организации и группы организаций, используя их ресурсы для реализации проекта.

Альтернативным подходом к организации работ по крупным проектам является разбиение проекта на последовательно реализуемые этапы без чрезмерного расширения его команды.

5. Определение заинтересованных сторон проекта. К заинтересованным сторонам, или участникам проекта относят сотрудников и руководителей, принимающих участие в проекте или затрагиваемых его результатами. Они участвуют в определении целей проекта, его содержания, в исполнении работ, выделении ресурсов на проект. Задача корректного определения состава участников проекта является крайне сложной¹. Данному вопросу посвящен не один десяток страниц *PMI PMBOK*. При этом «задачей» этот комплексный процесс назван не совсем корректно. Это не однократное выполняемое действие, а непрекращающийся набор действий, выполняемых командой проекта и руководителем проекта в ходе его жизненного цикла. В связи со сложностью и потребностью в задействовании специальных компетенций эта функция включена в список базовых функций руководителя проекта.

Типовые роли участников проекта приведены на рис. 6.3.

Традиционно выделяют следующие основные роли: заказчик, спонсор (в отечественной терминологии — куратор), руководитель проекта, команда проекта и исполнители.

¹ Верзух Э. Управление проектами: ускоренный курс по программе MBA.

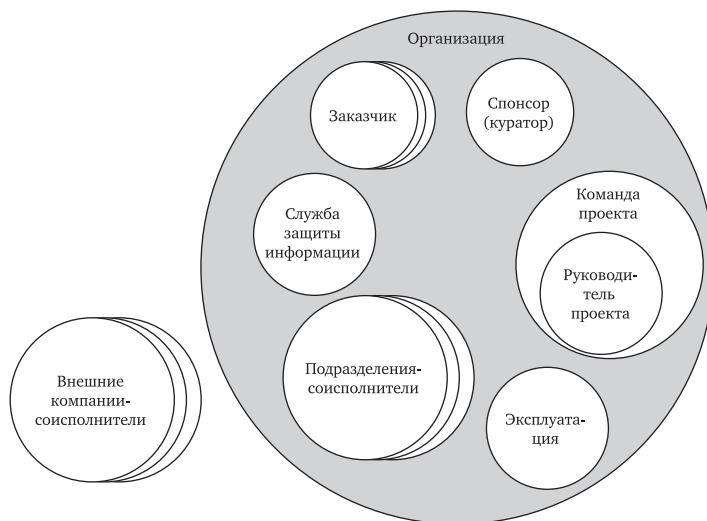


Рис. 6.3. Роли участников проекта

На практике выделяемые роли могут совмещаться в лице одного руководителя или подразделения. Необходимо отметить, что без наличия достаточных оснований совмещение ролей является крайне нежелательным (например, служба защиты информации + основной заказчик). Основные функции ролей начинают исполняться в ущербном виде с некорректным определением приоритетов и нарушением естественного конфликта интересов.

Самое негативное воздействие на проект обычно оказывает совмещение ролей заказчика и руководителя проекта. В этом случае практически полностью теряется функция контроля качества результата проекта. Основная принимающая сторона оказывается основным исполнителем и принимает результат работ сама у себя.

Проектная деятельность не только сталкивается с естественным сопротивлением среды новому, но и, представляя собой «организационную аномалию», нарушающую обычный ход текущей деятельности и иерархическое построение организаций, встречается с трудностями двойного подчинения человеческих ресурсов.

Часто уровня полномочий руководителя проекта недостаточно для разрешения организационных противоречий. В этом ему должен помочь спонсор проекта. Кроме того, спонсор отвечает за выделение проекту необходимого объема финансовых, человеческих и других ресурсов. Понятно, что данное лицо должно располагать достаточным уровнем властных полномочий в организации.

Власть спонсора не привязана к отдельному проекту. Это команда отдельного проекта пользуется «административным ресурсом» спонсора, источником которого является его положение во властной иерархии. Здесь стоит обратить внимание на первую строчку табл. 5.3, в которой представлены основные факторы успешности проек-

тов. По результатам анализа международной статистики, поддержка руководства (англ. *executive support*), обеспечивает около 15 % вклада в успех проекта. Это максимальный уровень вклада из списка, выделенного статистикой.

Обычно под заказчиком понимают лицо или подразделение, определяющее цели и содержание проекта, а также обеспечивающее его финансирование. Коротко — «кто платит, тот и заказывает музыку».

На практике встречается ситуация, когда финансирование обеспечивается не зарабатывающим подразделением, а, например, бэк-офисом Банка, обеспечивающим рутинную обработку потока банковских транзакций. В этом случае финансирование проекта может обеспечиваться ими за счет будущего сокращения затрат на выполнение своих функций. То есть источником финансирования может выступать не доход подразделения, а достигаемая им экономия затрат.

Следует обратить внимание на то, что заказчиков в проекте обычно несколько. Современные системы, даже узкоспециализированные, практически всегда предоставляют набор сервисов более чем одному подразделению. При этом очень часто заказчик, первично инициировавший проект, стремится исключить других заказчиков из состава участников в целях приоритетного соблюдения своих интересов. Он обычно не заинтересован в согласовании своих интересов и целей с другими заказчиками. Начинаящий руководитель проекта также заинтересован в упрощении своей работы по согласованию содержания проекта. В результате проект столкнется с необходимостью уточнения содержания проекта на более поздних стадиях реализации, что приведет к необходимости корректировки и исправления ошибок при попытке сдачи «готовой» системы.

Опытный руководитель проекта знает, что стоимость исправления застарелых ошибок в десятки и сотни раз превышает стоимость своевременного исправления, поэтому профессионал не позволит необоснованно сократить список заказчиков.

Корректным подходом при работе с множественными заказчиками является организация их совместной согласованной работы с выделением единого уполномоченного. В данном вопросе определяющую поддержку может обеспечить спонсор проекта.

Контекст данного пособия задается информационными технологиями, поэтому список типовых ролей несколько отличается от канонического, приводимого, например, в *PMI PMBOK*; основное отличие в том, что выделена роль службы защиты информации (СЗИ).

С одной стороны, СЗИ является соисполнителем по любому информационно-технологическому проекту. С другой стороны, это подразделение играет роль принимающей стороны — участвует в процедуре контроля результата проекта при его приемке. А если учесть, что современные ИТ-системы обычно содержат в своем составе функционал защиты информации, то данное подразделение оказывается и заказчи-

ком проекта. Это же относится к подразделениям, сведенным под наименованием «эксплуатация».

6. Обеспечение принятия технических решений проекта. На практике часто приходится встречать подход, при котором руководителем проекта назначают руководителя бизнес-направления, сотрудника с недостаточными техническими знаниями. В этом случае происходит смешение понятий «спонсор» и «руководитель проекта» или «заказчик» и «руководитель проекта». То есть проект теряет реального руководителя проекта, и в связи с необходимостью выполнения большого объема административной работы, специфической для руководителя проекта, на эту роль назначают технического исполнителя, полностью подчиненного «узурпировавшим власть» неспециалистам в ИТ-области.

Таким образом, проект получает спонсора или заказчика, занятых не свойственной им работой, и неполнофункционального руководителя проекта. Это увеличивает уровень риска проекта до неприемлемого уровня. Спасти такой проект может только профессионал высокого уровня, случайно оказавшийся в команде проекта и сумевший без надления достаточными официальными полномочиями привести проект хоть к какому-то результату.

Почему-то руководителями проектов строительства мостов не назначают неспециалистов. При этом, вероятно, в связи с глубоким проникновением информационных технологий в обыденную жизнь, считается возможным и обоснованным возлагать руководство ИТ-проектами на неспециалистов.

Информационные технологии относятся к категории наиболее сложных инженерных областей. Затраты на информационные технологии в современном мире превышают затраты на энергетику. При этом значительная часть затрат оказывается под управлением неспециалистов. Безусловно, это не является достаточно обоснованным.

Руководитель проекта в области информационных технологий должен обладать необходимой квалификацией именно в этой области. Для чего она нужна?

Во-первых, руководитель проекта должен выполнить определяющий объем работ по верхнеуровневой декомпозиции создаваемой или модернизируемой системы, отдельных проектных задач и планированию основных сроков. Провести декомпозицию системы, связать ее компоненты с задачами, сроками и ресурсами, а затем поддерживать интеграцию этих составляющих проекта невозможно без высокой технической компетентности. Если эта задача передана по частям архитектору, руководителю разработки и другим участникам команды проекта, то баланс между «идеальным» проектным решением и его реализуемостью в срок практически всегда оказывается нарушен.

Во-вторых, ранее было показано, что реальный ИТ-проект связан с конфликтом множественных интересов большого количества участников. Руководитель проекта должен эффективно выполнить арбитраж конфликтующих точек зрения, обоснованно выработать техническое

проектное решение и согласовать его с заинтересованными сторонами. Передав эту функцию кому-либо, руководитель проекта отказывается от своих основных обязанностей и, соответственно, теряет большую часть властных полномочий.

6.3. Баланс прав и ответственности руководителя проекта

Обсуждая особенности деятельности руководителя проекта и его функции, мы неоднократно упоминали необходимость соблюдения баланса между его правами и ответственностью. Любое нарушение этого баланса приводит либо к потере управляемости проекта со стороны организации, либо к утрате возможности руководителя проекта реально влиять на проект. И в том, и в другом случае риски проекта возрастают до неприемлемого уровня.

Основными балансируемыми параметрами являются:

- право принимать принципиальные проектные технические решения для достижения целей проекта (например, выбор закупаемых компонент) и обязанность их согласовывать с заинтересованными сторонами;
- право распоряжаться выделенными на проект ресурсами и обязанность предоставить организации эффективное средство контроля за их расходованием;
- право выполнения декомпозиции систем, задач, работ и ответственность за поддержку их взаимосвязи и интегрированности;
- право утверждать детализированные рабочие планы и ответственность за предоставление результатов по контрольным точкам;
- право контролировать исполнение отдельных работ и ответственность за организацию прозрачных и объективных процессов контроля качества при приемке крупных результатов проекта по контрольным точкам с привлечением всех заинтересованных сторон;
- ответственность за достижение целей проекта и право участвовать в согласовании этих целей, выделяемых ресурсов и устава проекта в целом при его инициации.

Безусловно, балансируемых параметров гораздо больше. Здесь приведены основные. Важно понимать, что проектная деятельность основывается на балансе двух основных участников проекта — руководителя проекта и организации, заинтересованной в его реализации. Их «договор» должен быть справедливым и не должен быть дискриминирующим в отношении одной из сторон.

Необоснованно занижая права руководителя проекта, мы избавляем его от ответственности и, формально создавая эту ключевую роль, на практике ее сразу уничтожаем.

Необоснованно завышая права руководителя проекта, мы ставим под увеличенный риск организацию в целом. В организационной структуре, как в любой сложной системе, есть предел возможной способности адаптироваться к изменениям и затратам ресурсов.

Глава 7

УСТАВ ПРОЕКТА

Реализация проекта должна начинаться с подписания устава проекта. Часто этот документ в российской практике называют паспортом проекта. В рамках данного пособия будем использовать закрепленное в *PMI PMBOK* наименование — устав проекта.

В литературе¹ и в стандарте *PMI PMBOK* под **уставом проекта** понимается формальное объявление о старте проекта и наделении руководителя проекта полномочиями по его исполнению. При этом содержится достаточно много рекомендаций по содержанию документа и процедуре его согласования.

Более корректное понимание заключается в том, что устав проекта — это своеобразный многосторонний договор участников проекта о его целях, ресурсах и исполнении. Для руководителя проекта — это договор с организацией в целом о его правах и обязанностях; для заказчика — о целях проекта, стоимости и сроках; для спонсора (куратора) — о его обязанностях по поддержке проекта, выделяемых ресурсах и правах по контролю. Для всех остальных заинтересованных сторон — определение их требований к проекту и обязанностей по участию в работах.

В случае реализации проекта с привлечением внешнего подрядчика устав трансформируется в полноценный договор.

С такой точки зрения содержание и процедуры согласования устава становятся легко определяемыми и целостными для любого типа организации и уровня зрелости.

Содержание устава определяется необходимостью закрепления прав и ответственности сторон, а полномочия участников по участию в согласовании отдельных пунктов — балансом их прав и ответственности.

Естественно, все изменения в устав в ходе реализации проекта (необходимость этого, как правило, обычное явление) вносятся по согласованию со всеми участниками проекта.

Как и любой договор, устав должен содержать набор обязательной информации:

- наименование проекта;

¹ См., например: Верзух Э. Управление проектами: ускоренный курс по программе МВА.

- стороны — участники проекта (стейкхолдеры): спонсор, заказчики, руководитель проекта, соисполнители, служба защиты информации, эксплуатирующие и контролирующие подразделения;
- цели проекта, приоритеты и критерии оценки результата;
- ограничения проекта;
- срок реализации проекта и сроки предоставления промежуточных результатов (контрольный или базовый план проекта);
- бюджет проекта;
- команда проекта, передаваемая в подчинение руководителю проекта;
- обязательства соисполнителей проекта по выполнению работ в рамках проекта;
- оговорки о рисках проекта и управлении ими.

Часто путают устав проекта и содержание проекта (англ. *statement of work, SOW*). Понятно, что если мы рассматриваем устав в качестве договора, то только описание того, что должно быть сделано, не является достаточным. Поэтому под содержанием проекта будем понимать соответствующую часть устава (цели проекта и критерии оценки результата, ограничения проекта, контрольный или базовый план проекта).

Если в организации исполнение проектов поставлено на поток, то в ней, скорее всего, в виде регламента существует набор договоренностей, общих для всех проектов; тогда устав отдельного проекта может содержать несколько меньший объем информации. Например, права и обязанности службы защиты информации обычно стандартизированы и могут быть закреплены в общем регламенте. Набор рисков ИТ-проектов также почти всегда постоянен для всех реализуемых проектов и может быть выведен в регламент из отдельных уставов. Кроме того, целесообразным является выведение в регламент типовых прав и обязанностей руководителя проекта, спонсора и заказчика, а также типовых требований к проведению приемочных процедур.

Управление участниками и командой проекта описаны в характеристиках соответствующих функций руководителя проекта (см. гл. 6). Более подробно остановимся на других основных разделах устава.

7.1. Цели и приоритеты проекта

Заголовок данного раздела сознательно сформулирован во множественном числе. Реальные проекты всегда преследуют несколько целей. У промышленных систем в большинстве случаев оказывается много пользователей, а у соответствующих проектов — более одного заказчика. Кроме того, свои интересы при формулировании целей отстаивают и другие заинтересованные стороны.

Однако для этих целей всегда должны быть установлены приоритеты достижения. Иначе при принятии проектных решений будет крайне

тяжело достичь компромисса, а руководитель проекта при их согласовании лишится действенного механизма для отстаивания своего мнения.

Например, цели проекта могут быть сформулированы как достижение некоторой дополнительной функциональности, производительности и замещение существующей системы в связи с отказом поставщика от сопровождения и развития системы (такой случай встречался в практике автора)¹. В связи с постоянным конфликтом целей (замещение требует в минимальный срок, а наращивание функциональности этот срок удлиняет) проектные решения на протяжении нескольких лет принимались с задержками, базируясь на плавающих приоритетах и недостаточно обоснованно. Спустя четыре года проект пришлось полностью переформатировать. Первый приоритет был придан цели «замещение», после чего проект был реализован за полтора года. Результаты работ первых четырех лет были полностью списаны как бесполезные.

Международная статистика ИТ-проектов утверждает, что стоимость устранения ошибок, допущенных при определении требований к автоматизированным системам, примерно в 200 раз выше, чем затраты на устранение ошибок, допущенных в завершающие этапы проекта. Какова стоимость устранения ошибок, допущенных при определении целей проекта? Часто она равна или превосходит сумму бюджета проекта. Поэтому в первую очередь подробно рассмотрим вопрос уточнения целей проекта и проблемы при их определении.

Информационные технологии характеризуются высочайшим уровнем сложности (см. гл. 2), следовательно, точно определить цели проекта бывает крайне затруднительно, если вообще возможно. Это ограничение человеческого разума следует всегда принимать во внимание.

При этом категорически нельзя начинать проект, не предприняв все возможные усилия для того, чтобы цели проекта стали ясны всем участникам и были однозначно ими поняты. Поэтому основная рекомендация при формулировке целей выглядит следующим образом: необходимо максимально упрощать цели проекта до уровня, когда они становятся максимально различимы всеми участниками проекта. Именно данной рекомендации соответствует подход «разбиения кита на дельфинов» и концепция *MVP*.

При формулировке целей следует не только максимально упрощать требуемый результат, но и стараться предельно приближать по времени срок его получения и четко привязывать этот срок к результату. Долгосрочная перспектива «размывает» нашу способность видеть будущий результат. Кроме того, удлинение срока приведет к тому, что требования к результату проекта начнут меняться в связи с изменением внешних условий, например, требований регулирующих государственных органов и периодически сменяющегося руководства организации. Для ИТ-проектов обычной различимой перспективой является 4—6 меся-

¹ Здесь пока не указываем на другие недостатки подобной формулировки.

цев. Далее требования к результату начинают изменяться независимо от проекта.

По формулировке целей существует широко распространенная концепция SMART (табл. 7.1).

Таблица 7.1

Концепция SMART

Показатели	Характеристики
<i>Specific</i> (точная, ясная)	Необходимо убедиться, что цель ясна всем участникам проекта. Если кто-то из заинтересованных сторон формулирует цель, то он должен озаботиться тем, что она понятна для всех остальных участников, даже если она относится к специальной области деятельности. Здесь же формулируются и критерии, которые позволяют однозначно и объективно доказать, что цель достигнута. Недопустима формулировка цели с такими определениями, как «универсальный», «гибкий», «хороший», «высокий уровень» и им подобными
<i>Measurable</i> (измеримая)	Данная характеристика подкрепляет первую. Характеристики цели должны быть измеримы с применением общепринятых инструментов и методик. Допустима, например, цель «рост производительности на 20 %» или «снижение стоимости владения до 1 млн руб. в год». Недопустимы такие цели, как «снижение стоимости владения»
<i>Achievable</i> (достижимая)	Достижимость цели определяется наличием соответствующих ресурсов и характеристиками организации. В гл. 5 обосновывалась невозможность реализации масштабных сложных проектов в организации с низким уровнем зрелости. Даже при наличии ресурсов сложная цель не будет достигнута. Необходима адекватная оценка своих сил, знаний, ресурсов, эффективности процессов и других, внешних для проекта условий. Здесь в очередной раз может быть рекомендовано использование концепции MVP
<i>Relevant</i> (релевантная, актуальная)	При формулировке цели необходимо периодически задавать вопрос «Зачем это надо?» Цель проекта или отдельного этапа проекта должна быть связана с целями организации более высокого уровня. Например, цель проекта «увеличить производительность системы на 20 %» может быть связана с бизнес-целью «нарастить клиентскую базу на 20 %». А цель промежуточного этапа с контрольной точкой проекта «утвердить рабочий проект на создание системы А» служит цели проекта «внедрить систему А в промышленную эксплуатацию»
<i>Time bound</i> (привязанная к сроку)	Современные системы развиваются в ускоряющемся темпе. Достижение каких-либо характеристик автоматизированной системы несвоевременно означает недостижение цели проекта. Характеристики цели изменяются во времени, и поэтому если вы задали специфическую, измеримую, достижимую и актуальную цель на 1 января 2018 г., то все ее так хорошо сформулированные характеристики сохраняются именно на эту дату, а не на 1 января 2019 г.

7.2. Планирование и контроль проекта

В проектном управлении различают два основных вида планов: базовый (контрольный) план и рабочий план. Планы отличаются уровнем детализации. Базовый — верхнеуровневый. Рабочий — представляет отдельные простейшие работы и мероприятия.

Цели составления этих планов также различны. Базовый план служит для фиксации договоренностей между участниками проекта о сроках предоставления руководителем проекта результатов, которые могут быть распознаны всеми заинтересованными сторонами. Соответствующие контрольные результаты и даты образуют собой контрольные точки («вехи», или *milestones*), поэтому его часто называют контрольным.

Таким образом, базовый план является частью договора участников проекта и, соответственно, может быть изменен ими только совместно. Он содержит информацию о контрольных точках и является единственным механизмом эффективного объективного стороннего контроля.

Рабочий план составляется руководителем проекта и участниками для координации работ, расчета ресурсов, планирования второстепенных мероприятий. У руководителя проекта есть полное право изменять рабочий план в своих целях для координации мелких работ отдельных исполнителей, пока сохраняется его соответствие базовому плану в контрольных точках.

Поскольку обеспечивать поставку распознаваемых результатов проекта, которые могут быть подвергнуты процедурам объективной проверки качества и полноты, с высокой частотой достаточно затруднительно, то считается, что целесообразно обеспечивать периодичность появления контрольных точек в базовом плане (ориентировочно — ежемесячно). Рост количества контрольных точек приводит к избыточным затратам на производство контрольных результатов и выполнение контрольных процедур. Кроме того, базовый план становится слишком объемным для того, чтобы быть быстро согласованным всеми заинтересованными сторонами. Чем больше документ, тем дольше его согласовывают.

Исходя из ежемесячной периодичности базового плана, можно легко спрогнозировать реальный объем такого документа. Полуторалетний проект, например, будет иметь базовый план, состоящий ориентировочно из 20 пунктов (контрольных точек).

Следует обратить внимание на то, что в базовом плане не обязательно указывать начальную дату реализации мероприятий, которые должны завершаться появлением контрольного результата. Их координация и планирование — это ответственность руководителя проекта. Целью базового плана не является координация отдельных работ. Чем больше в нем будет подобных деталей, тем в большей степени сужается поле принятия решений для руководителя проекта. Баланс прав и ответственности искажается. Руководитель отвечает за поставку результата

в контрольные точки и, соответственно, должен иметь право принятия решений о том, как он это обеспечит.

Часто выделение распознаваемых контрольных результатов для указания в базовом плане становится достаточно сложной задачей, особенно в организациях с низким уровнем зрелости процессов. Качество и полнота контрольного результата должны быть подтверждены незаинтересованной и достаточно квалифицированной стороной или участником проекта, цели которого не совпадают с целями проектной команды. В большинстве организаций, реализующих ИТ-проекты, этим участником является служба защиты информации и подразделение, осуществляющее эксплуатацию будущей системы, — они не заинтересованы в поставке недоработанного продукта (он увеличивает их риск) и располагают необходимой квалификацией для его контроля. Безусловно, более корректным является выделение для этих целей подразделения, специализирующегося на контроле качества и тестировании.

Если у вас в организации не существует утвержденных регламентов, содержащих требования к содержанию и оформлению проектных документов, то процедура контроля готовности, например, технического задания может стать достаточно ресурсозатратной. У каждого из участников проекта оказывается свое мнение о составе и качестве документа. Учитывая разницу в их целях, можно сказать, что в этом случае реализация проекта становится похожей на «хождение по граблям», а его сроки будут, скорее всего, превышены.

Если руководство не сделает соответствующих выводов по опыту реализации проекта, столкнувшегося с подобными затруднениями, то и каждый последующий проект превратится в аналогичную каторжную процедуру. Если порядок реализации проектов в организации, во избежание данных сложностей, будет предусматривать контроль результата только проектной командой и руководителем проекта (например, согласование и утверждение документов внутри проектной команды), то необходимого для эффективного контроля конфликта интересов не будет, ценность такого контроля будет крайне низкой и базовый план перестанет выполнять свою функцию. Каждая контрольная точка в этом случае станет жертвой «правила 10—50—90» (если руководитель проекта говорит, что приступил к работе, то, скорее всего, даже не видел задачу, если говорит, что сделал 50 %, то, скорее всего, только начал, если говорит, что заканчивает работу, то, вероятно, находится в середине пути). Именно поэтому такую низкую контрольную ценность имеют самостоятельно подготавливаемые контрольные периодические отчеты руководителя проекта. Информационную контрольную ценность имеют только результаты, подтвержденные независимой и квалифицированной стороной.

При составлении базового плана можно руководствоваться четырьмя основными подходами: экспертная оценка по аналогам, от целевой даты результата проекта, «набегающей волной», на основе выполнения декомпозиции работ (WBS).

Вероятно, самым эффективным и часто применяемым на практике является первый подход — руководитель проекта, имеющий необходимые знания и опыт реализации аналогичных проектов, пусть даже в несколько отличных условиях, может достаточно точно задать укрупненные плановые даты промежуточных контрольных результатов. При этом он может сделать необходимые поправки в соответствии с его оценкой сложности целевой автоматизированной системы (см. гл. 2).

Также на практике мы вынуждены часто применять планирование от целевой даты. Если, например, центральный банк изменяет требования к ведению внутреннего учета по ценным бумагам с первого января следующего года, то коммерческий банк будет вынужден уложиться с проектом в отведенный срок. В этом случае руководитель проекта, получив в качестве жесткого условия конечную дату, в целях сбалансированности проекта будет манипулировать не сроками, а ресурсами проекта, пытаясь уложить все работы в заданный срок. Умение составить реализуемый план, отшагивая от целевой даты, является, безусловно, необходимым для профессионального руководителя проекта.

Планирование «набегающей волной» сходно с *agile* — подходом и в некотором смысле им и является. Отличие в том, что *agile* дополнительно предусматривает получение при завершении каждой «волны» полезного результата (*MVP*). Таким образом, в случае *agile* короткими «волнами» не только планируются работы, но и проектируется *MVP* как индивидуальный результат, а не промежуточная контрольная точка.

Применяя данный подход, мы планируем только те промежуточные результаты и работы, которые отчетливо различимы и предсказуемы. То есть планированию подлежат только работы на ближайшую перспективу, например, на один месяц. В этом случае говорим, скорее, уже о рабочем плане. На уровне базового плана задаем укрупненные этапы, а на уровне рабочего плана «набегающей волной» детализируем уровень простейших работ. Планирование отдаленных этапов не выполняется или выполняется только по крупным контрольным точкам.

При составлении базового и рабочего планов широко применяется формализованный подход, основанный на многоуровневой декомпозиции результата и отдельных работ — иерархическая структура работ (*ИСП*, *WBS*). Этот подход исходит из того, что перед стартом работ по проекту весь их состав может быть детально запланирован на уровне элементарных операций, а результат сколь угодно сложного проекта является полностью определенным как по полноте, так и по показателям качества. Это допущение не является верным в случае, когда мы пересекаем границу сложности (см. гл. 2). И результат, и состав работ не являются полностью предсказуемыми в случае сложных проектов. Реализуя проект, мы постоянно движемся в сторону уточнения цели и состава работ, достигая полной определенности только в момент утверждения акта сдачи-приемки и завершения проекта.

Однако для случая простых коротких проектов подход ИСР является полностью оправданным. Кроме того, он, безусловно, полезен в качестве элемента детального планирования очередной волны или в качестве прикидочного для проверки установленных контрольных точек базового плана. То есть для построения базового плана мы можем выполнить декомпозицию работ по одному или нескольким его этапам для проверки реалистичности и потребности в ресурсах.

Отдельные работы, возлагаемые на конечных исполнителей в проекте, могут быть детализированы до элементарных (неделимых). Они — продукт декомпозиции систем и работ. На практике принимают эффективную длительность мероприятий рабочего плана в 1—2 недели. Обратите внимание на то, что эта длительность соответствует рекомендуемой SCRUM-методом длительности спринтов.

Поскольку рабочий план служит для планирования и координации множества отдельных работ, то его объем в реальных проектах может быть достаточно большим. Количество отдельных работ в рабочих планах измеряется сотнями и тысячами. Например, в проекте длительностью 1,5 года с количеством отдельных исполнителей 10 человек, выполняющих свои работы двухнедельными «спринтами», количество пунктов рабочего плана составит не менее 300 (здесь учтено то, что длительность реальных элементарных работ составляет от недели до 2 и что исполнители по проекту не всегда выполняют работу непрерывно).

С рабочим планом и ИСР связано важное для проектного управления понятие **критического пути** (англ. *critical path method, CPM*). Так называют инструмент планирования проекта, который основан на определении наиболее длительной последовательности задач от начала проекта до его окончания с учетом их взаимосвязи.

Задачи, лежащие на критическом пути (критические задачи), не имеют резерва времени выполнения, и в случае изменения их длительности изменяются сроки всего проекта. Поэтому часто критический путь определяют как состав задач с нулевым резервом исполнения. Длительность их исполнения определяет длительность проекта. Критические задачи требуют непрерывного контроля с точки зрения прогнозирования проблем, влияющих на сроки их выполнения. В сложных проектах критический путь может неоднократно меняться в связи с тем, что какие-то работы изменили свою длительность в результате уточнения планового срока. Также понятно, что в случае сокращения сроков работ, находящихся на критическом пути, например, в два раза, общая длительность проекта совсем необязательно будет сокращена в два раза.

Критический путь может быть определен только на основе детализированного планирования элементарных операций, и точность его определения полностью зависит от корректности декомпозиции продукта по компонентам и задач проекта по элементарным работам, что часто является затруднительным.

При планировании используют разнообразные виды диаграмм. На рис. 7.1 представлены основные типы сетевых диаграмм: работы в узлах и работы в дугах.

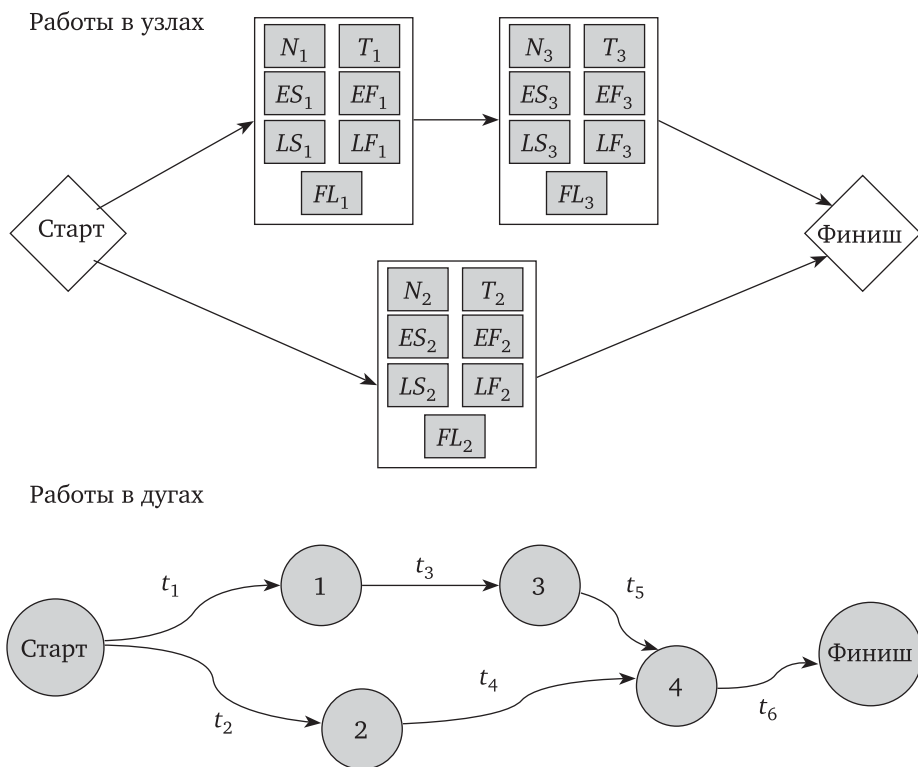


Рис. 7.1. Сетевая диаграмма с работами в узлах и сетевая диаграмма с работами в дугах

На диаграмме (рис. 7.2) возможны два пути от старта к финишу проекта: первый путь S_1 через вехи 1, 3, 4 или второй путь S_2 через вехи 2, 4. Длительность t пусть задана в днях. Соответствующая суммарная длительность работ (контрольные точки (вехи) не имеют самостоятельной длительности) составляет:

- для пути $S_1 = t_1 + t_3 + t_5 + t_6 = 1 + 2 + 1 + 3 = 6$ дней;
- для пути $S_2 = t_2 + t_4 + t_6 = 3 + 4 + 3 = 10$ дней.

Следовательно, критическим является путь S_2 .

На диаграмме (рис. 7.1) с работами в узлах приведены следующие обозначения, несущие дополнительную информацию:

- N — номер задачи или работы в проекте;
- T — время, необходимое для выполнения задачи;
- ES (*early start*) — дата, раньше которой нельзя начать выполнение работы, учитывая сроки выполнения предшествующих задач;
- EF (*early finish*) — дата, раньше которой нельзя завершить выполнение задачи, учитывая сроки выполнения предшествующих задач;

- *LS (late start)* — дата, позднее которой нельзя приступить к выполнению задачи, учитывая сроки завершения проекта;
- *LF (late finish)* — дата, позднее которой нельзя завершить задачу, учитывая сроки завершения проекта;
- *FL (floating days)* – разница между датами *ES* и *LS* образуют резерв времени на работу для отдельной задачи.

Различные диаграммы удобны для решения разных задач в ходе планирования проекта. Они представляют информацию о взаимосвязи отдельных работ и их количестве, а также контрольных точках или «вехах» проекта. На рис. 7.2 представлена иллюстрация определения критического пути по диаграмме «работы в дугах».

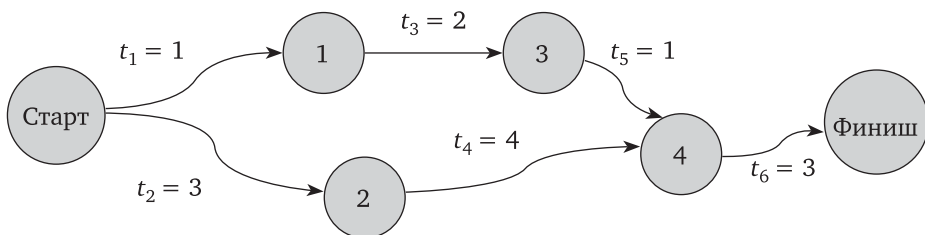


Рис. 7.2. Сетевая диаграмма с работами в дугах для вычисления критического пути

Для иллюстрации определений и их вычисления рассмотрим упрощенный проект «Разработка и внедрение автоматизированной системы «Домашний банк»». Для достижения цели проекта необходимо выполнить следующие задачи (табл. 7.1, рис. 7.3).

Таблица 7.1

Разработка и внедрение автоматизированной системы «Домашний банк»

Задача	Время исполнения (дни)	Исполнитель
1. Разработка требований	30	Технолог
2. Разработка программного обеспечения	40	Разработчик
3. Установка системы	5	Системный администратор
4. Разработка и согласование договорной базы для юридического оформления подключения клиентов к системе	30	Юрист

Задачи 1, 2, 3 должны выполняться последовательно, хотя они и не конфликтуют по ресурсам (выполняются разными исполнителями). Разработка программного обеспечения возможна после разработки требований, а установка системы возможна только завершения разработки.

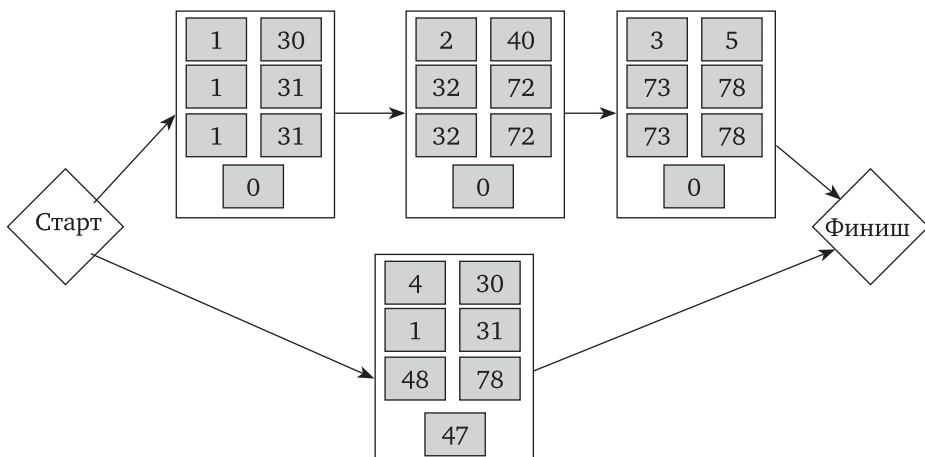


Рис. 7.3. Сетевая диаграмма с работами в узлах для проекта разработки и внедрения автоматизированной системы «Домашний банк»

Задача 4 не зависит от задач 1, 2, 3 ни по результатам, ни по исполнителям (будем считать, что предмет договора с клиентом не зависит от требований к системе и наоборот).

Ранний старт задачи 1 и задачи 4 совпадает и равен первому дню проекта ($ES_1 = ES_4 = 1$).

Ранний финиш задачи 1 равен раннему старту плюс время исполнения задачи 1 ($EF_1 = 1 + 30 = 31$).

Ранний старт задачи 2 равен дню, следующему, за ранним финишем задачи 1 ($ES_2 = EF_1 + 1 = 32$). Ранний финиш задачи 2 равен ее раннему старту плюс время исполнения ($EF_2 = 32 + 40 = 72$).

Ранний старт задачи 3 равен дню, следующему, за ранним финишем задачи 2 ($ES_3 = EF_2 + 1 = 73$). Ранний финиш задачи 3 равен ее раннему старту плюс время исполнения ($EF_3 = 73 + 5 = 78$).

Критическим является путь через работы 1, 2, 3 с общей длительностью 78 дней. Следовательно, планируемый срок проекта равен 78 дням.

Для определения позднего старта и финиша задач, а также резервов времени, необходимо выполнить так называемый «обратный проход» диаграммы.

Если за плановый срок проекта принимаем 78 дней (срок может быть задан внешними условиями), то поздним финишем задач 3 и 4 будет значение 78.

Поздний старт задачи вычисляется как разность между значением позднего финиша и временем исполнения.

Для задач критического пути поздний старт и ранний старт равны, поздний финиш и ранний финиш, также, равны. Резерв задач критического пути равен 0.

Поздний старт задачи 4 равен разности между значением ее позднего финиша и времени исполнения ($LS_4 = LF_4 - 30 = 78 - 30 = 48$).

Резерв времени задачи 4 равен разности между ее поздним стартом и ее ранним стартом ($FL_4 = LS_4 - ES_4 = 48 - 1 = 47$).

Здесь мы несколько упростили задачу расчета, предварительно определив критический путь. В реальных сложных проектах часто приходится полностью производить расчет значений позднего старта (финиша) и резерва времени, именно для определения критического пути.

Точно оценить время исполнения отдельных задач в реальных проектах бывает достаточно тяжело. Часто эта задача становится нереализуемой. Для учета вероятностного характера оценки длительности и ресурсоемкости работ рекомендуется производить планирование в два прохода (прямой и обратный) с определением для каждой работы резерва времени исполнения, а также вероятностной оценки длительности (англ. *project evaluation and review technique, PERT*).

Описанные диаграммы имеют значительный недостаток с точки зрения полноты информации для визуального анализа. В них присутствует информация о взаимосвязи работ, но отсутствует временная шкала, информация о времени реализации не проявляется визуально (только цифры). Этот недостаток устраняется в так называемой диаграмме Ганта. Диаграмма названа в честь своего автора Генри Ганта, который предложил данную форму представления в начале XX в. На рис. 7.4 представлена диаграмма Ганта для проекта разработки и внедрения системы «Домашний банк» (табл. 7.2). Для сокращения размера диаграммы примем сокращенное время исполнения задач (несмотря на то, что это нереалистично).

Таблица 7.2

План работ проекта «Домашний банк»

Задача	Время исполнения (дни)	Исполнитель
Разработка требований	3	Технолог
Разработка программного обеспечения	10	Разработчик
Установка системы	1	Системный администратор
Разработка и согласование договорной базы для юридического оформления подключения клиентов к системе	3	Юрист

На рис. 7.4 непрерывными горизонтальными полосами, вытянутыми по оси времени, обозначены выполняемые работы. Сверху буквами П, В, С, Ч, П, соответственно, обозначены рабочие дни недели.

Первичный вариант диаграммы Ганта не содержал информации о связях задач. В конце XX в. типовые диаграммы обогатили информацией о связи задач между собой. На рис. 7.5 представлены примеры таких связей. Связь задач в диаграмме позволяет автоматизированно управлять изменением сроков задач во всем плане проекта при коррек-

тировке отдельных (например, в *MS Project*). Если эти связи не установлены, то при изменении длительности одной задачи вы будете вынуждены корректировать и уточнять сроки всех задач в плане.

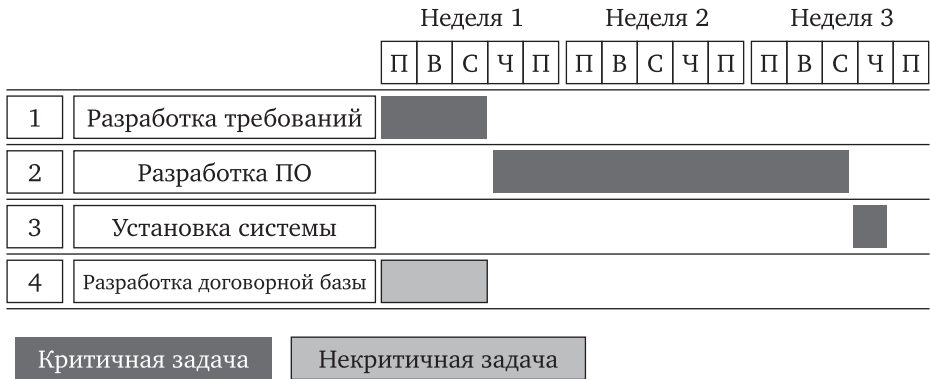


Рис. 7.4. Диаграмма Ганта для проекта разработки и внедрения автоматизированной системы «Домашний банк»

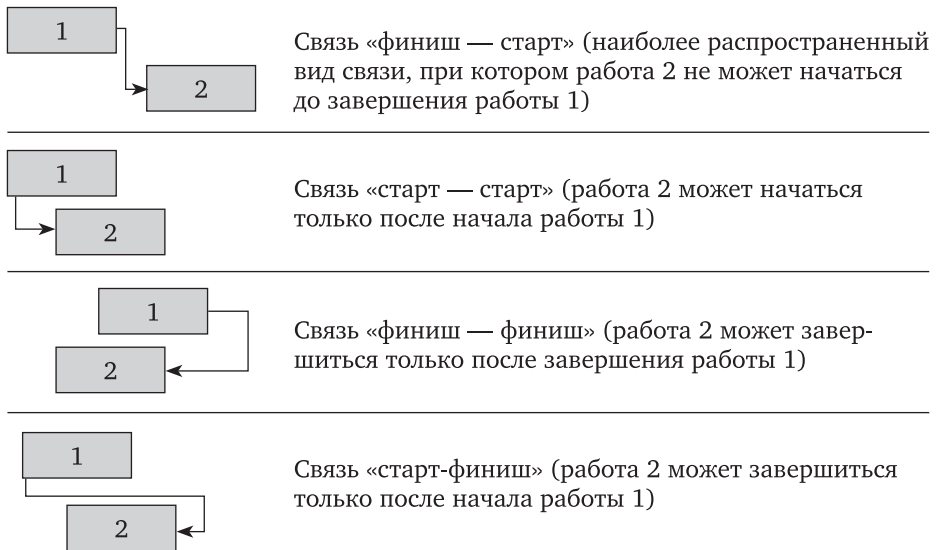


Рис. 7.5. Примеры связи задач

Примером связи «финиш — старт» может служить связь между задачами «разработка» и «тестирование». Тестирование не может начаться, пока не завершён этап разработки при традиционном подходе.

Примером связи «старт — старт» может служить связь между задачами «тестирование» и «исправление замечаний». Исправление замечаний может начаться только после того, как стартовала работа по тестированию и начали появляться первые замечания к системе.

Пример связи «финиш — финиш» — связь между задачами «проектирование» и «планирование». Планирование не может завершиться

до завершения проектирования системы. Состав работ в проекте, безусловно, зависит от состава будущего продукта, и планы будут уточняться по мере уточнения проектных документов.

Связь «старт — финиш» может применяться, например, в случае если одна работа ограничивает другую. Так, например, задача «принятие решения о завершении тестирования и приемке системы» ограничивает исполнение задачи «тестирование и исправление замечаний».

7.3. Пример контрольного (базового) плана ИТ-проекта

На практике формат контрольного (базового) плана ИТ-проекта может значительно изменяться в зависимости от требований организации, опыта руководителя проекта, приоритетов заинтересованных сторон проекта.

Рассмотрим особенности его основных блоков на основе примера, приведенного в табл. 7.3. Автор неоднократно использовал предложенный шаблон для составления контрольных (базовых) планов.

После таблицы приведены дополнительные комментарии о смысле задач плана и их взаимосвязи.

Таблица 7.3

Шаблон базового плана проекта

Задача	Дата исполнения	Исполнитель	Результат
1. Согласование и утверждение устава проекта	$T + 15$	Руководитель проекта	Устав проекта, включая контрольный план и состав команды проекта
2. Обследование участка автоматизации	$T + 30$	Команда проекта	Отчет об обследовании
3. Разработка и согласование требований	$T + 60$	Команда проекта	Требования
4. Разработка эскизного проекта	$T + 70$	Команда проекта	Эскизный проект
5. Презентация эскизного проекта	$T + 70$	Команда проекта	Протокол замечаний
6. Разработка и согласование частных требований	$T + 70$	Команда проекта	Частные требования
7. Рассмотрение предложений сторонних производителей	$T + 85$	Команда проекта либо специализированное подразделение	Сравнительный анализ, утвержденное решение о закупке

Задача	Дата исполнения	Исполнитель	Результат
8. Заключение договора о поставке	T + 90	Специализированное подразделение	Договор
9. Поставка и разработка подсистем	T + 150	Команда проекта либо внешний исполнитель	Установочный комплект
10. Разработка и согласование рабочего проекта	T + 150	Команда проекта	Рабочий проект
11. Разработка программы и методики испытаний	T + 150	Команда проекта	Программа и методика испытаний
12. Создание испытательного стенда и предварительные испытания	T + 200	Команда проекта, либо эксплуатирующее подразделение, либо служба защиты информации	Протокол испытаний
13. Разработка эксплуатационной документации	T + 230	Команда проекта либо внешний исполнитель	Инструкция пользователя, администратора
14. Исправление замечаний	T + 230	Команда проекта либо внешний исполнитель	Установочный комплект
15. Обучение пользователей	T + 250	Команда проекта либо внешний исполнитель	Журнал обучения
16. Портация данных и опытная эксплуатация	T + 260	Команда проекта, либо эксплуатирующее подразделение, либо служба защиты информации (пользователи)	Журнал опытной эксплуатации
17. Исправление замечаний	T + 260	Команда проекта либо внешний исполнитель	Установочный комплект
18. Установка системы и приемочные испытания	T + 270	Команда проекта, либо эксплуатирующее подразделение, либо служба защиты информации (пользователи)	Протокол испытаний, акт приемки-сдачи промышленной установки

Сначала дадим несколько общих комментариев.

В табл. 7.3 по отдельным задачам представлены сроки окончания работ и представления их результатов. Это контрольный план, а не рабочий координационный. На этапе составления контрольного плана высокой точности планирования добиться невозможно. У руководителя проекта должна сохраняться возможность «маневра» сроками старта отдельных работ на уровне рабочего планирования. Это его право.

При подготовке шаблона после уточнения списка задач и их результатов использовался комбинированный подход к определению сроков: по начальным задачам сроки определялись, исходя из оценки реалистичных достигаемых на практике сроков, затем планирование сроков было продолжено от целевой даты завершения проекта в обратном порядке. Ориентировочно на 11 пункте произошло совмещение и уточнение «плана старта» и «плана завершения». Этот подход может оказаться полезным для практического применения.

Задачи 1—7 могут быть не включены в правило ограничения 6 месяцев (табл. 7.3). Выполняя эти пункты, организация пока не несет значительных рисков в связи с возможным прекращением проекта.

Выполнение каждой задачи плана сопровождается выпуском и согласованием соответствующих документов, которые поддаются внешнему контролю со стороны заинтересованных сторон проекта и согласованы квалифицированными участниками проекта, не входящими непосредственно в команду проекта. Таким образом, мы создаем средство объективного контроля исполнения проекта. Включение пунктов, исполнение которых не поддается внешнему независимому контролю, не является эффективным и обоснованным.

Таблица 7.3 предполагает реализацию 18 задач в течение 270 дней. Таким образом, контрольные точки с предоставлением распознаваемых и согласуемых документов предусмотрены в среднем через каждые 14 дней. Это показывает, что, несмотря на значительную длительность (270 календарных дней), проект является очень интенсивным (ранее мы говорили о том, что контрольный (базовый) план предпочтительно должен включать контрольные точки с периодичностью 1 месяц).

План содержит единственный значимый резерв по времени, начиная с 8 пункта. Это означает, что потенциально проект может быть сжат по общему сроку примерно на 30 дней. Это может реализоваться, если и поставщик, и внутренний исполнитель смогут согласовать с командой проекта и обеспечить поставку в режиме *MVP* (сокращенной функциональности).

В случае разбиения проекта на функциональные этапы с поставкой частичной функциональности в каждом этапе будут повторяться пункты с 6 по 19. В рамках разработки эскизного проекта мы примем решения об итерационной поставке функционала, разобьем его на этапы в первом приближении и частные требования будем задавать на отдельные *MVP* этапов с доведением их до промышленной эксплуатации. При этом основные проектные решения по системе, ее систем-

ному аппаратному и программному обеспечению, архитектуре будут приняты при согласовании эскизного проекта.

Подробнее рассмотрим отдельные ключевые пункты плана.

Ранее мы подробно рассматривали содержание и цели формирования устава проекта. Здесь отметим, что включение задачи по его согласованию и утверждению в типовой контрольный план не является обязательным, но является полезным. На практике часто именно этот этап чрезмерно затягивается, а руководство склонно рассматривать в качестве даты старта проекта дату принятия принципиального решения о его инициации, а не дату утверждения устава проекта, наделяющего руководителя проекта необходимыми полномочиями. Бывает полезно сохранить информацию о реальных сроках исполнения «второстепенных» организационных задач.

Задача 2 — «Обследования участка автоматизации». Ранее, в гл. 2, рассматривались основные подходы к решению сложных инженерных задач. Было указано на то, что одним из основных подходов является последовательность «анализ — синтез». Часто на практике упускают необходимость выполнения первой фазы «анализ» и переходят сразу к фазе «синтез», тем самым значительно увеличивая риск ошибок при проектировании.

Полезность данного этапа не ограничивается непосредственно анализом, при котором, к слову, может быть выполнено уточнение списка заинтересованных стороны и других параметров устава проекта, дополнительный выигрыш получаем за счет того, что сборная команда проекта, ранее не имевшая опыта совместной работы, проходит первичное «боевое слаживание», получает недостающие знания в предметной области и знакомится с будущими пользователями продукта проекта, открывая для себя потенциал будущей эффективной коммуникации.

Этап «обследование» обычно занимает до двух недель на подготовку отчета и дополнительно одну неделю на его формальное согласование с заинтересованными сторонами. Отчет об обследовании не должен быть объемным. Перед ним не ставится задача содержать детальное описание деятельности автоматизируемых подразделений. Его основная задача — проверить допущения, сделанные при инициации проекта, продемонстрировать необходимый уровень «погружения» команды проекта в предметную область и дать представление об уровне сложности проекта для всех заинтересованных сторон. Он должен быть информативным и лаконичным, демонстрировать профессионализм ответственного за него руководителя проекта и быть доступным для прочтения всеми заинтересованными сторонами (это означает, что он не должен быть пересыщен техническими деталями).

На практике отчет обычно составляет 30—40 страниц и содержит следующие основные разделы: описание архитектуры, состав подразделений-пользователей и ответственные от них, количественные данные об обрабатываемых операциях, типы статической и динамической информации, состав администраторов участка автоматизации, количество форм ввода-вывода информации, список печатаемых отчетов, роли пользовате-

лей участка автоматизации. Легко заметить, что вся информация может в дальнейшем использоваться для разработки требований к системе. Более того, без этой информации требования не могут быть разработаны.

Раздел, содержащий описание существующей архитектуры модернизируемого участка автоматизации с указанием взаимодействующих систем, интерфейсов и режимов взаимодействия, обычно не занимает больше 2—3 страниц и состоит из графической схемы и последующей таблицы описания систем и их взаимодействия. Схема практически всегда доступна для размещения в «читабельном» виде на формате листа А4.

Интересна история, связанная с подобной схемой. До того, как автор принял участие в проекте, организация занималась его реализацией на протяжении четырех лет. Результаты проекта в использование не поступали. Разработка ядра автоматизированного комплекса велась итерационно, т. е. требования направлялись внешнему поставщику участками, после чего получаемый участок системы подвергался тестированию. Несмотря на высокую интенсивность работ (в проекте участвовало с полной занятостью до 60 человек с обеих сторон), система не «срасталась» в единый функционирующий комплекс. Главной причиной было отсутствие закрепления границ комплекса на начальной фазе проекта; у каждого технолога — разработчика требований было свое представление о том, какой функционал крупноблочно включен в проект. И это представление дополнительно изменялось с течением времени.

В результате быстрого рестарта проекта была оперативно набрана собственная группа разработчиков, и еще до ее полного укомплектования в течение трех недель проведено обследование с маркировкой на схеме модернизируемых участков. Непосредственно после демонстрации отчета руководству организации оно приняло решение о премировании новой команды проекта с формулировкой: «Мы, наконец, разобрались, что модернизируем».

Под номером 3 в шаблоне плана приведена задача «Разработка и согласование требований». Если мы имеем дело с мелким проектом, требования к которому принципиально могут быть разработаны за указанный срок, то они могут иметь достаточно детальный характер. Если проект масштабный, то требования должны быть высокоуровневыми и детализироваться в дальнейшем на этапе 6 (разработка частных требований). Повторно отметим, что требования должны согласовываться с заинтересованными сторонами проекта, включая службу защиты информации и эксплуатирующие подразделения.

Следующие две задачи (4, 5) — «разработка и согласование эскизного проекта» и «презентация эскизного проекта» — тесно взаимосвязаны и при этом разнесены в отдельные задачи для выделения значимости активной коммуникации команды проекта с заинтересованными сторонами проекта, желательно в виде специально подготовленной совместной презентации эскизного проекта. По своей сути эскизный проект в рамках Единой системы конструкторской документации

(стандарт ЕСКД) имеет целью выделение этапа, на котором принимаются верхнеуровневые принципиальные проектные решения и задается декомпозиция систем на крупные блоки.

Руководитель проекта, получив на вход утвержденные требования, должен выработать и согласовать ряд важных решений, которые определяют дальнейший ход проекта и позволят выработать проектные документы более мелкого уровня детализации (частные требования, технический и рабочий проекты). Примером такого решения может являться, например, закупка отдельных компонент создаваемого автоматизированного комплекса или отказ от наращивания производительности системы в рамках проекта с целью приоритетного решения функциональных задач.

Если подобные «тектонические» решения будут согласовываться в обычной форме официального согласования, то, скорее всего, задача не будет решена в срок, если вообще будет решена. Поэтому более эффективным является очная презентация перед заинтересованными сторонами с получением на выходе утвержденного эскизного проекта или замечаний к нему. Очное обсуждение будет облегчено тем, что в дискуссии на стороне руководителя проекта примет участие спонсор либо куратор проекта. Кроме того, данная презентация позволяет практически полностью обезопасить проект от замечаний по отсутствию необходимого информирования заинтересованных сторон и согласования с ними.

Не будем детализировать остальные задачи, включенные в план. Они не составляют секрета для технических специалистов и в то же время могут содержать излишние технические детали для читателя.

7.4. Управление рисками проекта

Ранее упоминалось, что планирование проектов, также как их реализация в целом, связаны с вероятностными оценками. С большей или меньшей вероятностью отдельные задачи проекта могут быть не выполнены. Часто говорят, что в проектах создания или модернизации автоматизированных систем присутствует вероятность изменения требований. Во всех этих случаях с некоторой вероятностью может произойти событие (или его отсутствие), которое негативно скажется на реализации проекта, качестве результата, сроке реализации или дополнительных затратах. И у вероятности, и у негативного результата есть величина. Если вероятность оценивается как стопроцентная, то риск теряет свою вероятностную составляющую и риском уже не является.

Обычно риск рассчитывают как произведение вероятности на величину ущерба; такой расчет называют двухфакторным. Расчет выполняется для различных источников возникновения риска и для различных видов ущерба. Например, риск увеличения срока реализации проекта в результате ошибочной оценки трудозатрат по отдельным задачам.

Поскольку ущерб не всегда имеет финансовое выражение, например, снижение качества результата проекта (количество ошибок в разработанном программном обеспечении), то все виды ущерба оценивают в балльной шкале с суммированием всех видов возможного ущерба.

Для чего производится расчет рисков? Не рассчитав значения по различным источникам риска и различным видам ущерба, мы не сможем выявить максимальные значения риска для того, чтобы эффективно им противодействовать. Если предпринимать шаги по снижению всех возможных в проекте рисков, то затраты на проект окажутся слишком большими. Общепринятой практикой является противодействие рискам в соответствии с диаграммой Парето (см. гл. 4). В этом случае выделяются крупнейшие риски, суммарная величина которых составляет 80 % от всех выявленных рисков, и предпринимаются меры, направленные на их снижение. Как уже говорилось выше, в этом случае удастся значительно сфокусировать свои усилия и добиться значительного снижения затрат.

Существует четыре основных способа управления рисками: снижение риска, уклонение от риска, изменение характера риска и принятие риска.

Снижение риска достигается, если удастся либо снизить вероятность реализации рискового события, либо смягчить его последствия путем снижения возможного ущерба. Это самый широко применяемый прием управления риском.

Например, вероятность ошибочного составления требований к системе снижается за счет проверки списка задокументированных ролей (актеров в терминах *UML*) по фактическому списку будущих пользователей системы. А негативное воздействие этой ошибки может быть снижено за счет ранней демонстрации прототипа системы будущим пользователям (что является одним из основных приемов в методиках *agile*). Чем быстрее устранена ошибка, тем дешевле обходится ее устранение.

Уклонение от риска — прием, который не всегда осуществим на практике. Уклонение — это снижение вероятности реализации рискового события до нулевого значения. Примером реализации такого подхода может служить отказ от заключения договора с внешней компанией для уклонения от риска нарушения внешним поставщиком своих обязательств по договору. Понятно, что в данном случае мы лишаемся и возможности привлечения внешних ресурсов.

Изменение характера риска достаточно применяется достаточно широко. Так, например, застраховав свой возможный ущерб, который может проявиться в результате проявления ошибок в системе, мы заменяем риск неэффективного тестирования кредитным риском — страховщик может не выплатить нам компенсацию.

Некоторый уровень риска имеет место в работающей системе или реализуемом проекте всегда (если мы не уклонились от него, закрыв проект). Невозможно сохранить работоспособность и снизить риск до нулевого уровня. Для некоторого уровня риска его дальнейшее снижение не является оправданным — мы начинаем затрачивать больше,

чем защищаем. Такой уровень риска должен быть принят. Принимаемый уровень риска обычно непосредственно связан с иерархией организации. Самый большой уровень риска принимает первое лицо (руководитель) организации. На нижние уровни иерархии делегируется право принимать более мелкие уровни риска. Руководитель проекта, также имеет право принимать некоторый уровень риска, но не риск в объеме проекта (неудачного завершения проекта). Этим правом обычно располагает куратор (спонсор) проекта. Для принятия подобного риска он должен располагать максимально точно рассчитанным значением риска. Этот расчет должен обеспечить руководитель проекта как профессионал в области управления проектами.

Расчет риска базируется на корректной оценке вероятности реализации негативного рискового события, выявления максимально полного списка возможных рисковых событий и оценке возможного ущерба.

В широко распространенных методиках по управлению проектами, в том числе в *PMI PMBOK*, часто рекомендуется выявлять индивидуальный набор рисков для отдельных проектов. Однако индивидуальный набор рисков характерен скорее для организации в целом, находящейся на определенном уровне зрелости, чем для отдельного проекта. Если организация находится на нижнем уровне зрелости (первый), который характеризуется отсутствием регламентации и устойчивых процессов, то каждый крупный проект подвержен почти стопроцентному риску неуспеха, требования к системе в любом случае будут составлены с большими ошибками, а тестирование не выявит значительного количества ошибок в системе (в виду отсутствия тестирования как процесса). И это не особенность отдельного проекта, а особенность организации. Соответственно, меры по управлению рисками в этом случае для всех реализуемых проектов должны быть типовыми. По крайней мере, не стоит тратить ресурсы проекта на то, чтобы выявить список индивидуальных проектных рисков. Их базу можно поддерживать и уточнять в виде единого методического документа, рекомендующего набор мер для каждого руководителя проекта. Существуют и централизованно поддерживаемые базы данных по рискам, в том числе свойственным для ИТ-проектов. Например, подобная база данных длительное время развивается *Software Engineering Institute*.

Вероятность, входящая в расчет риска, обычно определяется на основе накопленной статистики. Однако вероятность можно определять и экспертно. Здесь, безусловно, играет роль уровень накопленного опыта руководителя проекта. Реализовывая проекты различного уровня сложности в различных организациях, руководитель проекта получает необходимые экспертные навыки для точного выделения значимых рисков и выработки стратегии для их снижения.

Подробнее остановимся на нестандартных типах рисков, которые часто упускаются из виду широко применяемыми методиками управления проектами, однако имеют большое значение на практике.

Российская история характеризуется долгими периодами затишья и периодически проявляющимися вспышками активности. Данный вид

риска выходит на первое место именно в «активные» периоды, причем становится общим и для коммерческих, и для государственных организаций. Здесь мы говорим о досрочном прекращении проекта в результате смены руководства ключевых подразделений, являющихся заинтересованными сторонами в проекте, или руководства организации в целом. Практически всегда смена руководства (спонсора, заказчика, руководителя ИТ — *CIO*) приводит к тому, что пересматриваются все долговременные проекты, не успевшие представить значимый результат. Новое руководство склонно корректировать цели и затратные инициативы своих предшественников. Учитывая, что ИТ-проекты обычно являются долговременными и затратными, можно говорить о том, что вероятность наступления данного рискованного события является высокой, а последствия для проекта должны получить максимально возможную оценку. Уклониться от такого риска невозможно, но можно смягчить ущерб или снизить вероятность принятия трагичного для проекта управленческого решения.

Основным приемом является максимально скорая поставка системы в промышленную эксплуатацию с частичной функциональностью. Полезно используемые системы обычно не «выбрасывают», и проекты получают шанс на дальнейшее развитие. Следует обратить внимание на то, что данный подход полностью соответствует методикам *agile* в части поставки минимально жизнеспособного продукта (*MVP*). Рекомендуемый срок поставки системы в эксплуатацию на практике составляет до шести месяцев. Далее риск приобретает недопустимо высокое значение.

Следующим по значимости риском, характерным для российской практики, является отказ эксплуатирующих подразделений, службы защиты информации или пользователей от приемки реализованной в рамках проекта системы. Этот риск особенно характерен для организаций с низким уровнем зрелости. Во-первых, подобные организации в целом склонны к проявлениям необоснованного конфликтного поведения, а во-вторых, как уже говорилось ранее, проекты инициируют процессы изменений, которым сопротивляется общая среда организации.

Процедуры приемки-сдачи не только служат для передачи ответственности за продукт проекта от команды проекта эксплуатирующим подразделениям, но и служат точкой отсчета периода, когда эти подразделения и команда проекта уже не имеют возможности отказаться от «раздражающего взаимодействия». И именно в этот период проявляется весь накопленный нереализованный объем необходимой адаптации.

Основным приемом, служащим для снижения данного риска, является регулярная коммуникация между заинтересованными сторонами и поэтапное, с самых первых шагов проекта, согласование проектных документов с принимающими результат подразделениями. Опытный руководитель проекта постарается начать «процесс переговоров» об условиях сдачи-приемки задолго до его начала.

Глава 8

СТАНДАРТ PMI PMBOK. ИСТОРИЯ СОЗДАНИЯ СТАНДАРТА. СТАНДАРТНЫЕ ПРОЦЕССЫ

В настоящее время существует целый ряд широко применяемых в мире стандартов в области проектного управления. Одним из наиболее универсальных и распространенных является *Project Management Institute Project Management Body of Knowledge (PMI PMBOK)*.

Строго говоря, набор знаний *PMBOK Guide*, несмотря на часто включаемую приставку *ANSI* (от англ. *American National Standards Institute*), не является стандартом в полном объеме. Официальным стандартом признается *Project Management Standard for Managing a Project* объемом около 40 страниц (в целом *PMBOK Guide* составляет более 600 страниц). И, безусловно, приставка *ANSI* не дает права считаться международным стандартом.

PMBOK — это в большей степени набор шаблонов, терминологии и описаний. В нем чаще можно найти ответ на вопрос «что?», нежели «как?» Его применение без адаптации и разработки детальных внутренних методик не является достаточным для налаживания проектной работы.

Сертификация специалиста по *PMBOK* обеспечивает для него возможность общаться с коллегами в едином поле понятий на едином языке, анализировать и декомпозировать проектную деятельность на единый набор элементов и процессов с единым набором ролей.

Содержание *PMBOK* структурировано по пяти основным направлениям:

- 1) жизненный цикл проекта;
- 2) группы процессов;
- 3) области знания;
- 4) методы;
- 5) инструменты.

Каждый процесс *PMI PMBOK* принадлежит одной из групп процессов, при этом входит в одну из областей знаний, но может реализовываться в разных фазах жизненного цикла проекта и применяет различный набор методов и инструментов.

8.1. История стандарта *PMBOK*

Исторически управление проектами как специальная область деятельности начало формироваться в рамках космических и оборонных программ ведущих стран мира после Второй мировой войны. Начиная

с 1960-х гг. это направление начало обретать черты, которые закрепились за ним и по настоящее время.

В рамках данной международной активности в США в 1969 г. был создан Институт по управлению проектами — *Project Management Institute* при Технологическом институте Джорджии (англ. *Georgia Institute of Technology*).

Сформулированные в 1975 г. основные цели деятельности института звучали следующим образом:

- развивать понимание и потребность профессиональной проектной деятельности;
- обеспечивать форум для свободного обмена проблемами, решениями и инструментами;
- координировать отраслевые и академические исследования в области проектного управления;
- развивать единую терминологию и шаблоны для совершенствования взаимодействия и обмена знаниями;
- обеспечивать интерфейс между пользователями и производителями программного обеспечения;
- обеспечивать руководство профессиональным развитием специалистов в области проектного управления.

Институт *PMI*, несмотря на отсутствие прямой связи с государством США, был с ним достаточно жестко ассоциирован в связи с тем, что его деятельность в основном проводилась в интересах государства. Большинство участников организованного Института являлись сотрудниками различных государственных организаций.

К 2017 г. Институтом сертифицировано более 850 тыс. специалистов по всему миру. Членами сообщества института являются 489 тыс. профессионалов проектного управления. Институт насчитывает 289 отделений в различных странах.

В конце 1980-х гг. деятельность *PMI* в области стандартизации начала оформляться в виде создания руководства по управлению проектами. Первая версия *PMBOK* была опубликована в 1996 г. В сентябре 2017 г. была выпущена шестая версия *PMBOK Guide*. В целом было издано более пяти с половиной миллионов экземпляров *PMBOK Guide*.

В ходе эволюционной доработки *PMBOK Guide* на протяжении многих лет его состав значительно изменялся. Изменялся набор подходов, описываемых областей знаний и процессов. Например, вышедшая в 2008 г. четвертая версия дополнилась новыми подходами: аналитическая работа, прототипирование, итеративность и применение систем искусственного интеллекта для прогнозирования проекта в части сроков и бюджета. Вышедшая в 2017 г. шестая версия дополнена отдельной книгой с описанием методологии *agile* (англ. *Agile Practice Guide*). Шестое издание было опубликовано в том числе на русском языке.

Выпущенный в 2017 г. *Agile Practice Guide* уделяет большое внимание исключению противопоставления традиционных и *agile* методов управ-

ления проектами и содержит описание всех основных применяемых на практике методов гибкого проектного управления.

Кроме *PMBOK* Институтом был выпущен ряд так называемых *Foundational Standards* — дополнительных стандартов по управлению проектной деятельностью в более широком смысле:

- *The Standard for Portfolio Management* — Стандарт по управлению портфелем проектов;
- *The Standard for Program Management* — Стандарт по управлению программами;
- *Organizational Project Management Maturity Model (OPM3)* — Модель зрелости проектного управления в организации.

8.2. Жизненный цикл проекта *PMBOK*

PMBOK базируется на процессном подходе. Это означает, что деятельность по управлению проектами согласно стандарту обеспечивается набором типовых процессов, т. е. набором стандартных операций, выполняемых типовыми ролями, имеющими условия и артефакты старта (англ. *inputs*), инструменты и результаты (англ. *outputs*). Процессный подход объединяет *PMBOK* с другими стандартами и моделями управления, такими как *ISO 9000* и *СММІ*. Таким образом, *PMBOK* в целом соответствует основным требованиям общей теории качества (*TQM*).

Взаимодействие процессов в рамках стандарта систематизировано в соответствии с предложенным жизненным циклом проекта (англ. *project management life cycle*).

Жизненный цикл проекта — это последовательность распознаваемых фаз проекта, которые иногда могут перекрываться и должны быть регламентированы в организации в соответствии с особенностями ее деятельности и ее уровнем зрелости. Обычный вид жизненного цикла проекта представлен на рис. 8.1.

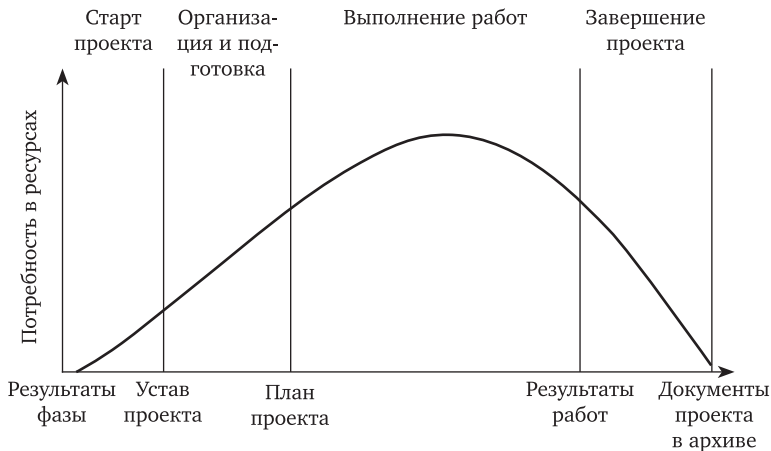


Рис. 8.1. Жизненный цикл проекта

На различных фазах проекта обеспечивается взаимодействие в различной конфигурации 49 типовых процессов, объединенных в пять больших групп.

Наименования групп процессов могут ввести в заблуждение, поскольку сходны с наименованиями фаз жизненного цикла проекта. Важно понимать, что процессы и группы могут исполняться параллельно и, например, группа процессов завершения (англ. *closing*) начинает исполняться непосредственно после старта проекта, не дожидаясь фазы «завершение проекта». Данная идея проиллюстрирована на рис. 8.2.

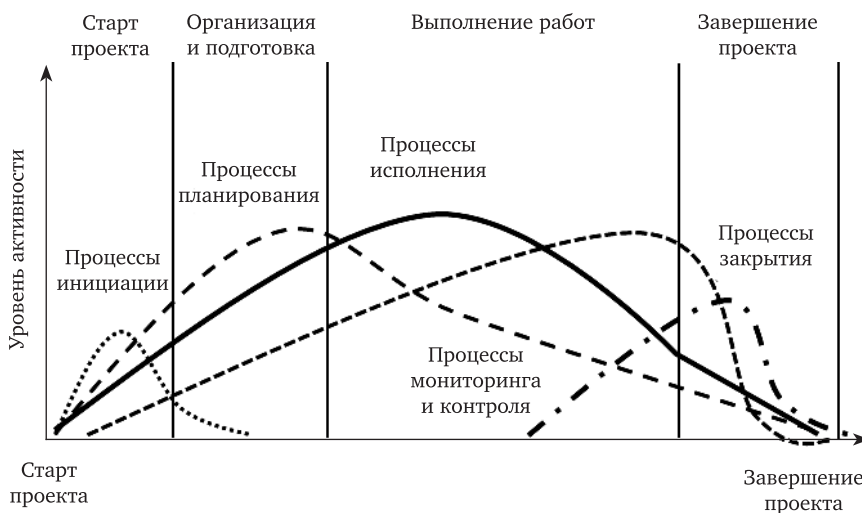


Рис. 8.2. Жизненный цикл проекта и группы процессов

8.3. Области знания PMI PMBOK

Приведенные в PMI PMBOK группы процессов используют в своей реализации отдельные методики и знания, более подробно описываемые в разделе «областей знания». Шестая версия PMI PMBOK включает десять областей знания, не считая *Agile Practice Guide*.

Управление интеграцией проекта (англ. *Project Integration Management*) включает деятельность по идентификации, определению, унификации, объединению, консолидации и координации работ в рамках управления проектом для его успешного завершения и удовлетворения ожиданий заинтересованных сторон. В этой области знаний в том числе описывается подход к разрешению конфликтов в проекте и приводится схема процессов разработки устава проекта и плана управления проектом, руководство и управление работами проекта, управление знаниями проекта, мониторинг и контроль работ проекта, интегрированный контроль изменений, закрытие проекта или фазы.

Управление содержанием проекта (англ. *Project Scope Management*) описывает каким образом определить содержание работ, необходимых

для успешного завершения проекта, каким образом свести данный набор работ к необходимому минимуму. В этой области знаний описываются схемы процессов: планирование управления содержанием, сбора требований, определения содержания, создания иерархической структуры работ (англ. *work breakdown structure, WBS*), подтверждения содержания, контроля содержания.

Управление расписанием проекта (англ. *Project Schedule Management*) описывает деятельность, посредством которой обеспечивается своевременное завершение проекта и его отдельных работ. Включает планирование расписания, определение операций и последовательности операций, оценку длительности операций, разработку расписания, контроль расписания.

Управление стоимостью проекта (англ. *Project Cost Management*) включает описание процессов, служащих для планирования и уточнения бюджета, финансирования работ, управления и контроля расходов для того, чтобы проект был реализован в рамках утвержденного бюджета (планирование управления стоимостью, оценку стоимости, определения бюджета и контроль стоимости).

Управление качеством проекта (англ. *Project Quality Management*) описывает деятельность по соблюдению политики обеспечения качества, критериев качества и целевых показателей качества. В данной области знаний в том числе приведено описание цикла постоянного улучшения процессов, который должен быть задействован в ходе всего проекта, а также процессы планирования управления качеством, управления качеством и контроля качества.

Управление ресурсами проекта (англ. *Project Resource Management*). Здесь приводится описание процессов, направленных на управление, приобретение и использование в нужное время в нужном месте ресурсов, доступных руководителю проекта, включая команду проекта. Схема процессов управления человеческими ресурсами включает в себя планирование управления ресурсами, оценку ресурсов операции, приобретение ресурсов, развитие команды, управление командой, контроль ресурсов.

Управление коммуникациями проекта (англ. *Project Communications Management*). Процессы управления коммуникациями применяют с целью обеспечения своевременного формирования, подготовки, распространения, архивации, передачи, получения, использования информации при реализации проекта. Эффективность коммуникации определяется тем, насколько хорошо обеспечивается взаимодействие между заинтересованными сторонами. Схема процессов управления коммуникациями проекта включает в себя планирование коммуникаций, управление коммуникациями, мониторинг коммуникаций.

Управление рисками проекта (англ. *Project Risk Management*). Под процессами управления рисками проекта понимается планирование управления рисками, идентификация и анализ рисков, выработка методов реагирования на риски, контроль, мониторинг и управление

рисками в ходе реализации проекта. С помощью этих процессов руководитель проекта может повышать вероятность возникновения благоприятных случайных событий и снижать вероятность возникновения и воздействия неблагоприятных случайных событий. Схема процессов управления рисками проекта включает в себя планирование управления рисками, идентификацию рисков, качественный анализ рисков, количественный анализ рисков, планирование реагирования на риски, осуществление реагирования на риски, мониторинг рисков.

Управление закупками проекта (англ. *Project Procurement Management*). Процессы управления закупками проекта включают в себя приобретение необходимых для проекта объектов (продукты, услуги, результаты, документы) у внешних поставщиков. Схема процессов управления поставками проекта включает в себя планирование управления закупками, проведение закупок, контроль закупок.

Управление заинтересованными сторонами проекта (англ. *Project Stakeholder Management*). Под процессами управления заинтересованными сторонами проекта понимается обеспечение эффективного взаимодействия с заинтересованными сторонами команды проекта, а также меры по удовлетворению потребностей заинтересованных сторон. В состав схемы управления заинтересованными сторонами входят такие процессы, как идентификация заинтересованных сторон, планирование вовлечения заинтересованных сторон, управление вовлечением заинтересованных сторон, мониторинг вовлечения заинтересованных сторон.

8.4. Группы процессов проекта

Области знания содержат общее описание процессов, которые в рамках *PMI PMBOK* дополнительно объединены в пять групп в соответствии с жизненным циклом проекта:

- группа процессов инициации (2 процесса);
- группа процессов планирования (24 процесса);
- группа процессов исполнения (10 процессов);
- группа процессов мониторинга и контроля (12 процессов);
- группа процессов закрытия (1 процесс).

Группа процессов инициации (англ. *Initiating*) включает процессы формального согласования условий и утверждения старта проекта или его фазы (табл. 8.1).

Группа процессов планирования (англ. *Planning*) включает процессы определения объема и содержания (англ. *scope*) проекта, уточнения целей, последовательности действий для их достижения. На протяжении проекта осуществляется итерационный возврат к одному или нескольким процессам планирования в связи с уточнением требований и приоритетов заинтересованных сторон (табл. 8.2).

Группа процессов инициации

Наименование процесса	Описание
Разработка устава проекта	Целью является наделение руководителя проекта необходимыми полномочиями по управлению ресурсами и заключение соответствующего соглашения в виде устава. Устав включает: Ф. И. О. руководителя проекта и уровень его полномочий, назначение проекта, исходя из высокоуровневых целей, измеримые цели проекта, высокоуровневое описание требований, границ, результатов, рисков, список контрольных событий, бюджет, список заинтересованных сторон, критерии завершения и требований к контролю результатов
Идентификация заинтересованных сторон	Процесс обеспечивает постоянное выявление заинтересованных сторон, их приоритетов, интересов и меры воздействия на проект с целью использования этой информации в интересах успешного завершения проекта. Заинтересованные стороны определяют требования к проекту, участвуют в его реализации в ходе всей реализации с различной ролью. Ошибка в их определении значительно увеличивает риск проекта

Таблица 8.2

Группа процессов планирования

Наименование процесса	Описание
Разработка плана управления проектом	В рамках процесса формируется и постоянно поддерживается в актуальном состоянии комплексный документ координации отдельных работ проекта
Планирование управления содержанием	Обеспечивает создание и поддержку в актуальном состоянии плана действий: как содержание проекта и продукта будет определяться, подтверждаться и контролироваться
Сбор требований	Процесс сбора, документирования и управления требованиями заинтересованных сторон. На основе требований определяется содержание проекта и продукта
Определение содержания	Разработка подробного описания проекта и продукта. Позволяет описать границы и критерии приемки продукта, услуги или другого результата проекта
Создание иерархической структуры работ	Процесс декомпозиции результата проекта и его работ на меньшие компоненты, поддающиеся более простому управлению
Планирование управления расписанием	Процесс документирования, планирования, разработки и управления расписанием проекта
Определение операций	Процесс определения и документирования конкретных действий, необходимых для получения результата проекта

Наименование процесса	Описание
Определение последовательности операций	Определяет взаимосвязи и логическую последовательность операций в проекте
Оценка длительности операций	Обеспечивает оценку количества рабочих периодов, необходимых для выполнения отдельных операций с учетом доступных ресурсов
Разработка расписания	Позволяет создать расписание с плановыми датами взаимосвязанных операций с учетом доступных ресурсов и их трудоемкости
Планирование управления стоимостью	Обеспечивает управление тем, как стоимость проекта будет оцениваться, включаться в бюджет, исполняться и контролироваться
Оценка стоимости	Процесс предварительной и периодической оценки денежных ресурсов, необходимых для реализации операций проекта
Определение бюджета	Процесс консолидации оценочных стоимостей отдельных операций и пакетов операций проекта, формирования и авторизации базового плана по стоимости, на основе которого будет контролироваться стоимость проекта
Планирование управления качеством	Процесс определения стандартов и требований к качеству результатов проекта. Устанавливает, каким образом проект будет подтверждать соответствие этим требованиям качества, с помощью каких проверок
Планирование управления ресурсами	Определяет, каким образом происходит оценка, приобретение и использование ресурсов проекта. Позволяет планировать и оценивать управленческие трудозатраты для данной активности
Оценка ресурсов операций	Оценка ресурсов команды проекта, привлеченных работников, материалов, оборудования, необходимых для проекта
Планирование управления коммуникациями	Процесс разработки и согласования плана действий по информированию и налаживанию взаимодействия между заинтересованными сторонами
Планирование управления рисками	Определяет, каким образом, с помощью какой последовательности действий осуществляется управление рисками, как будет достигаться пропорциональность действий по управлению рисками их величине и интересам заинтересованных сторон
Идентификация рисков	Процесс определения и документирования значимых для проекта рисков, чтобы команда проекта могла выстроить ответственные меры по управлению риском
Качественный анализ рисков	Процесс ранжирования рисков проекта по вероятности их реализации и силы воздействия на проект для приоритизации дальнейших действий по управлению рисками

Наименование процесса	Описание
Количественный анализ рисков	Позволяет количественно оценить степень возможного воздействия отдельных рисков на цели проекта
Планирование реагирования на риски	Планирование подходов и действий по реагированию на индивидуальные риски и общий риск проекта
Планирование управления закупками	Процесс документирования решений по действиям, необходимым для осуществления закупок в интересах проекта извне организации или внутри
Планирование вовлечения заинтересованных сторон	Поставляет план действий по обеспечению эффективного взаимодействия с заинтересованными сторонами

Группа процессов исполнения (англ. *Executing*) включает процессы, обеспечивающие исполнение работ в соответствии с утвержденным планом проекта, необходимых для достижения целей и требований проекта (табл. 8.3).

Таблица 8.3

Группа процессов исполнения

Наименование процесса	Описание
Руководство и управление работами проекта	Непосредственное руководство работами в проекте в соответствии с его планом для достижения целей проекта
Управление знаниями проекта	Использование знаний организации для реализации проекта, а также накопление и передача знаний, полученных в проекте для последующего использования в организации
Управление качеством	Процесс исполнения плана управления качеством для успешной реализации проекта, выявление неэффективных процессов и других причин снижения качества результатов проекта
Приобретение ресурсов	Получение ресурсов, материалов, членов команды проекта и их распределение по операциям проекта для достижения его целей
Развитие команды проекта	Расширение компетенций членов команды проекта, эффективности их взаимодействия для успешного достижения целей проекта и повышения эффективности проектных процессов
Управление командой проекта	Контроль работы команды проекта с целью повышения ее эффективности, снижения уровня конфликтов и разрешения проблем с целью успешного завершения проекта

Наименование процесса	Описание
Управление коммуникациями	Обеспечение эффективного сбора, документирования, архивирования, передачи информации между заинтересованными сторонами проекта
Осуществление реагирования на риски	Выполнение согласованных планов реагирования на риски с целью минимизации негативного воздействия на проект и максимизации позитивного воздействия
Проведение закупок	Процесс взаимодействия с продавцами, выбора поставщиков, заключение договора
Управление вовлечением заинтересованных сторон	Выполнение действий для эффективного взаимодействия с заинтересованными сторонами для усиления поддержки проекта и минимизации сопротивления ему

Группа процессов мониторинга и контроля (англ. *Monitoring and Controlling*) включает процессы, обеспечивающие отслеживание исполнения проекта, его аудит, регулирование прогресса и производительности работ по проекту, идентификацию областей плана проекта, нуждающихся в корректировке (табл. 8.4).

Таблица 8.4

Группа процессов мониторинга и контроля

Наименование процесса	Описание
Мониторинг и контроль работ проекта	Отслеживание прогресса работ по проекту для своевременного устранения проблем и прогнозирования его состояния с учетом прогнозов стоимости и расписания
Интегрированный контроль изменений	Процесс сбора, регистрации, документирования, взаимной привязки запросов на изменение с целью своевременного реагирования на возникающие в связи с ними риски проекта
Подтверждение содержания	Процесс формализованной приемки результатов проекта
Контроль содержания	Процесс мониторинга и контроля содержания продукта и проекта
Контроль расписания	Процесс мониторинга статуса проекта и поддержки актуальности его расписания
Контроль стоимости	Процесс мониторинга статуса проекта по плану стоимости и актуализация данного плана в случае необходимости
Контроль качества	Процесс мониторинга и документирования выполнения операций проекта по управлению качеством для обеспечения того, чтобы результаты проекта соответствовали ожиданиям заинтересованных сторон и не имели ошибок
Контроль ресурсов	Обеспечение доступности, своевременного использования и высвобождения ресурсов в ходе проекта

Наименование процесса	Описание
Мониторинг коммуникаций	Мониторинг эффективности реализации плана коммуникаций
Мониторинг рисков	Мониторинг выполнения соответствующих планов реагирования на риски, отслеживания рисков, выявления и анализа новых рисков
Контроль закупок	Обеспечение соответствия деятельности продавца и покупателя интересам проекта, организации и юридическим договоренностям и требованиям законодательства
Мониторинг вовлечения заинтересованных сторон	Мониторинг взаимодействия заинтересованных сторон проекта в целях повышения его эффективности

Группа процессов закрытия (англ. *Closing*) включает процессы, служащие для завершения операций по проекту или фазе проекта с официальным оформлением статуса, результата и высвобождением ресурсов проекта.

8.5. Инструменты и методы PMI PMBOK

Кроме детального описания проектных процессов, *PMI PMBOK* содержит описание различных инструментов и техник, которые повышают эффективность деятельности по управлению проектами (всего более 20 методов и 11 инструментов). Эти инструменты и методы не имеют эксклюзивного характера для *PMI PMBOK* и широко применяются в инженерной практике, управлении рисками, менеджменте, управлении качеством и т. п. Часть из них описана выше, часть имеет слишком объемное описание для непосредственного включения:

- *SWOT*-анализ (от англ. *Strengths, Weaknesses, Opportunities, Threats Analysis*) — анализ сильных и слабых сторон, возможностей и угроз;
- декомпозиция (англ. *Decomposition*);
- метод Дельфи (англ. *Delphi method*);
- метод критического пути (англ. *Critical Path Methodology*);
- метод Монте-Карло (англ. *Monte Carlo Analysis*);
- мозговой штурм (англ. *Brainstorming*);
- планирование методом набегающей волны (англ. *Rolling Wave Planning*);
- диаграмма Ганта (англ. *Gantt Chart*);
- диаграмма Парето (англ. *Pareto Chart*).

В связи с ограниченным объемом пособия предлагаем читателю самостоятельно изучить соответствующие теоретические материалы.

Глава 9

ОСНОВНЫЕ ОШИБКИ ПРОЕКТНОГО УПРАВЛЕНИЯ

Выше рассматривались проблемы проектного управления и приводилась общая международная статистика успешности проектов в области информационных технологий. Низкий уровень успешности проектов, не превышающий 30 %, объясняется не только объективными факторами, которые находятся вне контроля участников проекта, но и их типовыми ошибками, устраняя которые, можно перевести проектную деятельность организации в «управляемую зону».

Нижеприведенный материал в части ошибок проектного управления в значительной степени базируется на личном опыте автора, хотя и подкреплен значительным количеством зарекомендовавших себя теоретических и научных работ, а также материалами ряда практических конференций. Поэтому в ряде случаев не удалось избежать личностных оценок.

Проект является деятельностью, заключенной в четкие временные рамки, имеющей ограниченный состав последовательных этапов. При этом ошибки, допускаемые на ранних этапах проекта, обходятся гораздо дороже с точки зрения их исправления. Например, считается, что ошибка, допущенная при определении требований к системе, в 200 раз дороже ошибки, допущенной на завершающих этапах работ.

Для лучшего восприятия материала рассмотрим типовые ошибки проектного управления в соответствии с их приоритетом и в последовательности этапов проектных работ, начиная с предпроектного этапа, на котором определяется общая методология проектной деятельности организации.

9.1. Необоснованная замена текущей деятельности проектной формой

В настоящее время проектное управление стало по-своему «модным». Организация проекта во многих случаях воспринимается как панацея для лечения любых проблем компаний и достижения любых стратегических целей. При этом корпорации часто не отдают себе отчет в том, что проектная форма организации работы в большинстве случаев бывает необоснованной.

Часто приходится наблюдать ситуации, когда предприятия пробуют за счет открываемых проектов подменить или скомпенсировать недостаточную результативность неэффективной текущей деятельности в рамках типового технологического процесса. Отсюда часто одним из самых сложных для них является вопрос о необходимости открытия проекта в каждом отдельном случае — отличительные признаки текущей деятельности для такой организации составляют не меньший вопрос, чем проектной, поэтому оказывается размыта и граница принятия решения. Любая скоординированная по времени деятельность сразу же распознается в качестве проектной.

Не имея возможности обеспечить контролируемый и эффективный технологический процесс по типовым продуктам, такие организации пробуют сконцентрировать контроль, финансы и человеческие ресурсы на отдельных проектных участках. На первый взгляд подобные намерения логически обоснованы, но нередко такие попытки приводят к безрезультатной неэффективной растрате дефицитных ресурсов в связи с тем, что проектная работа организуется на том же уровне зрелости, как и текущая деятельность.

Проектная деятельность связана с концентрацией избыточных ресурсов на приоритетных направлениях. Аналогично наступательным военным действиям организация проектной группы предполагает создание качественно и количественно превосходящего обычные потребности исполнительского ресурса, сосредоточенного на узкой цели на ограниченный период времени. При открытии проекта наиболее квалифицированные кадры частично выводятся из действующей иерархии подчиненности, создавая дополнительный кадровый дефицит и конфликтуя за исполнительские и руководящие кадры с текущими процессами поддержки систем и их эволюционного развития. Проектным задачам присваивается высший приоритет, что часто приводит к необоснованному перераспределению вычислительных ресурсов. Компания начинает нести дополнительные накладные расходы на организацию различных проектных офисов и поддержку проектной методологии. В результате реализуемый проект, часто не достигая заявленных целей, наносит прямой ущерб текущей деятельности, что при расширенном портфеле проектов может полностью блокировать поступательное развитие организации.

Здесь следует отметить, что аналогичные затраты на текущую деятельность и усилия по ее развитию несут значительно меньший риск и почти всегда являются более продуктивными. В общем можно сказать, что в виде проекта должно реализовываться только то, что не может реализовываться в виде текущей деятельности.

При достижении высоких уровней зрелости процессов (например, по модели *СММТ*), организации естественным образом выходят из области проектного управления в процессную. При постановке целей проектные задачи автоматически разбираются на подзадачи

процессными областями и реализуются в рамках текущей отлаженной координации.

На практике приходится встречаться со следующими примерами ошибочных критериев для выделения проектной деятельности:

- *целью работ является создание в периметре организации новой системы автоматизации.* Ошибочная цепочка рассуждений выглядит следующим образом: «Новая — значит, нетиповая и, значит, должна создаваться в рамках проекта». При этом руководители, принимающие решение об открытии проекта, не осознают, что накладные проектные процедуры могут оказаться более затратными, чем непосредственно выполнение необходимых работ;

- *расчетные плановые затраты на работы, лицензии, вычислительные ресурсы, коммуникационные каналы и т. п. превышают лимит в 5 млн руб. (10 млн, 15 млн руб. и т. п.).* Дополнительный контроль таким затратам не мешает. Поэтому следует открыть проект для контроля сроков, затрат и возложения дополнительной ответственности на руководителя проекта. В данном случае подразделения, в чьи функции входит контроль обоснованности расходов и бюджетных ограничений, расписываются в своей низкой эффективности, рассчитывая возложить на взаимодействующих участников проекта дополнительный контроль в его рамках. Однако каким образом можно наладить эффективный контроль по десяткам открываемых проектов, не умея его налаживать в рамках рутинной текущей деятельности? Часто данная ситуация осложняется попытками контролеров, подпавших под иллюзию дополнительного контроля, «упаковать в проектную оболочку» все расширяющуюся область мелких типовых задач. Им необоснованно кажется, что бюджетные ограничения проекта работают лучше обычного бюджетного контроля, построенного на временных периодах. В результате получаем две пересекающиеся, а иногда и конфликтующие системы контроля, работающие одинаково неэффективно и требующие одновременной поддержки.

9.2. Недоценка сложности проекта

Сложные системы выходят далеко за рамки упрощенных подходов к их анализу и созданию. Каждый успешный проект реализуется на пределе риска неуспеха. Ни в каком случае не следует поддаваться намерениям необоснованно усложнить организационное построение команды проекта или расширить набор требований к результату. Это гарантированно выводит проект за пределы допустимого риска.

В гл. 2 рассматривались проблемы, связанные со сложностью технических систем. При открытии проектов их часто недооценивают. Проектная форма организации работ представляется гарантированным способом решения сложных задач. Между тем следует принимать во внимание то, что чем сложнее проект, тем меньше у него шансов

на успешное завершение. Более того, в теории, развиваемой на основе модели зрелости *СММІ*, достаточно обоснованно утверждается, что сложность проектов, которые могут успешно реализовываться на низких уровнях зрелости (первый и второй), значительно ниже сложности проектов, допустимых для верхних уровней зрелости (с третьего по пятый).

Именно поэтому необходимо критически подходить к утверждениям о высокой перспективности некоторого проекта для организации *A*, который был успешно реализован в организации *B*. Для начала желательно объективно сравнить уровни эффективности отдельных процессов, в различных организациях.

В начале гл. 5 приведена международная статистика успешности проектов в зависимости от их масштаба (табл. 5.2). Реальная граница сложности, которая достижима в проектах для большинства организаций, расположена в диапазоне от 0,4 до 1 млн долларов. При достижении масштаба проекта цифры в 1,3 млн долл. вероятность его успеха падает с 44 до 7 % (*waterfall*). Если объективная оценка эффективности и зрелости процессов позиционирует организацию на 1—2 уровень, что является нормой для большинства российских компаний, то эту границу желательно дополнительно занижать при рассмотрении вопроса об открытии проекта.

Приведенная статистика выглядит не настолько негативно для организаций, применяющих *agile*-подход. Однако следует принимать во внимание, что этот подход выработан на основе высокоэффективных методологий быстрой разработки и не может использоваться эффективно организациями, находящимися на нижних уровнях зрелости с неразвитыми процессами.

9.3. Ошибка определения цели проекта

Ошибки формулирования цели являются наиболее часто встречающимися в настоящее время. На что они влияют?

В ходе реализации любого сложного проекта многократно принимаются различные организационные и технические решения. Инженерная задача — это поиск оптимума при наличии ограничений, принятие некоего технического решения, обеспечивающего максимизацию результата, и его реализация. В роли ограничений выступают сроки, финансы, человеческие и технические ресурсы, выбранные функциональные ограничения и т. д. Из математики известно, что задачи оптимизации являются решаемыми только при наличии целевой функции, например, надежности.

Стандарт *ANSI PMI PMBOK* для упрощения требует явного задания единой цели проекта, что приводит проект к состоянию, в котором проектные решения принципиально могут быть приняты: проектная

группа получает четкий критерий выбора оптимального варианта решения. Варианты организационных и технических решений рассматриваются относительно того, насколько они приближают к достижению цели проекта или ей препятствуют.

Если цель задана ошибочно или недостаточно точно, то на каждом этапе работ будет создаваться ошибка в принимаемых решениях. Это особенно критично для начальных этапов. Известно, что ошибки, допущенные в начале проекта, имеют максимальную стоимость устранения — большой объем работ потребует переделки.

Практика показывает, что требование одной цели не всегда является обоснованным. Проект может реализовываться для достижения нескольких целей, но в этом случае должен быть явно сформулирован приоритет этих целей, что также делает возможным обоснованный выбор отдельных проектных решений. Иначе в условиях обычного дефицита ресурсов проекта его руководитель и другие участники (лица и подразделения, затрагиваемые проектом) лишаются возможности обоснованного принятия решений, а при наличии расходящихся целей, что является вполне естественным, попадают в зону конфликтного взаимодействия. При этом вопрос будет решен не оптимально технически или организационно, а исходя из превосходства административного ресурса или настойчивости одной из сторон.

В части ошибки при формулировке цели показателен пример проекта, реализованного в одном из инфраструктурных финансовых институтов.

На момент привлечения автора к проекту работы велись уже четыре года. В качестве исполнителя выступала крупная индийская компания, располагающая большим опытом аналогичных проектов и соответствующим достаточно развитым программным продуктом.

Цель проекта формулировалась следующим : «Модернизировать технологическую платформу для повышения масштабируемости, функциональности и надежности». Легко заметить, что цель не является измеримой, что обоснованно требуют все руководства по управлению проектами. Не может быть создан технический объект, для которого не заданы количественно ограничения функциональности и надежности; такой результат не будет достигнут никогда.

Кроме того, не заданы приоритеты целей. При работе над проектом неоднократно принимались решения, затрагивающие одновременно различные характеристики системы. В различные моменты времени приоритет отдавался различным характеристикам. Легко представить, как выглядел путь оптимизации с постоянно изменяющимся направлением движения.

Отсутствием приоритета и измеримости целей ошибки формулирования целей проекта не ограничивались. После короткого обследования выяснилось, что руководство компании вошло в конфликт с поставщиком эксплуатируемой системы и настоящей целью про-

екта было замещение существующего ключевого звена программного комплекса на новое. То есть настоящая цель проекта должна была на уровне официальных документов формулироваться как «замещение в минимальный срок программного обеспечения системы без потери функциональности, производительности и масштабируемости». В результате участники проекта и члены рабочей группы, занимаясь проектом высокой сложности, были лишены возможности выбирать оптимальные проектные решения не только коллективно, но и на персональном уровне.

После корректировки цели проекта пришлось переделывать всю проектно-технологическую документацию, разработанное программное обеспечение не использовалось в дальнейшем. При этом проект оказался успешно завершен после реформирования в рекордные для такого уровня сложности 1,5 года.

Наиболее частая ошибка в настоящее время, наряду с множественностью формулировки цели и отсутствием задания приоритетов, — формулировка цели проекта как «внедрение автоматизированной системы *N*».

Программное обеспечение является элементом автоматизированного комплекса предприятия, выполняющего заданный набор функций. Внедрение очередного элемента в этот комплекс не может быть самодостаточной целью. Целью может служить достижение им некоторых задаваемых характеристик: функциональности, надежности, масштабируемости, модернизируемости и т. д.

Если бы в рамках советского космического проекта формулировались цели внедрения какого-либо конкретного двигателя в космический аппарат, а не выведение в космос космического аппарата с человеком на борту, то, скорее всего, этот проект не был бы реализован до сих пор. При этом в рамках ИТ-проектов почему-то «внедрение» часто признается допустимой целью.

Следует отметить, что иногда потребность в подобной формулировке возникает. Обычно реальная цель в этом случае будет формулироваться как «демонстрация кому-либо внедрения системы или способности руководителя проекта внедрить сложную систему в карьерных целях». Например, компания планирует первичное размещение акций, и консультанты рекомендуют публично закупить и установить международно распознаваемую систему управления финансами для демонстрации потенциальным инвесторам своей финансовой прозрачности и технологичности.

В случае подобной формулировки для достижения цели проекта руководителю должно быть достаточно подписать акт сдачи-приемки программного обеспечения, закрывающий договор закупки. То есть получить формальное подтверждение внедрения, с минимизацией затрат на реальную интеграцию и настройку системы. Более того, расширение усилий по встраиванию системы в технологические процессы

организации потребует не только дополнительных затрат, но и приведет к росту пассивного сопротивления коллектива — любая сложная система естественно сопротивляется изменениям. А значит, шансы на успешное завершение проекта снизятся.

На практике встречаются примеры успешного применения данного подхода, когда руководитель проекта, корректно понимая реальную цель «внедрения», сознательно использует особенности сложившейся ситуации для реализации проекта. Например, в 2001 г. перед одной из крупнейших российских компаний головной организацией холдинга была поставлена задача внедрить систему *SAP R/3* класса *Enterprise resource planning* (система управления деятельностью корпорации). Основным требованием руководства холдинга было обеспечить отражение финансовой и бухгалтерской отчетности дочерней компании в автоматизированной системе головной компании.

Типовым подходом к реализации проекта было бы полномасштабное внедрение достаточно дорогой системы *SAP R/3* с полным замещением системы, эксплуатировавшейся на тот момент. При этом проекты внедрения систем *ERP* заслуженно пользуются репутацией сложных, затратных и рискованных. Руководителю проекта удалось убедить руководство компании в оптимальности подхода, при котором на начальном этапе *SAP R/3* устанавливалась бы в минимальной конфигурации без подключения пользователей с обеспечением выгрузки данных из старой системы и перегрузкой на сервер головной организации.

Внедрение было реализовано в минимальный срок, с низкими затратами, без переобучения бухгалтерских и финансовых подразделений с полным достижением основных целей.

В дополнение следует отметить такой недостаток формулировки цели «внедрение»: в этом случае нарушается баланс прав и ответственности руководителя проекта. Мало того, что ему была навязана некорректная цель. Он лишается возможности выбора оптимального проектно-технического решения в части ключевого элемента системы при сохранении ответственности за проект.

Еще одной часто встречающейся ошибочной целью проекта, которая при этом редко формулируется в явной форме, является замещение существующей группы разработчиков вместе с системой, которую они развивают в рамках текущей деятельности. Иногда такая цель может иметь место, например, при шантаже со стороны разработчиков. Однако в большинстве случаев все претензии сводятся к недостаткам эффективности процессов развития и сопровождения существующей системы или к недостатку ресурсов, выделенных на это технологическое направление.

Помимо понятия справедливости руководителю проекта и лицам, принимающим решение о его открытии, следует принимать во внимание, что за неэффективность процессов несут ответственность

не разработчики, а руководство ИТ-блока, и повышение эффективности процессов — это предмет развития в рамках ранее описанного цикла Деминга. Подобные проблемы не решаются за счет внедрения новых систем.

Кроме того, в подавляющем большинстве случаев именно «старая неэффективная» команда становится наиболее востребованным в новом проекте ресурсом исполнителей, а старая «плохо работающая» система становится в рамках проекта основным источником данных и функциональных наработок.

На практике большинство подобных проектов удается вывести из зоны риска путем организационного слияния команд проекта и замещаемой системы, иногда с передачей лидирующих функций старой команде.

9.4. Ошибка определения ограничений проекта

Задачи поиска оптимума для сложных функций по большей части не имеют решения, если не заданы необходимые ограничения. По ходу выполнения проектов приходится периодически принимать решения по выбору технических и организационных вариантов реализации. Если ограничения не заданы или их недостаточно, решения либо не будут приниматься, либо будут приниматься с ошибками.

Сама по себе проектная деятельность в обязательном порядке связана с ограничением по времени. В этом ее основное отличие от текущей деятельности. При этом, реализуя проекты, часто забывают о необходимости максимально ограничить не только временные параметры проекта, но и функциональные.

Обычно заказчик, инвестирующий значительные суммы в проект, требует от исполнителей максимизировать функциональные и эксплуатационные характеристики будущего результата. Например, для какой-либо автоматизированной системы это часто выражается в задании требования создать «настраиваемую универсальную подсистему разграничения доступа» или «возможность загрузки данных из общеупотребимых форматов файлов». В этом случае заказчик сам увеличивает риск неуспеха проекта. Для случая эксплуатационных характеристик особое значение имеет корректное задание надежностных характеристик. Завышенные требования к надежности могут увеличить затраты денежных, временных и человеческих ресурсов не пропорционально, а на порядок.

Другой типичный случай ошибки — когда функциональность и другие параметры результата не привязываются к конкретной точке во времени. Характеристики задаются для автоматизированной системы без учета того, что старт жизненного цикла прикладной автоматизированной системы (результата проекта) в любом случае связан с меньшей функциональностью и надежностью, чем периоды, отстоящие от него

на месяцы. Будущие характеристики системы никогда не следует задавать для первичного результата проекта.

Если бы при разработке первых версий зенитных ракетных комплексов СССР С-75 (1957) не были жестко заданы ограничения по предельным высотам (3—22 км), то работы над ним проводились бы безрезультатно по настоящее время.

9.5. Необоснованная передача внешней стороне функций руководства проектом

Руководство проектом заключается в координации и контроле действий участников проекта для достижения целей проекта. Цели организации-заказчика и организации-исполнителя принципиально различны. Заказчик стремится получить качественный результат с минимизацией рисков и затрат. Основной целью исполнителя в проекте является получение прибыли. Достижение целей другой стороны контракта для него не может стать приоритетной целью. Даже в условиях, когда организация-исполнитель стремится повышать приоритетность целей заказчика в своей деятельности, эти приоритеты целей заказчика постоянно подвергаются сомнению в ходе реализации проектов.

Именно в случае передачи управления проектами третьей стороне (англ. *outsourcing*) изменения требований всегда трактуются как риск и в обязательном случае перекладываются на затраты заказчика, причиной всех задержек проекта начинает считаться сторона заказчика, при этом сроки проектов растут, увеличивая доход исполнителя и т. п.

Руководитель проекта в рамках своей деятельности должен постоянно компенсировать риски проекта и принципиальную непредсказуемость сложных организационных и технических систем. Его истинные цели не совпадают с целями заказчика. Естественно, что во всех сложных случаях он склоняет баланс проектных решений в сторону увеличения своего дохода.

9.6. Ошибка определения требований проекта

Данный тип ошибок непосредственно связан с предыдущим. Следует понимать, что в настоящее время не существует методов и инструментов точного описания функциональных требований. Функциональные требования всегда задаются с некоторой ошибкой. Существуют процедуры и инструменты выявления несоответствий разработанного программного обеспечения и требований — это тестирование, или верификация разработанного программного обеспечения и реальной жизни (валидация или опытная эксплуатация). Надежных методов устранения ошибок постановки требований не существует (рис. 9.1).

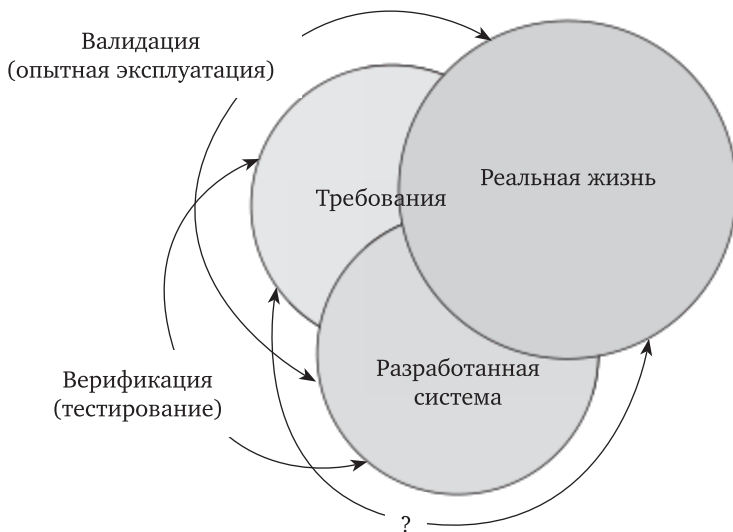


Рис. 9.1. Несоответствие реальной жизни, требований проекта и разработанной системы

Кроме того, в случае превышения срока проекта 4—6 месяцев реальные требования жизни неизбежно начнут меняться. Современные сложные системы в любом случае динамично развиваются. Реальная жизнь не ждет завершения реализуемых проектов, заставляя развиваться требования к системам даже до их запуска в эксплуатацию.

9.7. Некорректное определение рисков проекта

Мы знаем, что не существует методик безошибочного задания требований к сложным прикладным автоматизированным системам. Кроме того, жизнь заставляет корректировать требования к системам до завершения проекта. Таким образом, риск изменения требований, обычно фиксируемый в проектах как основной, риском не является. Вероятность уточнения требований по ходу проекта равна 100 %. Риски реализуются с вероятностями менее 100 %. В качестве такого риска можно задавать не риск изменения требований, а риск изменения требований, например, в объеме, превышающем трудоемкость 1000 чел/дней.

На практике часто приходится встречать ситуации, когда проектный офис стремится в материалах на каждый проект включать список основных рисков, присущих данному проекту. На деле этот список оказывается полностью однотипным для всего портфеля проектов. Скорее, риски являются следствием низкого уровня зрелости и эффективности организации. Данные риски можно вывести в единый нормативно-методологический документ, определяющий проектную деятельность в организации, но при этом утрачивается ценная информация о рисках, индивидуальных для отдельного проекта.

9.8. Ошибка определения списка участников проекта

Ошибки определения требований к проекту являются наиболее дорогостоящими. Надежных методов их минимизации, кроме согласования с предполагаемыми пользователями (заказчиками), не существует. Таким образом, допуская ошибку при определении возможных участников проекта, мы заранее увеличиваем потенциальную ошибку определения требований.

Достаточно часто участники проекта сознательно сужают круг привлекаемых к проекту в надежде упростить процедуры согласования требований и другой проектной документации, а также пытаются облегчить процедуры приемки результата проекта. Однако практически всегда подобные проекты сталкиваются с необходимостью корректировки на завершающих фазах, поскольку оказывается, что система нуждается во взаимодействии с дополнительным кругом пользователей или других систем.

Особенно часто подобная ошибка заключается в исключении из списка участников проекта службы защиты информации. На завершающем этапе приемки и установки системы в любом случае служба защиты информации будет привлекаться к участию в тестировании или к установке в промышленную среду. Поскольку именно эта служба в крупных организациях накладывает определяющие требования на ядро систем в части протоколирования, взаимодействие компонент, подсистему разграничения прав и т. д., то с высокой вероятностью отсутствие согласования с ней на начальных фазах проекта приведет к масштабной корректировке на завершающих этапах.

Другая ошибка, связанная с определением состава участников проекта, — это необоснованное расширение их списка. В подобных случаях резко возрастают затраты ресурсов на согласование проектных документов и другие виды коммуникации. Кроме того, может быть необоснованно расширен список требований, реализация дополнительно усложнится и проект практически наверняка будет завершён неуспешно.

9.9. Ошибка выбора руководителя проекта

Наиболее часто встречающаяся на практике ошибка подобного типа — это привлечение заказчика из бизнес-подразделения или высшего руководителя организации в качестве руководителя ИТ-проекта. Часто такая ошибка допускается проектным офисом намеренно. Заказчик, отвечая за проект в качестве руководителя, примет результат даже со значительными недостатками, поскольку приемку делает сам у себя. В этом случае не остается места для эффективных процедур контроля качества — QA (от англ. *quality assurance*). Кроме того, в этом случае,

за редким исключением, к руководству проектом привлекается человек, не компетентный в области ИТ-проектов. Таким образом, проектный офис намеренно заранее на систематической основе жертвует качеством проектов, особенно если привлечение заказчиков к руководству проектами закреплено в официально принятой методологии проектного управления.

9.10. Нарушение баланса прав и ответственности руководителя проекта

Утверждая паспорт¹ проекта, коллегиальный или единоличный орган управления организации заключает своего рода договор с руководителем проекта. Руководителю проекта предоставляется необходимый набор ресурсов. В обмен он обязуется достичь цель проекта с заданными показателями качества. У него есть определенный набор прав и ответственности.

Если сделка несправедлива изначально, то риски проекта значительно увеличиваются. Права руководителя проекта должны быть сбалансированы с его ответственностью.

На практике встречаются случаи нарушения баланса в обе стороны. Иногда руководителю проекта предоставляются значительные ресурсы, но цель задана неточно. У него появляется возможность сдать под видом результата проекта, например, некоторый объем проектных документов, не прошедших необходимые согласования, или автоматизированную систему, формально соответствующую требованиям, но неспособную выполнять полезные функции.

В обратном случае руководителю проекта при его инициации формально выделены ресурсы: финансовые, временные, человеческие, вычислительные. Однако по ходу реализации проекта он вынужден повторно согласовывать мельчайшие затраты с широким кругом контролирующих подразделений, контроль над проектом включает ежедневную и еженедельную отчетность, рабочая группа наполнена сотрудниками, занятыми на 100 % непроектной деятельностью, на создание тестовых стендов уходят месяцы проектной и административной работы и, в довершение всего, за руководителя заранее выбрали основное программное обеспечение, которое он обязывается внедрить.

Таким образом, руководитель проекта фактически оказывается лишен не только выделенных на проект ресурсов, но и даже основного права инженера — выбрать оптимальное техническое решение, которое обеспечит успех его проекта. При этом заказчики ожидают, что проект создаст необходимую им функциональность. Эта ситуация явно

¹ Может называться по-разному, например, уставом.

дискриминирует ответственного руководителя проекта. Баланс прав и ответственности нарушен. Единственный выход руководителя проекта — сдать заказчику результат, который соответствует по качеству объему предоставленных прав. То есть, с точки зрения заказчика, проект с высокой вероятностью не будет успешным.

9.11. Усложнение организационного построения проекта (включая необоснованное наращивание численности проектной группы)

С одной стороны, сложная система не может быть создана простым (малым) коллективом, даже за значительный период времени. Чем сложнее система, тем сложнее оказывается организация коллектива, способного ее проанализировать, спроектировать и ввести в действие. С другой стороны, система, с которой приходится взаимодействовать участникам проекта, включает не только объект их работы, но и организационную структуру, обеспечивающую реализацию проекта.

Ранее было показано, что работа со сложными системами производится на грани реализуемости и предсказуемости результата. Вероятность неуспеха крайне велика. При этом в проектной практике часто встречаются случаи, когда организационное построение исполнительной структуры дополнительно необоснованно усложняют за счет привлечения крупных аудиторов, страховщиков, закупки услуг по управлению проектом, закупки внешних работ по созданию проектной документации и программного обеспечения, с которыми могли справиться сами, а также за счет создания многоуровневой управленческой пирамиды и сложной системы коммуникации с набором разнообразных управляющих комитетов, координаторов по работе с поставщиками, с бизнесом и т. д.

Такой проект будет неуспешен с высокой вероятностью, если по ходу реализации проектная группа (реальные исполнители) не сможет значительно упростить структуру управления и коммуникации — все ресурсы реальных исполнителей по проекту будут затрачены на избыточную коммуникацию, согласование сложных договоров, которые повышают шансы на успех только на первый взгляд, а также на предоставление избыточной управляющей структуре формальных результатов по ходу работ.

В случае наращивания проектной группы за счет различного рода формальных участников (чье участие в проекте ограничивается единицами процентов рабочего времени), контролеров и согласующих, во-первых, затрудняется выработка решений проектной группой — лишние участники скорее мешают, чем помогают, во-вторых, для руководителя проекта усложняется деятельность по управлению и контролю. При этом в случае неуспешности проекта значительный размер проектной группы (выделенный ресурс) будет поставлен руководителем в вину вне зависимости от меры занятости отдельных участников.

На практике приходится сталкиваться с проектными группами, в которых формально числилось (и претендовало на участие в разделе итоговой премии) до 80 человек, а реально работало над проектом не более 10.

Для исключения формальных участников приходится изобретать различные методики «чистки рядов». Согласование и контроль должны выполняться при взаимодействии проектной группы с другими структурными единицами организации в рамках внешней коммуникации.

9.12. Передача проектной группе полномочий по утверждению проектных документов

Реализация проекта — это всегда создание уникального нового результата для организации. Новый результат должен заставить организацию адаптироваться к нему, т. е. условия функционирования организации меняются. Чем результат проекта новее и масштабнее, тем большую работу по адаптации к нему должна выполнить вся организация, все взаимодействующие с проектом структурные единицы.

Для любых сложных систем, включая живые, естественным является сопротивление изменению условий, сопротивление новому. Причина в том, что адаптация к новому требует затрат ресурсов. В проектной практике сопротивление выражается во множественных конфликтах при согласовании проектных документов и решений, поэтому все проектные группы склонны уклоняться от сопротивления организации. Для этого они «прячут» свои решения и свою деятельность от взаимодействующих подразделений, однако процесс адаптации организации к проекту должен быть выполнен. Он является ресурсоемким, требует времени и не может быть совершен в момент приемки результатов проекта.

Если проектная группа будет уклоняться от взаимодействия с организацией, реализующей проект, то она столкнется с накопленным сопротивлением и конфликтом на завершающем этапе проекта. При этом сопротивление может превысить ресурсы проектной группы — в конфликте выигрывает сторона, располагающая большими ресурсами. Ресурсы проектной группы редко бывают достаточно велики, а к завершающим этапам проекта обычно снижается и административный ресурс, выражающийся в поддержке руководства. В таком случае проект, несмотря на готовность результата, будет признан неуспешным и закрыт.

Дополнительно необходимо отметить, что ряд исследований в области конфликтов при управлении проектами показал прямую зависимость падения качества результата проекта от возрастания уровня конфликтов в проекте¹. Таким образом, накапливая конфликт, руково-

¹ Чекмарев А. В. Анализ конфликтов в управлении проектами и оценке уровня зрелости на основе СММІ и РМІ РМВОК. LAP, 2012.

дитель проекта накапливает риск падения качества результата проекта или риск его неуспешного закрытия.

Для избегания подобной ситуации служит разбиение процесса принятия проектных решений на этапы с последовательным их согласованием. Причем согласование должно выполняться со всеми структурными единицами, на чью деятельность будет влиять результат проекта. В этом случае заинтересованные подразделения будут иметь возможность выполнить свою часть работы по адаптации и, последовательно согласовывая отдельные проектные решения, потеряют возможность отказаться от приемки результата на завершающих этапах проекта. Поэтому предоставление проектной группе права самостоятельно принимать проектные решения, уклоняться от конфликтов при их согласовании и за счет этого экономить ресурсы не увеличивает эффективность проектов, но значительно увеличивает риски проектов.

Иногда предоставление проектным группам права утверждать проектные документы без дополнительного согласования с заинтересованными сторонами вынуждает их включать в состав проектных групп руководителей заинтересованных подразделений для контроля решений. Если от них включены рядовые исполнители, то руководитель проектной группы практически всегда может игнорировать их мнение. С руководителями подразделений в составе проектных групп это становится затруднительным, но в этом случае получаем ситуацию, приведенную в предыдущем параграфе — исполнительский слаженный коллектив проектной группы размывается координаторами и контролерами, что негативно сказывается на ее работе.

Кроме того, как было показано выше, уклонение от согласования с заинтересованными подразделениями может привести к эффекту, сходному с ошибкой при определении состава участников проекта, и таким образом увеличить размер ошибки при определении требований и принятии решений. Это дополнительно увеличит риск неуспеха.

9.13. Недостаток взаимодействия и коммуникации со стороны проектной группы

На рис. 9.2 представлена сравнительная диаграмма уровня конфликта, являющегося следствием адаптационных процессов организации к реализуемому проекту. Слева показан обычный для неэффективной проектной практики «седловидный» график — конфликт достигает максимума при инициации проекта и при приемке результата. Обычно на этапе инициации заинтересованные стороны спорят из-за выделяемых ресурсов, назначения руководителя проекта, выбора закупаемого программного обеспечения и т. п. Затем проектная группа «прячется» от взаимодействия, минимизируя избыточные, с ее точки зрения, контакты и экономя ресурсы.



Рис. 9.2. Уровень конфликта при отказе от коммуникации в период реализации и при регулярной коммуникации и согласовании решений

При достижении определенной готовности результата проекта проектная группа вынуждена предъявить этот результат заинтересованным структурным единицам организации. В этот момент обычно выясняется, что для части заинтересованных сторон результат работы является неприемлемым (с ними не согласовывали проектные документы) и для всех этот результат проекта, как минимум, неудобен — у них не было времени на то, чтобы к нему адаптироваться и соответствующим образом перестроить свою деятельность.

Отторжение результата проекта и временами даже саботаж со стороны подразделений и сотрудников, затрагиваемых проектом, не являются результатом лени или личных пристрастий. Конфликты в этих случаях являются системным явлением, и при правильно выстроенной проектной группой регулярной коммуникации с заинтересованными сторонами их уровень не превышает опасных для проекта значений. Справа на рис. 9.2 представлен график уровня конфликта при регулярной эффективной коммуникации проектной группы с взаимодействующими подразделениями. Поэтапное согласование документов и сотрудничество обеспечивают снижение уровня конфликтов и непрерывную адаптацию организации к результату проекта. К моменту завершения проекта организация оказывается готова его принять.

9.14. Ритуализация

Реализация проектов проходит в различных условиях. Организации, применяющие проектную форму работ, находятся на различных уровнях зрелости. Проекты различаются масштабом и направленно-

стью. Сам по себе проектный результат всегда уникален для организации. При этом достаточно часто приходится встречаться со случаями, когда неопытный руководитель проекта или проектного офиса, получив первичные знания о проектном управлении или поучаствовав в проектной работе в другой компании, пробует реализовать «лучшие», по его представлению, практики без учета сложившихся вокруг него условий и не принимая во внимание некоторые «мелочи» проектной методологии.

В этом случае некрупные проекты обзаводятся «управляющим комитетом» и набором из 20 и более ролей специалистов в проектной группе, разработанная документация становится самоцелью и оценивается по объему, а не по ценности содержания, коммуникация с широким кругом причастных обеспечивается с завидной регулярностью при полном отсутствии полезного результата совещаний, при тестировании в качестве критичных фиксируются малейшие замечания, базовые (контрольные) планы прорабатываются до уровня детальных рабочих, акты сдачи-приемки регулярно подписываются без указания целей приемки и т. д.

Причиной подобных ошибок является недостаток профессионализма руководителя проекта, который обеспечивается не только формальными знаниями, но и опытом. Руководить проектом эффективно может только специалист, который ранее успешно реализовывал подобные проекты, возможно, меньшего масштаба. Кроме всего прочего, этот специалист располагает необходимым набором психологических характеристик, например таких, как «удержание цели», «завершение»¹.

Новичок, не обладающий необходимым опытом и психотипом, будет реализовывать свое представление о проектных методиках, не понимая их возможных модификаций и назначения. Стандарты и методики в этом случае превратятся в ритуалы, которые не смогут обеспечить достижение результата.

Выявить подобный подход бывает очень просто — достаточно руководителю проекта, когда он требует выполнения некоторого общепроектного действия, например, разработки требований, задать вопрос: «Зачем?» Если ответом будет ссылка на «стандарты» и «лучшие практики», значит, руководитель не понимает, зачем определенные проектные методики применяются, в чем их реальная задача и ценность.

В случае гарантированного неуспеха проекта такой руководитель искренне недоумевает — почему проект не достиг своих целей, ведь он делал все правильно, но «ритуалы» и «заклинания», работающие у других, его подвели? Все спишет на негативные сложившиеся независимо от него внешние условия и будет прав. Он плохой руководитель проекта, у него нет способности влиять на внешние условия и эффективно с ними взаимодействовать.

¹ Ben-Hur S. Talent Intelligence: What You Need to Know to Identify and Measure Talent.

9.15. Формальное построение контроля на основе отчета руководителя проекта

В реальной жизни руководитель проекта (как и любой ответственный исполнитель нестандартной работы) на вопрос о прогрессе со стороны высшего руководства обычно, мягко говоря, преувеличивает. В целом ситуацию хорошо описывает правило 10—50—90. Если руководитель утверждает, что уже приступил к активной работе (10 %), значит, еще даже не анализировал задачу. Если говорит, что проект на середине реализации (50 %), значит, только приступил. Если руководитель с уверенностью рапортует о завершающих шагах (90 %), смело можно прогнозировать длительность и затраты на завершение проекта, сравнимые с предыдущими¹.

Связана данная ситуация прежде всего с тем, что проектная деятельность, в отличие от текущей, плохо поддается независимым количественным измерениям. В рамках проекта создается уникальный результат, который мы пока не умеем измерять. Во вторую очередь, конечно, проявляется вынужденный оптимизм руководителя проекта, который должен заставить организацию «привыкнуть» к мысли о неизбежности успешного результата и предпринять шаги по адаптации к нему.

В итоге все попытки построить достоверную оценку прогресса по портфелю проектов на основе периодических отчетов их руководителей оказываются безуспешными. Несмотря на объемные папки накопленных отчетов, в соответствии с которыми проекты реализуются в установленные сроки, практически каждый проект оказывается либо неуспешным, либо превышает сроки на месяцы и годы.

Если предоставленная в отчете информация не поддается сторонней проверке, значит, в отчете будет предоставляться ложная информация. Стороннему контролю поддается только распознаваемый полезный результат или документ, согласованный с незаинтересованной стороной. Таковым является, например, акт приемки, подписанный службой защиты информации (если они не являются исполнителями) или требования, согласованные со всеми заинтересованными подразделениями.

Помимо недостоверности, у периодической отчетности (в некоторых случаях еженедельной или даже ежедневной) есть дополнительный недостаток. Она требует значительных трудозатрат и «творческий подход» к ее составлению у руководителя проекта их значительно увеличивает.

Если результат этапа проекта был проконтролирован кем-то, значит, он может быть зарегистрирован немедленно после прохождения контроля в специальной базе данных. Отчет по состоянию проектов может строиться по этой базе данных без внесения дополнительной информации в момент построения отчета. Информация вводится в момент

¹ Верзух Э. Управление проектами: ускоренный курс по программе MBA.

стороннего контроля очередного распознаваемого результата, который должен появляться после каждого этапа работ, предусмотренного календарным базовым планом проекта. В этом случае отчетность всегда актуальна и не требует участия руководителя проекта.

9.16. Отсутствие системы мотивации участников проекта

В случае низкого уровня зрелости проектного управления и соответствующего недостаточного уровня контроля организация обычно не имеет возможности выстроить объективную систему мотивации на основе распознаваемых результатов проектов.

Качество результатов не поддается контролю, а момент фиксации результата оказывается растянут по времени. Например, программное обеспечение готово с точки зрения исполнителя, но присутствует большой список замечаний, продолжительный период устранения которых делает необоснованным вопрос о дополнительном финансовом стимулировании участников проекта. Затем этот период плавно перерастает в непрерывную эволюцию в целом некачественного продукта.

Часто системный характер подобной ситуации приводит к полному отказу от дополнительной финансовой мотивации всех участников проекта. Это, в свою очередь, негативно сказывается на проектах. Участники проекта всегда выполняют сверхнормативную работу. Они создают новый для организации результат. Взаимодействующие подразделения в основном пассивно или активно сопротивляются новшествам, и участникам проекта приходится преодолевать их сопротивление, затрачивать свои ресурсы и энергию на процессы адаптации организации.

Отсутствие дополнительного финансового мотивирования приводит либо к снижению активности участников проекта, либо к вымыванию из проектных команд наиболее трудоспособной части сотрудников. В этом случае риск неуспеха проекта приближается к 100 % — рядовые специалисты обычно не в состоянии справиться с проектной работой. В таких организациях наиболее способные специалисты начинают избегать проектной деятельности.

Кроме непосредственных исполнителей — участников проектов должны быть дополнительно мотивированы также и сотрудники подразделений-заказчиков, а также подразделений, принимающих участие в приемке и контроле результатов (служба защиты информации, эксплуатирующие подразделения).

Потенциальные пользователи новой системы сталкиваются с необходимостью адаптироваться к ней, переучиваться, проводить тестирование, параллельную эксплуатацию. Это увеличивает нагрузку. Выделение им доли финансового стимулирования не только справедливо, но и обеспечивает дополнительную ресурсную подпитку на персональном уровне, снижая риски проекта. Кроме того, финансовое стимули-

рование лишает сотрудников повода участвовать в пассивном саботаже изменений. Автору приходилось сталкиваться с ситуацией, когда критичные замечания к внедряемой системе в рамках масштабного проекта внезапно стали не критичными после выплаты принимающим результат сотрудникам премии в размере $\frac{1}{2}$ оклада, что в сумме составило не более 150 тыс. руб., или 1 % от суммы проекта.

Для эксплуатирующих подразделений и службы защиты информации новая система — это не только дополнительная нагрузка, но и риски — любые масштабные изменения увеличивают риск сбоя и уровень их ответственности. С учетом рисков для них естественно ужесточение приемочных процедур, что в условиях недостаточной регламентации процессов, характерных для низких уровней зрелости, приводит к полной блокировке приемки — эксплуатирующие подразделения и служба защиты информации сами устанавливают правила приемки и уровень жесткости контроля. Таким образом, даже функционально готовые результаты проектов не принимаются, а проект становится неуспешным.

9.17. Ошибка неиспользования уроков проекта

В ходе реализации проекты взаимодействуют с большинством процессов, протекающих в информационно-технологических подразделениях. Проекты не осуществляются в изоляции от организации. Реализация успешных проектов приводит к адаптации взаимодействующих с проектом и его результатами подразделений и процессов. Неуспешные проекты выявляют основные недостатки взаимодействующих процессов.

Адаптационные процессы проявляются в периодически возникающих конфликтах, уровень которых поддается экспертному оцениванию, а информация, которую можно получить от конфликтующих сторон при «усреднении», оказывается достаточно объективной¹. Если эту информацию не анализировать и не принимать меры по повышению эффективности процессов, низкий уровень развития которых провоцирует конфликты, то каждый последовательно реализуемый проект будет наталкиваться на одни и те же недостатки, «сжигая» ресурсы на бессмысленное противодействие и подвергаясь дополнительному риску, а организация лишается такого общеизвестного «двигателя» для развития, как естественный конфликт (вспомним диалектику).

Примером конфликтной ситуации может служить отсутствие регламента подготовки и согласования актов сдачи-приемки. Если в организации не установлены правила оформления данных документов, создаваемых на наиболее «конфликтоопасном» этапе проекта, то в рамках

¹ Чекмарев А. В. Анализ конфликтов в управлении проектами и оценке уровня зрелости на основе СММІ и РМІ РМВОК.

каждого проекта будет возникать спор о том, представители каких подразделений должны включаться в состав подписантов и какой смысл придается их подписи, а также самому документу.

На удивление часто, например, возникает вопрос о том, является ли внутренний акт сдачи-приемки безусловным поводом для оплаты внешнему поставщику и почему тестирующие продукт пользователи должны нести ответственность за приемку-оплату часто не ими выбранного программного обеспечения, в особенности когда предусматривается ответственность за крупный освоенный проектом бюджет.

В реальности акт сдачи-приемки может иметь различный смысл и далеко не всегда создает безусловный повод для оплаты и ответственности подписантов, но правила его подготовки, детализация данных вопросов должны быть установлены и понятны всем участникам проекта.

Глава 10

ГИБКИЕ МЕТОДИКИ УПРАВЛЕНИЯ ПРОЕКТАМИ (AGILE)

В предыдущих главах, посвященных сложным системам, управлению проектами, тенденциям развития информационных технологий, мы неоднократно касались отдельных факторов, связанных с гибкими подходами к разработке программного обеспечения.

Зародившись в индустрии разработки программного обеспечения, сегодня *agile*-подходы внедряются в самых разных областях деятельности, которые в основном касаются инноваций, невозможности точного планирования результата и творчества.

Вышедшая в 2017 г. новая шестая редакция *PMI PMBOK* включила в себя *Agile Practice Guide*¹, что явилось определенным прорывом в области методик гибкой разработки, предложив стандарт в области терминологии, описания предиктивных, итерационных, инкрементальных и гибких подходов. Кроме того, предложенный набор описаний и рекомендаций устранил имевшее место в течение долгого времени противопоставление так называемых *waterfall*- и *agile*-подходов. Однако *Agile Practice Guide* не содержит описания общих тенденций развития производства программного обеспечения и информационных технологий в целом, которые явились источником создания в 2001 г. Манифеста *agile*. Повторим и уточним основные источники рождения Манифеста.

10.1. Источники *agile*

Первым источником *agile*, как было показано в гл. 5, является нарастающее противоречие между ростом трудозатрат и времени на первичное создание автоматизированных систем и необходимостью их быстрого эволюционного развития (рис. 10.1)².

Из рис. 10.1 видно, что срок разработки прикладных промышленно используемых систем (на основе международной статистики) в 1970-х гг. составлял около одного года. В это же время срок жизни внедренной системы без изменений достигал семь лет. То есть систему

¹ Agile Practice Guide. Project Management Institute, 2017.

² Громов Г. Р. Очерки информационной технологии.

создавали около года, а затем около семи лет в нее не требовалось вносить существенных изменений.

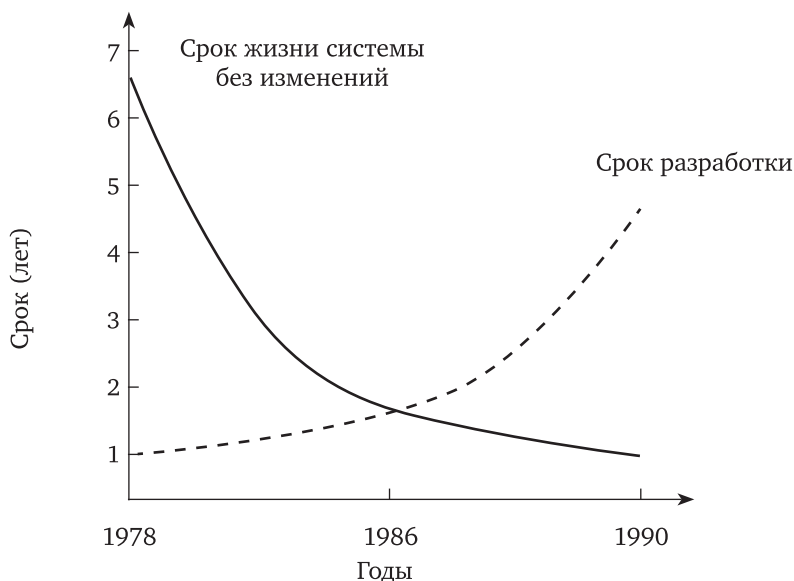


Рис. 10.1. Время первичной разработки и срок жизни системы без изменений промышленных прикладных систем

В середине 1980-х гг. срок разработки и сохранения функциональности сравнялся и составил 2—3 года. То есть непосредственно после внедрения новой системы начинался цикл ее обновления. Начиная с этого времени, срок разработки промышленных прикладных систем начал превышать срок их жизни без изменений.

Другими словами, современную большую прикладную систему невозможно разработать и внедрить, пользуясь традиционным линейным (*waterfall*, или предиктивным) подходом — требования к системам в большинстве случаев начинают изменяться задолго до финиша проекта. Отметим здесь год перелома — 1986.

Вторым источником является непрерывное усложнение автоматизированных систем. Производительность компьютеров нарастает, количество обрабатываемых объектов постоянно растет (с темпом около 1000 раз за 15 лет), автоматизированные системы объединяются в распределенные вычислительные комплексы, которые многократно усложняют анализ и проектирование взаимодействия компонент и при этом охватывают одновременно все большее количество процессов предприятия и даже групп предприятий.

Это приводит к тому, что характеристики и функционал новой создаваемой системы становятся непредсказуемыми не только в точке старта проекта, но и практически до самого его завершения — последние штрихи в характеристики продукта вносятся даже на этапе тестирования и опытной эксплуатации (валидации).

Кроме того, с ускоряющимся темпом изменяется и внешняя для предприятия конкурентная среда, внося свой вклад в неопределенность продукта. В настоящее время говорят уже не об инновационной эволюционирующей экономической и технологической среде, а о «подрывных» (англ. *disruptive*) инновациях. То есть о революционных непредсказуемых инновациях, которые оказывают драматическое воздействие на поступательно эволюционно развивающиеся предприятия.

Первый раз термин *disruptive innovation* был определен и проанализирован американским ученым Клейтоном Кристенсенем (*Clayton M. Christensen*) с коллегами в начале 1995 г. (отметим этот год).

Таким образом, сегодня требования к промышленно используемым крупным системам на старте проекта точно не могут быть определены в связи с непредсказуемостью и сложностью систем, а также постоянно меняющимися внешними условиями.

Хорошая иллюстрация данного фактора приведена в *Agile Practice Guide*¹. Приведенный в источнике рисунок несколько изменен с учетом информации табл. 5.2 (рис. 10.2).

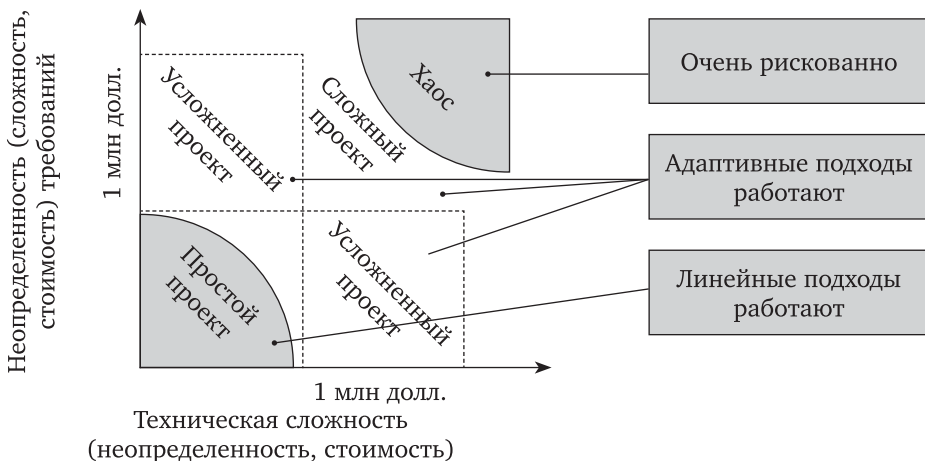


Рис. 10.2. Модель сложности и неопределенности (на основе *Agile Practice Guide* и статистики *Standish Group*)

Напомним, что в соответствии с международными статистическими данными *Standish Group* (табл. 5.2) явная граница применимости линейного (*waterfall, predictive*) подхода проходит ориентировочно по стоимости проекта около 1 млн долл. Проекты стоимостью до 0,4 млн долл., реализуемые с подходом *waterfall*, успешны в 44 % случаев. Проекты стоимостью от 1,3 млн долл., реализуемые с подходом *waterfall*, успешны всего в 7 % случаев.

Третьим источником является то, что, несмотря на ряд достаточно эффективных попыток, к настоящему времени не разработана мето-

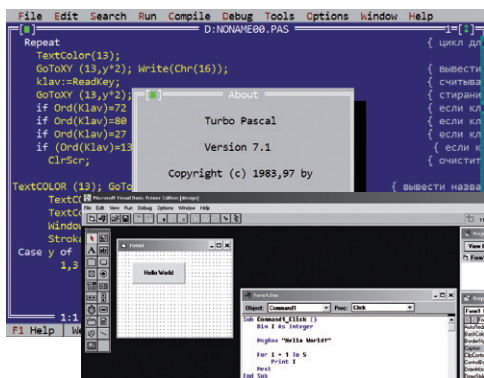
¹ Agile Practice Guide. Project Management Institute, 2017.

логия, язык и инструментарий описания бизнес-требований к автоматизированным системам (постоянно усложняющимся), которые позволили бы обеспечить единое прочтение пользователем и разработчиком программного обеспечения.

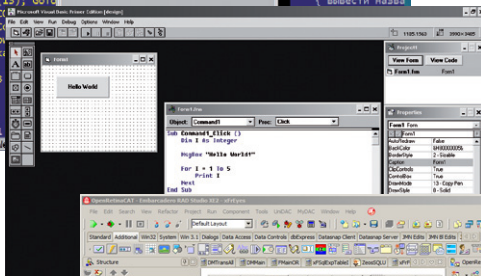
Таким языком и инструментарием является в настоящее время только непосредственно язык программирования и среда разработки, которые позволяют разработчику программного обеспечения в виде работающей программы или прототипа донести в однозначном виде свое представление о функционале до бизнес-пользователя (заказчика).

Затруднительность налаживания формализованного взаимодействия между программистом и заказчиком создала давление в направлении повышения эффективности непосредственного непрерывного и неформального общения между ними в ходе работы. Это позволяет, кроме всего прочего, обеспечить вовлечение эмоциональных способностей человека в рабочий процесс.

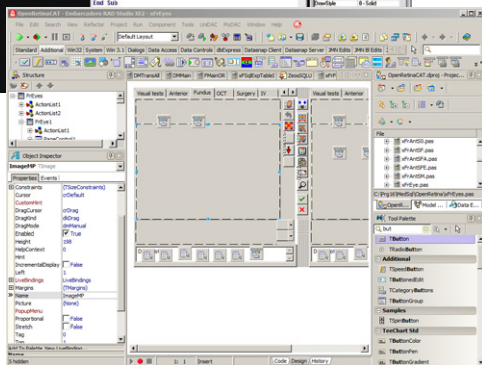
Технология «обратного показа» и прототипирования (приведенная в описании третьего источника), стала четвертым источником *agile* и развивается ускоренными темпами, начиная ориентировочно с 1983 г. Именно в этом году компанией *Borland* была представлена первая промышленная среда быстрой разработки программного обеспечения *Borland Turbo Pascal*. На рис. 10.3 представлена иллюстрация первичного развития сред быстрой разработки (англ. *Rapid Application Development*).



Borland Turbo Pascal 1983 г.
(компиляция по кнопке,
пошаговая отладка)



Microsoft Visual Basic 1991 г.
(макетирование графического
интерфейса)



Borland Delphi 1995 г.
(полнофункциональная
среда разработки
на объектно-ориентированном
языке
высокого уровня)

Рис. 10.3. Первый этап развития сред быстрой разработки ПО

Основными вехами этого периода явились: революционный *Borland Turbo Pascal* (1983), ориентированный на текстовый интерфейс и качественно изменивший труд программистов, *Microsoft Visual Basic* (1991), обеспечивший возможность визуального конструирования программных интерфейсов для среды *Windows*, а также *Borland Delphi* (1995), которая не только предложила расширенный инструментарий визуального проектирования программ, но и предоставила полный спектр средств языка высокого уровня, основанного на объектно-ориентированном подходе, что позволило практически полностью исключить необходимость параллельного использования невизуальных инструментов программирования.

Отличительной особенностью *RAD*-сред явилось многократное ускорение процедур написания приложений и возможность запуска «по кнопке» неполнофункционального прототипа будущей программы для просмотра ее внешнего «поведения» без доводки функциональности внутренних алгоритмов.

То есть эти среды были изначально направлены на проверку функциональности будущими пользователями по внешним признакам до завершения разработки, тестирования и даже до финализации требований.

Здесь следует обратить внимание на даты выпуска продуктов, а также на то, что *RAD*-среды «переворачивают» обычную последовательность жизненного цикла разработки. Разработку можно начинать до утверждения требований, а валидацию (проверку жизнью) — до завершения разработки либо тестирования.

Шестой источник связан с пятым и качественно расширил в свое время возможности инструментов разработки программного обеспечения за счет обеспечения возможности коллективной работы. Им стала выпущенная в 1986 г. система *CVS* (от англ. *Concurrent Versions System*) с последующими аналогами — система коллективного управления версиями, которая обеспечивает хранение истории изменений набора файлов исходного кода программного обеспечения и делает возможной совместную работу программистов над одним проектом. Представьте как могла бы выглядеть работа *agile*-команды без использования подобного инструмента и, следовательно, без возможности параллельной работы программистов: мы были бы вынуждены значительно удлинить периоды выпуска релизов, и инкрементальный подход, скорее всего, выродился бы в несколько ускоренный *waterfall*.

Дальнейшим развитием пятого и шестого факторов стала концепция *DevOps*, автоматизирующая полный цикл разработки программного обеспечения с ускоренным выпуском и установкой обновлений от регистрации задачи в системах управления исполнением до автоматизированного тестирования, сборки и установки.

Следующим седьмым источником Манифеста стало развитие процессного подхода в разработке программного обеспечения, характерного для любого зрелого промышленного производства. Основными

вехами проникновения процессного подхода в разработку программного обеспечения стали разработка в 1989 г. по заказу правительства Великобритании библиотеки ИТ-услуг *Information Technology Infrastructure Library* и разработка в 1991 г. по заказу Министерства обороны США модели зрелости разработки программного обеспечения *Capability Maturity Model*.

Процессный подход, в свою очередь, основывается на общей теории качества (*TQM*), основными принципами которой являются: стандартизация операций и продуктов, измеримость процесса, зависимость качества продукта от качества процесса, предсказуемость сроков и качества, а также необходимость эмоциональной мотивированной вовлеченности в постоянное улучшение процесса всех заинтересованных участников. Здесь следует обратить внимание на то, что процессный подход в значительной степени становится возможным при стандартизации процесса и продукта. Он невозможен при отсутствии правил, стандартов и типизации. Поэтому понятие типового по объему и сроку выпуска минимально жизнеспособного продукта (*MVP*) стало для *agile* ключевым.

Самое развитое применение общая теория качества и процессное управление в разработке программного обеспечения получили в *Lean* — методологии Мэри Поппендик и Тома Поппендика. В этой методологии были применены к разработке программного обеспечения принципы бережливого производства в виде набора из 22 инструментов (практик).

Восьмым источником, безусловно, следует назвать всеобщую компьютеризацию населения развитых стран за счет массового распространения персональных компьютеров в бытовой и в рабочей сферах. То, что основная масса трудоспособного населения, в том числе связанного с компьютерами только в качестве пользователей, получила значительные компетенции в области ИТ-технологий, сделало возможным их вовлечение в тесную коллективную работу в ИТ-проектах. В настоящее время граница между квалифицированным пользователем, технологом и программистом стала не всегда различимой. Часто квалифицированные пользователи выступают в качестве технологов и даже разработчиков в рамках своей прикладной области, не говоря уже о том, что на протяжении карьеры многие из них неоднократно совершают «трансграничные» переходы из специализации в области информационных технологий в бизнес и обратно.

Девятый источник *agile* — развитие гипертекстовых сред документирования и накопления информации как ответ на невозможность эффективного использования простых текстовых документов с объемом в несколько тысяч страниц.

Современные сложные системы в связи со своим масштабом давно потеряли возможность эффективного документирования в виде плоского текста. О каком подробном документировании автоматизированной системы в части пользовательской и другой эксплуатацион-

ной документации можно говорить, если объем документов превысил бы несколько тысяч страниц, а обновления системы должны выходить с еженедельной или ежедневной периодичностью? Если бы даже эта документация была подготовлена, то не найдется аудитории, которая смогла бы это все прочитать. Поэтому один из пунктов Манифеста *agile* отдельно указывает на снижение ценности традиционного документирования. Документация становится неотъемлемой частью непосредственно автоматизированной системы, обновляется параллельно с ней, и в значительной степени замещается в части администрирования использованием стандартных инструментов управления ИТ-сервисов, а в части пользовательского интерфейса следованием стандартным соглашениям о его элементах — очень немногие мобильные приложения содержат отдельные текстовые описания и в основном полагаются на «интуитивную понятность».

Те же утверждения справедливы для проектной документации. Текстовые проектные документы для крупных систем давно стали рудиментом. Минимальным уровнем развития технологий проектирования сегодня является гипертекстовая среда подготовки требований (например, *Jira Confluence*), а желательным — наличие *CASE*-технологии (от англ. *computer-aided software engineering*), обеспечивающей автоматизированное проектирование программного обеспечения и, в случае встраивания в *DevOps*-цепочку, не только генерацию программного кода и структур данных, но и автоматическую поддержку актуальности описания функционала и конфигурации системы (например, *Enterprise Architect*).

Десятым и, возможно, одним из ключевых источников появления *agile* стало развитие методик управления проектами и их стандартизация. Поэтапное развитие стандартов в области управления проектами, например, *PMI PMBOK*, дает очень хорошее представление о дороге движения к *agile* за счет развития практик, инструментов и компонентов, характерных для *agile*-подхода, включая методы обеспечения эффективной работы команд проекта, управление конфликтами, методы снижения рисков проектов за счет их разбиения на короткие итерации, обеспечение регулярных коммуникаций, планирование «набегающей волной» и многое другое.

Основным выводом из информации о приведенных источниках *agile* является следующее заключение: *agile* — это не «придуманная» методология, а скорее название некоторого этапа (скорее всего, не последнего) развития производства в области информационных технологий, явившегося ответом на рост сложности задач, стоящих перед инженерами, и базирующийся на ряде методов, подходов и инструментов, обеспечивающих использование потенциала вычислительных систем и человеческих отношений для построения новой формы производства, основанной на эффективной гибридной человеко-машинной экосистеме, живущей по особым правилам и имеющей свои этапы развития (рис. 10.4).

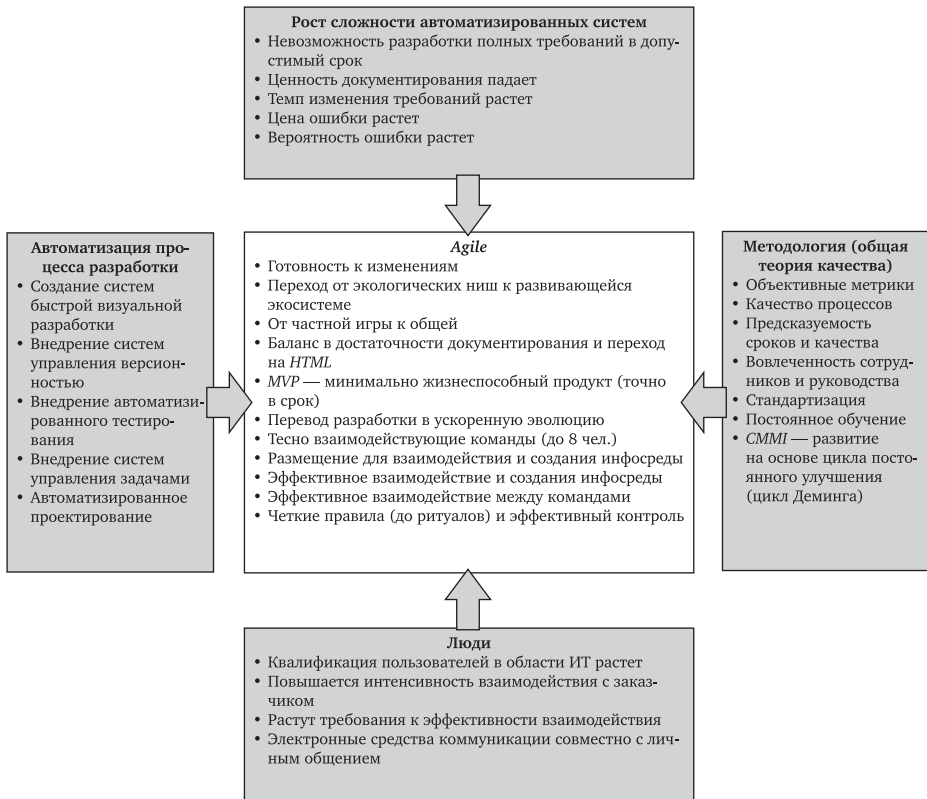


Рис. 10.4. Экосистема agile

Потребность в этой форме производства сформировалась к середине 1980-х гг., а все компоненты, сделавшие ее возможной, оказались готовы к середине 1990-х гг. После достаточно короткого периода утряски и методологических экспериментов в 2001 г. был сформулирован Манифест *agile*.

10.2. Манифест agile

Поскольку к концу XX в. накопилась критическая масса новых методик и инструментов разработки программного обеспечения, 17 основоположников новых методик посчитали необходимым встретиться и обсудить формулировку общих принципов и ценностей, которые характеризуют новое состояние производственного процесса, несмотря на разнообразие конкурирующих между собой подходов.

Это произошло в феврале 2001 г. в США. После двухдневного обсуждения, несмотря на наличие множества противоречий и коммерческую конкуренцию между применяемыми подходами, коллеги смогли выработать четыре основные ценности (рис. 10.5) и 12 принципов *agile* (табл. 10.1).

Мы постоянно открываем для себя более совершенные методы разработки программного обеспечения, занимаясь разработкой непосредственно и помогая в этом другим.

Благодаря проделанной работе мы смогли осознать, что:

Люди и взаимодействие важнее процессов и инструментов, работающий продукт важнее исчерпывающей документации, сотрудничество с заказчиком важнее согласования условий контракта, готовность к изменениям важнее следования первоначальному плану.

Не отрицая важности того, что справа, больше ценим то, что слева.

<i>Kent Beck</i>	<i>James Grenning</i>	<i>Robert C. Martin</i>
<i>Mike Beedle</i>	<i>Jim Highsmith</i>	<i>Steve Mellor</i>
<i>Arie van Bennekum</i>	<i>Andrew Hunt</i>	<i>Ken Schwaber</i>
<i>Alistair Cockburn</i>	<i>Ron Jeffries</i>	<i>Jeff Sutherland</i>
<i>Ward Cunningham</i>	<i>Jon Kern</i>	<i>Dave Thomas</i>
<i>Martin Fowler</i>	<i>Brian Marick</i>	

Рис. 10.5. Манифест agile

Таблица 10.1

Двенадцать принципов agile

Принцип	Пояснения
1. Наивысшим приоритетом для нас является удовлетворение потребностей заказчика благодаря регулярной и ранней поставке ценного программного обеспечения	Принцип является ответом на проблему неопределенности требований и необходимость регулярной выработки стандартного минимально жизнеспособного продукта
2. Изменение требований приветствуется, даже на поздних стадиях разработки. Agile-процессы позволяют использовать изменения для обеспечения заказчику конкурентного преимущества	В условиях ускорения инновационного прогресса и множачихся «подрывных» инноваций требования не могут сохраняться продолжительное время. Соответственно, необходима готовность к изменениям
3. Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев	Как было показано выше, требования на продукт после середины 1980-х гг. не сохраняются более трех месяцев. Если продукт не будет выпущен раньше, то он потребует доделки в связи с изменением требований
4. На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе	Не существует формализованного языка передачи знаний. Следовательно, непосредственное взаимодействие с заказчиком — необходимость

Принцип	Пояснения
5. Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия, обеспечьте поддержку и полностью доверьтесь им	Проектами должны заниматься профессионалы, имеющие необходимый опыт, знания, психологические характеристики (см. гл. 6). Ошибки организационного построения проекта описаны в гл. 9
6. Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды	Правая половинка мозга человека, отвечающая за эмоции, визуализацию и рутинные задачи, требует очного общения для повышения эффективности взаимодействия
7. Работающий продукт — основной показатель прогресса	Только работающий продукт обеспечивает распознавание результата проекта со стороны заказчика (см. гл. 9 в части построения контроля проекта)
8. Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно. <i>Agile</i> помогает наладить такой устойчивый процесс разработки	См. источники <i>agile</i> гл. 10 в части процессного подхода и стандартизации продукта по объему и времени реализации
9. Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта	См. источники <i>agile</i> гл. 10 в части сред быстрой разработки и документирования. Автоматизированные инструменты поддержки цикла разработки, проектирования и документирования — необходимый элемент <i>agile</i>
10. Простота — искусство минимизации лишней работы — крайне необходима	Основа процессного подхода — стандартизация процесса и продукта. См. источники <i>agile</i> гл. 10 в части документирования и <i>MVP</i>
11. Самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд	Эффективное мотивированное, бесконфликтное и эмоционально наполненное взаимодействие команды — необходимый элемент <i>agile</i> -экосистемы
12. Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы	Теория качества в части вовлеченности всех участников в цикл постоянного улучшения процесса — один из источников <i>agile</i> (см. гл. 10)

10.3. Основные методик *agile*

Манифест *agile* разработали специалисты, которые с середины 1980-х гг. занимались созданием и применением на практике новых методов организации производства программного обеспечения и реа-

лизации соответствующих проектов. Ряд методик был оформлен уже после разработки Манифеста *agile* (табл. 10.2).

На рис. 10.6 проиллюстрировано соотношения методик *agile*. Схема разработана на основе приведенной в *Agile Practice Guide*, 2017.

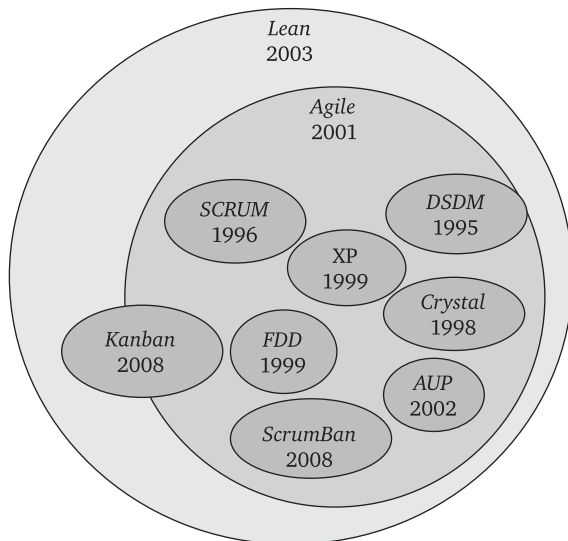


Рис. 10.6. Методики *agile*

Таблица 10.2

Основные характеристики методов *agile*

Метод	Описание	Основные принципы и особенности
1. <i>Lean</i>	Разработан на основе методологии «бережливого производства» <i>Toyota Motors</i> , которая в свою очередь базируется на теории управления качеством и понятии цикла постоянного улучшения	<p>Исключение потерь. Потерями считается все, что не добавляет ценности для потребителя (излишняя функциональность, паузы в процессе, нечеткие требования, бюрократизация, неэффективные коммуникации).</p> <p>Усиленное обучение. Короткие циклы разработки, раннее тестирование, частая обратная связь с заказчиком для создания цикла непрерывного обучения новым подходам и требованиям.</p> <p>Предельно отсроченное принятие решений. Решения, которые могут повлечь критичную для проекта ошибку, следует принимать не на основе предположений и прогнозов, а после открытия существенных фактов.</p> <p>Предельно быстрая доставка заказчику. Короткие итерации с поставкой полезного продукта.</p> <p>Наделение команды властью. Профессионалы должны располагать полномочиями для принятия решений.</p>

Метод	Описание	Основные принципы и особенности
		<p>Встроенная интегрированность. Поддержка целостности и полноты восприятия командой и заказчиком продукта в его жизненном цикле от архитектуры до ценообразования и маркетинга. Поддержка целостной архитектуры с постоянной чисткой технического долга.</p> <p>Широкое видение. Эффективное взаимодействие между участниками проекта на основе разделения единых принципов бережливости: «Мыслить широко, делать мало, ошибаться быстро, учиться стремительно»</p>
2. <i>Kanban</i>	<p>Визуализация этапов производства и промежуточных продуктов для управления производственным циклом.</p> <p>В основе — методология «бережливого производства» <i>Toyota Motors</i></p>	<p>Рекомендует ограничение параллельно выполняемых работ, что позволяет снизить потери на переключение между задачами.</p> <p>Визуализация с помощью «канбан-доски» позволяет быстро выявлять операционные проблемы и стимулирует взаимодействие, направленное на постоянное улучшение.</p> <p>Предполагает эволюционное поступательное развитие существующих процессов для снижения сопротивления, а также фокус на заказчике. Обеспечивает визуализацию рабочего процесса, его улучшений, поддержку и усиление полезных изменений, обучение, снижения воздействия и исключение неэффективных</p>
3. <i>SCRUM</i>	<p>Основывается на работе небольших (до 10 человек) команд, которые разбивают работу на фиксированные по времени и объему итерации (спринты до одного месяца) с ежедневным контролем прогресса*</p>	<p>Закрепляет жесткие шаги процесса и ключевые роли:</p> <ul style="list-style-type: none"> • <i>Product Owner</i> отвечает за повышение ценности продукта для заказчика, определяет продукт в терминах и в интересах заказчика (англ. <i>user stories</i>), обеспечивает включение заказов в бэклог продукта (список заказов), обеспечивает приоритизацию и последовательность реализации заказов; • <i>Scrum Master</i> обеспечивает выполнение процесса <i>SCRUM</i> и его постоянное улучшение, помогает <i>Product Owner</i> в ведении бэклога, борется с препятствиями в ходе проекта, управляет регулярными коммуникациями. <p>Требует регулярной поставки коротких инкрементов полезного продукта (<i>MVP</i>) и совместного расположения команды.</p> <p>Отдает предпочтение рейтинговой оценке сложности продукта (англ. <i>story points</i>) перед оценкой трудоемкости</p>

Метод	Описание	Основные принципы и особенности
4. <i>eXtreme Programming (XP)</i>	Предусматривает доведение «лучших практик» до «экстремального» уровня в целях повышения качества продукта и удовлетворения заказчика	<p>Продукт поставляется короткими циклами (1 неделя).</p> <p>Для максимального использования полезности процедур <i>code review</i> применяется программирование парами (один программирует, другой контролирует), встроенные тесты (англ. <i>unit testing</i>) всего кода, исключение разработки особенностей, пока они действительно не потребовались, плоская управленческая система, простота кода и стандарты написания, ожидание изменений требований со стороны заказчика, регулярная коммуникация с заказчиком</p>
5. <i>Dynamic Systems Development Method (DSDM)</i>	Базируется на RAD и принципе приоритета ограничений	<p>Срок, качество и стоимость проекта задаются жестко, функциональность приоритизируется на основе формальных правил.</p> <p>Принципы:</p> <ul style="list-style-type: none"> • фокусировка на нуждах бизнеса; • поставка вовремя; • эффективное взаимодействие; • никогда не терять качество; • задать основу и наращивать функциональность инкрементами; • разработка итеративная; • открыто и часто коммуницировать; • обеспечивать прозрачный контроль
6. <i>Feature driven development (FDD)</i>	Основывается на объектно-ориентированном подходе	<p>Основной целью является разработка реального, работающего программного обеспечения систематически, в поставленные сроки.</p> <p>Заданы основные роли («хозяин класса» (англ. <i>class owner</i>) и «главный программист» (англ. <i>chief programmer</i>)) и последовательность работ:</p> <ul style="list-style-type: none"> • разработка общей модели; • составление списка необходимых функций системы; • планирование работы над каждой функцией; • проектирование функции; • реализация функции. <p>Первые два процесса относятся к началу проекта. Последние три осуществляются для каждой функции в цикле.</p> <p>Основные практики:</p> <ul style="list-style-type: none"> • объектное моделирование области. Результатом является общий каркас классов объектов, который можно в дальнейшем дополнять функциями; • разработка по функциям. Любая функция, которая слишком сложна для разработки в течение двух недель, разбивается на меньшие подфункции до тех пор, пока каждая подзадача не может быть названа свойством (может быть реализована за 2 недели);

Метод	Описание	Основные принципы и особенности
		<ul style="list-style-type: none"> • индивидуальное владение классом. Каждый индивидуально обособляемый блок кода (класс) закреплен за конкретным владельцем-разработчиком. Владелец ответственен за согласованность, производительность и концептуальную целостность своих классов; • команда по разработке функций. Маленькая, динамически формируемая команда разработчиков, занимающаяся небольшой подзадачей; • проверка кода (англ. <i>inspections</i>). Проверки обеспечивают качество кода за счет оперативного встраивания в процесс; • конфигурационное управление. Обеспечивает идентификацию исходного кода для всех функций и протоколирование изменений; • регулярная сборка. Гарантирует постоянную готовность продукта для предоставления заказчику, обеспечивает поиск ошибок при объединении частей исходного кода на ранних этапах; • обозримость хода работ и результатов. Объективные метрики хода работ на всех уровнях внутри и за пределами проекта
7. <i>Crystal Clear</i>	Семейство «мягких» методологий, масштабируемых в зависимости от масштаба, критичности проекта, численности команды. Приоритет отдан гибкости в применении методов, инструментов и политик**	<p>Принципы:</p> <ul style="list-style-type: none"> • «человекоориентированные» методологии приносят больше пользы, чем «процессоориентированные»; • методология проекта должна быть подобрана индивидуально под проект и команду — нет универсальной методики разработки приложений. <p>Основы:</p> <ul style="list-style-type: none"> • люди — это взаимодействующие существа. Лучшая практика — это персональное общение лицом к лицу с оперативными ответами на вопросы; • люди не могут с одинаковой эффективностью работать непрерывно; • люди очень изменчивы в зависимости от места и времени; • люди обычно хотят быть хорошими гражданами, восприимчивы к среде, готовы к инициативе и ответственны за проект. <p>Общие черты методологий <i>Crystal</i> (высокий уровень присутствия увеличивает шансы проекта на успех):</p> <ul style="list-style-type: none"> • частая поставка. Регулярная инкрементальная поставка полезного продукта (периодичность от одной недели до квартала);

Метод	Описание	Основные принципы и особенности
		<ul style="list-style-type: none"> • чувствительное улучшение. Вовлеченность разработчиков в постоянное улучшение процесса на основе опыта поставок. Предусматриваются периодические встречи для анализа опыта (англ. <i>reflective workshop</i>); • тесная коммуникация. Команда должна располагаться в едином помещении для исключения потерь на взаимодействие и совместного использования информации; • персональная безопасность. В команде должна поддерживаться атмосфера доверия, бесконфликтности и взаимной поддержки; • фокус. Приоритетная фокусировка на индивидуальных задачах, затем на целях проекта. Обеспечение для разработчика двухчасового периода, когда его не должны отвлекать, и двухдневного — для переключения на другой проект; • для обеспечения фокусировки на целях проекта должны предоставляться понятные и обоснованные тезисы и определения целей. Руководитель проекта должен четко приоритезировать цели для фокусировки на отдельных областях деятельности; • упрощенный доступ к экспертам. В доступе разработчика всегда должны быть реальные пользователи продукта, готовые немедленно отвечать на вопросы (минимально — раз в неделю двухчасовая встреча); • техническая инфраструктура с автоматизированным тестированием, управлением конфигурациями и возможностью непрерывной сборки. Непрерывный выпуск версий с автоматическим тестированием и сборкой обеспечивает раннее обнаружение ошибок и проблем
8. <i>Agile Unified Process (AUP)</i>	<p>Основывается на <i>Rational Unified Process</i>, объектно-ориентированном подходе и <i>UML</i>. Предусматривает два вида итерационных поставок продукта: тестирование (демонстрация) и продуктивная</p>	<p>Требует выполнения итерационных коротких поставок с обязательными «дисциплинами» (этапами):</p> <ul style="list-style-type: none"> • модель. Понимание бизнес-целей, проблем и выработка обоснованных решений; • имплементация. Преобразование модели в исполняемый код, доступный для тестирования (например, <i>unit-тест</i>); • тестирование. Обеспечение объективного тестирования, включая обнаружение ошибок и подтверждение соответствия требованиям; • установка. Планирование установок и обеспечение доступности продукта для пользователей; • управление конфигурацией. Управление доступом к компонентам продукта и контроль их версииности;

Метод	Описание	Основные принципы и особенности
		<ul style="list-style-type: none"> • управление проектом. Планирование и контроль активностей в проекте, включая управление рисками, управление задачами людей, взаимодействие с внешней средой проекта, для обеспечения соблюдения сроков и бюджета; • рабочая среда. Обеспечение необходимого процесса, сервисов, инструкций, инструментов и ресурсов для команды. <p>Принципы:</p> <ul style="list-style-type: none"> • команда должна осознавать, что она делает. Недостаточно иметь документацию, необходима дополнительная высокоуровневая коммуникация и обучение; • простота. Документация должна быть настолько лаконичной, насколько это возможно; • гибкость. Соблюдение принципов Манифеста <i>agile</i>; • фокусировка на высокоуровневых задачах. Не каждое мероприятие проекта стоит фокусировки; • независимость от инструментария. Может быть использован любой набор инструментов, если это обоснованно, но он должен быть максимально простым и стандартным
9. <i>ScrumBan</i>	Комбинация <i>SCRUM</i> и <i>Kanban</i>	Заимствует основные роли и компоненты <i>SCRUM</i> , но предусматривает визуализацию в стиле <i>Kanban</i> и возможность отказа от спринт-периодов с переходом на непрерывную работу над эпизодически возникающими задачами. Допускает территориально распределенные команды с обеспечением совместной работы с использованием специальных автоматизированных систем управления задачами, например, <i>Jira</i>

* Сазерленд Дж. Scrum. Революционный метод управления проектами. М. : Манн, Иванов и Фербер, 2016.

** Коберн А. Быстрая разработка программного обеспечения. М. : Лори, 2016.

10.4. Место *agile* и его ограничения

Обычно нет необходимости следовать в ходе всей реализации проекта какому-либо одному подходу *agile* или *waterfall*. Применяемый подход зависит от этапа проекта, привлекаемой команды, инструментария, целей проекта, уровня зрелости организации и многих других факторов.

В плане проекта часто присутствуют участки, которые характеризуются четкими целями, предсказуемостью и необходимостью высокого уровня согласованности с взаимодействующими проектами и подразделениями, например, создание аппаратной и сетевой инфраструктуры будущей системы. Эти участки плана проекта могут реализовываться с использованием *waterfall*-подхода. Учитывая плановый и предсказуемый характер этого подхода, в *Agile Practice Guide* он назван *predictive*.

Часто встречаются участки плана или содержания проекта, требования по которым сформулированы нечетко. Такие участки обоснованно рекомендуются *agile* или менее радикальному инкрементальному подходу.

Промежуточным по характеристикам является итерационный подход, при котором обеспечивается единая продуктовая поставка, но с итерационно повторяющимися в случае необходимости промежуточными фазами анализа (проектирования) и разработки (тестирования).

В целом корректным является представление о предиктивном подходе как об экстремальной реализации итерационного, а *agile* — как об экстремальной реализации инкрементального подхода, предусматривающего регулярную поставку полезного продукта. На рис. 10.7 приведены схемы жизненного цикла продукта, реализуемого каждым из подходов.

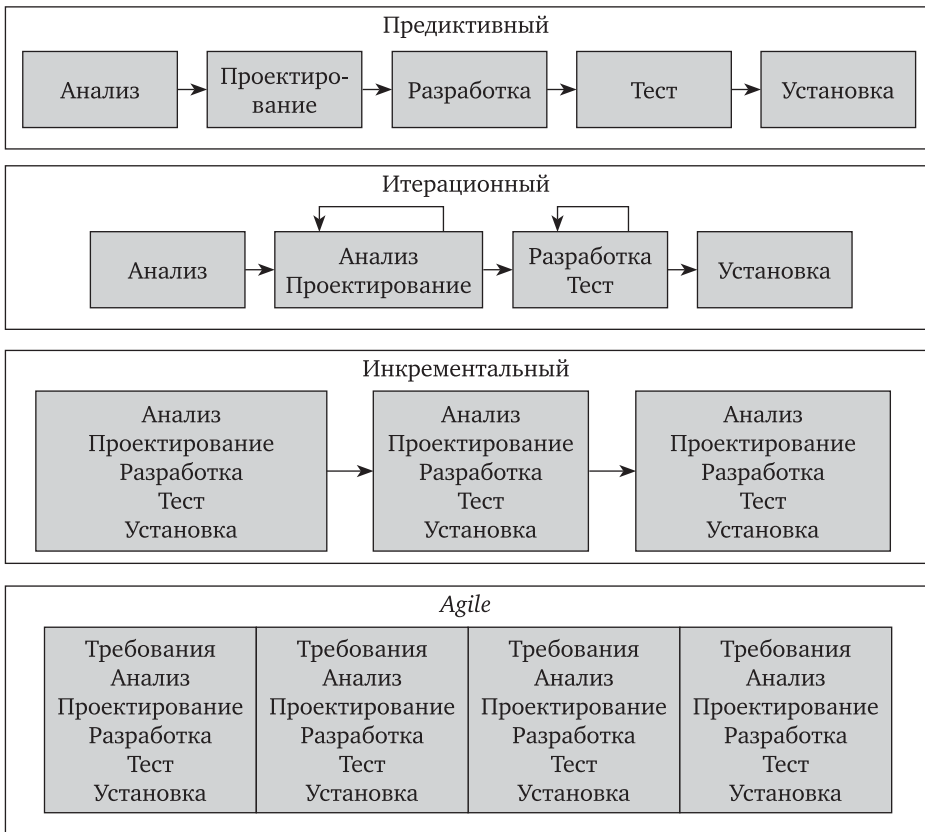


Рис. 10.7. Жизненный цикл проекта при различных подходах

На первый взгляд схожие, инкрементальный и *agile*-подходы отличаются тем, что в *agile* чаще используются равные по времени циклы. Циклы в инкрементальном подходе могут отличаться по объему функциональности и срокам реализации.

Учитывая возможность параллельного выполнения разнородных задач в проекте, одновременно могут применяться два разных подхода параллельно работающими командами (рис. 10.8).

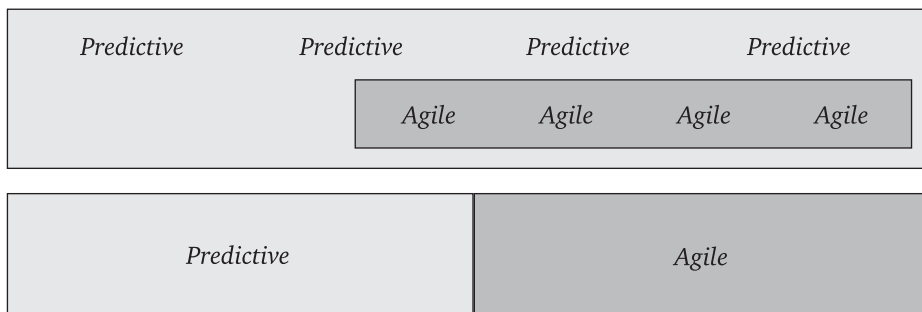


Рис. 10.8. Наиболее применяемые варианты гибридного подхода

10.5. Минимально жизнеспособный продукт (MVP)

Практически все методы, реализующие гибкий поход, базируются на понятии инкрементального продукта, поставляемого в ограниченный период времени, который обычно составляет не более одного месяца. С учетом того, что рекомендуемая численность команд составляет обычно до 10 человек, функциональный объем такого инкрементального продукта жестко ограничен.

Одной из самых больших проблем при внедрении *agile* на практике является то, что, несмотря на готовность ИТ-коллективов к восприятию гибких методик, любые усилия будут бесполезны, если заказчики не проявят готовности к формулировке требований инкрементально и не согласятся на эволюционное поступательное развитие продукта с регулярной поставкой частичного функционала. Более того, чаще всего заказчики всячески этому препятствуют, стараясь «не отпустить» команду разработчиков до полного удовлетворения своих постоянно изменяющихся и возрастающих требований. В таких случаях *agile* быстро вырождается в итерационный процесс с непрерывной доработкой (тестированием) без поставки полезного продукта.

Ситуация значительно осложняется, если команда разработчиков разделяется между большим количеством заказчиков, что является обычной практикой.

Декомпозиция систем на компоненты является одним из базовых умений инженера, но разбиение полезного продукта на инкрементальные продукты, которые представляют индивидуальную ценность для заказчика и могут быть полезно использованы, — задача гораздо более

сложная, и она становится нерешаемой при отсутствии содействия со стороны заказчика.

Хорошую аналогию представляет развитие сетевых технологий. На начальном этапе были созданы сети коммутации каналов (например, аналоговые телефонные сети). При установке соединения канал оказывался занят для других потенциальных пользователей на все время «разговора». Даже при значительном резервировании междугородних линий такие сети не обеспечивали возможности доступности «по требованию» и постоянно оказывались заняты. Их общая пропускная способность была в целом низкой. Эта модель характерна для предиктивного и итерационного методов.

На следующем этапе появились сети коммутации сообщений. Они позволяли освобождать канал после передачи отдельных сообщений. Ситуация несколько улучшилась, но общая пропускная способность осталась невысокой в связи с тем, что длина сообщений варьировалась в достаточно больших пределах и свободные «окна» не были предсказуемы или возможны для использования конкурирующими пользователями. При этом неиспользованных «окон» оказывалось достаточно много. Такая модель сходна с инкрементальным подходом.

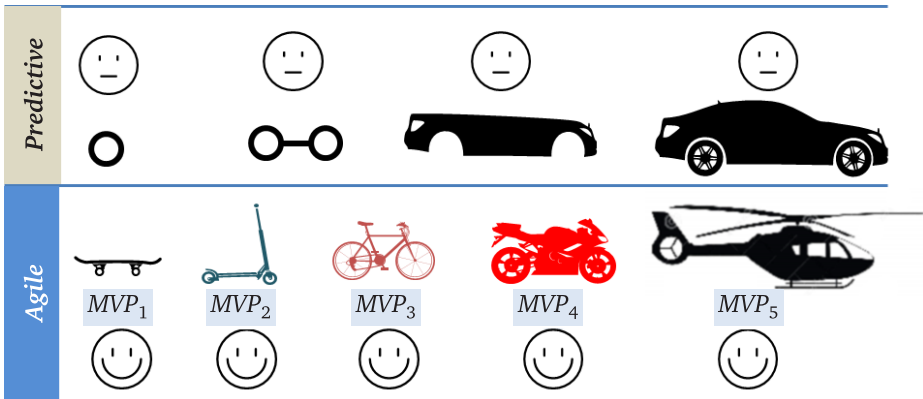


Рис. 10.9. Концепция MVP

Полностью поменял ситуацию с пропускной способностью переход на технологию с коммутацией пакетов. Пакеты, стандартные по длине, управляются с высокой скоростью, на основе стандартных алгоритмов, могут перенаправляться по наиболее выгодному пути и т. д. Сегодня мы сталкиваемся с ситуацией занятости канала связи только в случае занятости абонента. Проблема потери производительности сети из-за неэффективного управления ушла в прошлое. Схема коммутации пакетов близка технологии agile, и может быть реализована только для MVP. Именно MVP обеспечивает реальную гибкость в управлении «пакетами» разработки продукта. На рис. 10.9 представлена доработанная по известной иллюстрации Boston Consulting Group концепция MVP.

10.6. Определение трудоемкости

С понятием *MVP* тесно связана задача определения трудоемкости задачи и проекта в целом.

Существует достаточно большое количество методов определения трудоемкости и стоимости при решении задач по разработке программного обеспечения. Одним из таких методов является метод функциональных точек (англ. *function points*).

Метод был разработан в 1979 г. для оценки продуктивности разработчиков в компании *IBM*, базируется на подсчете разнородных требований (связанных с бизнес-функцией) к программному обеспечению по категориям: выходы (формы отображения, сообщения об ошибках и т. д.), запросы (запросы пользователя с немедленным ответом на экран), входы (формы ввода), внутренние файлы (блоки данных, связанные с бизнес-функциями), внешние интерфейсы (интеграционные интерфейсы с внешними системами).

Все подсчитываемые функции экспертно взвешиваются по сложности (умножаются на коэффициент). Затем производится подсчет суммы функциональных точек. Сумма может быть дополнительно умножена на коэффициент, связанный со сложностью среды разработки, сложностью алгоритмов и требованиями к надежности.

Пока не существует формального стандартизированного метода определения коэффициентов сложности. Их приходится определять на практике статистически для схожих программных систем.

Невозможность однозначного объективного расчета трудоемкости и принципиальная изменчивость требований значительно снижает потенциал всех формализованных методов расчета трудоемкости и выводит на первый план методы экспертного оценивания (с использованием опыта реализации аналогичных проектов). Большинство *agile*-методик, учитывая данный вывод, предлагают полностью отказаться от оценивания трудоемкости задач в часах либо деньгах и перейти на разновидности рейтинговой оценки в абстрактных единицах, например, в «собаках» или «покерных пунктах» (*SCRUM*). Такие единицы позволяют ориентировать эксперта сразу на оценку рейтинга сложности, минуя фазу формального подсчета распознаваемых особенностей продукта. Это часто бывает более эффективно и быстро, чем сначала подсчитывать «функциональные точки», затем дискутировать с исполнителем на тему определения «функциональной точки», а в финале подгонять получившуюся сумму под интуитивную экспертную оценку за счет субъективно определяемых весовых коэффициентов. Для более полного понимания вопросов, связанных с получением объективных оценок на основе субъективного мнения экспертов предлагаем самостоятельно изучить работу А. Орлова «Организационно-экономическое моделирование. Экспертные оценки»¹.

¹ Орлов А. Организационно-экономическое моделирование. Экспертные оценки. М. : Изд-во МГТУ им. Баумана, 2011.

Проверочные тесты к разделу II

1. Проект — это:

- а) многократно повторяемый набор операций по созданию уникального или нового продукта;
- б) не ограниченная по времени деятельность по созданию нового или уникального продукта;
- в) ограниченная по времени деятельность по созданию уникального или нового продукта, результата или услуги.

2. В настоящее время доля успешных проектов по международной статистике (*The Standish Group International*):

- а) превышает 50 %;
- б) ниже 30 %.

3. Основным инструментом промежуточного контроля исполнения проекта является:

- а) периодическая отчетность руководителя проекта;
- б) календарный план проекта с предусмотренными на каждом этапе документами, подтверждающими исполнение данного этапа.

4. Выберите корректно сформулированную цель проекта автоматизации:

- а) обеспечить внедрение новой системы автоматизации;
- б) модернизация автоматизированной системы контроля лимитов для повышения производительности до 1000 транзакций в минуту;
- в) провести модернизацию автоматизированной системы предприятия для обеспечения повышенной конкурентоспособности и масштабируемости.

5. Утверждаемый календарный план проекта должен содержать информацию:

- а) о промежуточных целях проекта и достигаемых на различных этапах функциональных возможностях системы без предоставления отчетных документов и других результатов работ;
- б) о последовательности разработки проектных документов, функциональных этапах, измеримых поддающихся проверке результатах и отчетных документах.

6. При инициировании проекта основное действие:

- а) определение и согласование основной цели или набора целей проекта с приоритетностью, определение состава участников проекта;
- б) составление плана коммуникаций в рамках проекта;
- в) определение состава работ проекта.

7. Утверждаемый календарный план проекта (базовый) может быть изменен:

- а) руководителем проекта;
- б) заказчиком проекта;
- в) спонсором проекта;
- г) всеми вышеперечисленными совместно.

8. Основной причиной возникновения конфликтов при выполнении проектов является:

- а) недостатки воспитания участников проектной группы и сотрудников организации;
- б) недостаточная мотивация участников проектной группы;
- в) отсутствие лидерских навыков руководителя проекта;
- г) процессы изменений, инициированные проектом в организации, и естественное сопротивление взаимодействующих подразделений изменениям.

9. Наиболее эффективное управление конфликтами в проектной деятельности предполагает:

- а) избегание конфликта;
- б) настаивание на своей точке зрения;
- в) компромисс;
- г) поиск решения;
- д) полное принятие точки зрения контрагента.

10. Наиболее эффективная длительность этапов планирования либо периодичность появления результатов в контрольном (базовом) плане:

- а) 1 день;
- б) 1 неделя;
- в) 1 месяц;
- г) 1 год.

11. Наиболее эффективная длительность планирования этапов работ в рабочем плане:

- а) 1 день;
- б) 2 недели;
- в) 2 месяца.

12. Участники проекта — это:

- а) участники рабочей группы;
- б) проектная группа, подразделения, заинтересованные в реализации проекта, спонсор и заказчик;
- в) проектная группа и подразделения, заинтересованные в реализации проекта.

13. Полное отсутствие противоречий и конфликтов при выполнении проектов свидетельствует:

- а) о недостаточной интенсивности работ по проекту;
- б) о высокой эффективности работ по проекту.

14. Метод agile:

- а) подразумевает обеспечение тесного взаимодействия разработчика с заказчиком в едином рабочем пространстве до полного удовлетворения требований к системе без документирования;
- б) подразумевает обеспечение ускоренной эволюции минимально жизнеспособного программного продукта на основе эффективного процесса производства ПО;
- в) не учитывает неспособность полностью определить детальные требования к системе на начальном этапе проектирования.

15. MVP — это:

- а) минимальный жизнеспособный (полезный) продукт, который может быть реализован в стандартный срок;
- б) максимальный полезный продукт, закладываемый в требования на старте проекта и предусматривающий все возможные опции использования в дальнейшем;
- в) минимальный результат достижимый в спринт-период.

Раздел III

УПРАВЛЕНИЕ ИТ-ПРОЦЕССАМИ

В результате изучения данного раздела студент должен:

знать

- теоретические основы процессной деятельности;
- типовые ошибки проектного управления;
- основные положения базовых стандартов в области проектного управления;

уметь

- оценивать уровень зрелости процессов и подразделений;
- разрабатывать и согласовывать основные документы организации управления ИТ-службой;
- управлять рисками ИТ;

владеть

- методикой анализа рисков.
-

Глава 11

ТИПОВЫЕ ОШИБКИ ПРОЦЕССНОГО УПРАВЛЕНИЯ

Система организации состоит из людей, используемых ими инструментов и оборудования, методов и процедур работы. Эти составляющие поддаются замене — специалист может занять вакансию и покинуть ее, оборудование замещается в связи с износом или устареванием, отдельные процедуры и методы обработки продукта или предоставления услуги могут совершенствоваться, замещаться или исключаться. При этом функционирование организации, предоставление услуг и выпуск продуктов с ее стороны сохраняются за счет поддержания компоненты, которая является основополагающей и объединяющей остальные составляющие — процесса. Вполне корректно противопоставлять понятие процесса и проекта в управлении информационными технологиями. Проект — это однократно реализуемый набор действий для создания нового результата. Процесс — многократно повторяемый набор действий.

В данной главе приводятся наиболее критичные и часто встречающиеся ошибки построения процессов в управлении информационными технологиями. Список далеко не исчерпывающий.

11.1. Ориентация исключительно на результат с недооценкой контрольных функций

Теоретики качества достаточно емко сформулировали идею постоянного улучшения на основе четырех основных этапов (англ. *plan, do, check, act*), однако на практике цикл Деминга — Шухарта реализуется крайне редко.

На рис. 11.1 представлена рекомендуемая и обычная ошибочная реализация производственного цикла (интерпретация предложена Яном Грэхемом).

Данная ошибка характерна для низкого уровня развития производственных процессов и является следствием ориентации исключительно на объем «продуктивного» результата с игнорированием необходимости «непроизводительных» работ по планированию, контролю и анализу.

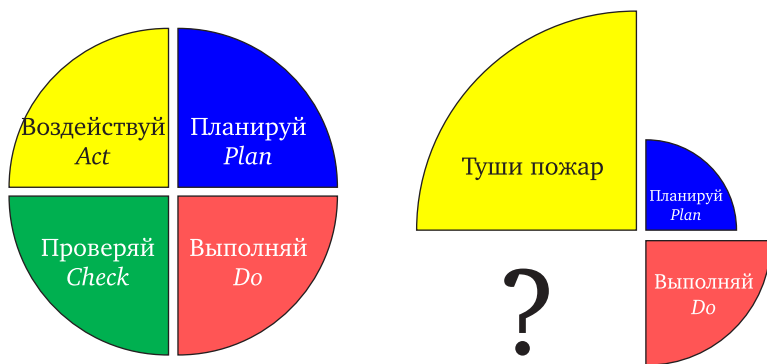


Рис. 11.1. Рекомендованная и обычно реализуемая на практике последовательность шагов производственного процесса

Например, в области разработки программного обеспечения обычно ошибочно считается, что разработчик, поставляющий программный код, является продуктивным звеном, в то время как тестировщик, планировщик версий и последовательности реализации очереди задач, поставленных пользователями, — некие довески к процессу, едва ли не помехи.

Для незрелых и неэффективных производственных процессов характерно отсутствие контрольных точек, анализа недостатков и планирования работ.

Большинство организаций ориентируются на полезный результат и часто оказывают огромное давление на производственный процесс с точки зрения ориентации всех ресурсов развития на рост выработки, что приводит к деформации производственных процессов и подавлению роста их эффективности.

При этом Грэхем отмечает, что, несмотря на ошибочность подхода, «тушение пожаров» — весьма уважаемый род деятельности. На этом занятии многие люди делают карьеру. При этом часто забывается, что их основная задача — эти пожары не допускать. На практике случается, что «пожарники» сознательно провоцируют пожары, оказывая дополнительное давление на производственные подразделения и «новаторов» с целью подавления развития контрольных и плановых функций¹.

11.2. Совмещение разработки и администрирования автоматизированных систем

На практике часто встречается построение процессов разработки и эксплуатации прикладных систем, при котором функции, связанные с поддержкой их функционирования и обслуживания пользователей,

¹ Нив Г. Организация как система. Принципы построения устойчивого бизнеса Эдвардса Деминга. М. : Альпина Паблишер, 1990.

совмещаются одним или группой сотрудников с функциями развития этих систем¹. То есть одни и те же сотрудники заняты выпуском обновлений к программам (их доработкой) и администрированием промышленно эксплуатируемых комплексов — установкой обновлений, настройкой прав доступа, устранением инцидентов и т. д.

Данный подход не является допустимым как с точки зрения современных стандартов управления информационными технологиями, так и с точки зрения здравого смысла.

Во-первых, при совмещении сотрудник вынужден постоянно переключаться между разнородными задачами по написанию целостного, часто очень объемного программного кода крупной системы и ответами на запросы пользователей, число которых в крупных организациях измеряется тысячами.

Если вы не программировали и для вас суть проблемы не очень ясна, то представьте ситуацию, при которой вы пишете многостраничный документ, а вас периодически отрывают от этого занятия звонками по темам, которыми вы занимались месяцы и годы назад.

Как минимум, данная модель создает чрезмерную нагрузку на сотрудника, которая не компенсируется никаким профессионализмом и умением управлять рабочим временем. Проведенные на практике замеры занятости разнородными задачами программистов-администраторов крупных систем показывают распределение загрузки разработка/администрирование в среднем как 40/60 %. При этом приведенные 60 % распределены по времени в течение дня. Таким образом, программисты заняты в течение дня разработкой программ короткими порциями, многократно переключаясь между задачами и тратя суммарно значительное время на восстановление внутреннего видения структуры и текущего состояния поставленной задачи разработки после очередных прерываний. Производительность разработчика при такой работе снижается не до 40 % от рабочего времени, а едва ли не до 0. Представьте реальную ситуацию, при которой неизбежный рост функциональности крупной системы приводит к постоянному росту количества пользователей, сбоев и задач по администрированию. Функция доработки такой системы неизбежно коллапсирует со временем.

Крайняя нежелательность данного режима для разработки программ подробно обсуждается, например, в работе Коберна². К слову, данный автор является одним из основателей *agile* и создателем Манифеста *agile*.

Во-вторых, при совмещении функций у одного сотрудника вы лишены возможности дублирования компетенций по поддержке системы. Персональная зависимость, а с ней и персональный риск, многократно возрастают. Организация становится заложником отдельных сотрудников, что не является допустимым.

¹ Здесь мы не обсуждаем случай совмещения данных функций в рамках одного подразделения, когда их совмещение у одних сотрудников отсутствует.

² Коберн А. Быстрая разработка программного обеспечения.

В-третьих, если функции создания новых версий программ и их установки на промышленный контур совмещены у одного сотрудника, то в процессе установки отсутствует необходимый разрыв между операциями, который позволяет сделать обязательным контроль качества устанавливаемого обновления.

Тестирование остается на доброй воле сотрудника и рано или поздно под влиянием давления недовольных пользователей в случае критичного сбоя или в случае частичной потери контроля, например, в результате болезни, будет установлено некачественное обновление. При этом вы, как руководитель, будете лишены возможности расследовать причины сбоя. Некачественное обновление будет так же бесконтрольно устранено, как было установлено. На практике подобные ситуации встречаются достаточно часто, и даже применение систем управления версионностью или развертыванием не гарантирует от их возникновения.

В-четвертых, поддержка и администрирование промышленных критичных систем имеет своей основной целью обеспечение их надежного функционирования. То есть одним из основных ключевых показателей эффективности (англ. *key performance indicators, KPI*) сопровождающего администратора является надежность, которая может быть выражена, например, в совокупном времени недоступности системы для пользователей за период.

Основным показателем эффективности работы программиста-разработчика является скорость, с которой он выполняет заказы на развитие системы, и количество выполненных заказов на доработку.

Любое изменение промышленной системы несет риск сбоя, связанный с недостатками тестирования, который тем выше, чем менее эффективно оно выполняется.

Совмещая функции администратора и разработчика, мы получаем сотрудника, имеющего конфликтующие показатели эффективности. С одной стороны, он должен как можно чаще устанавливать обновления, с другой — должен сопротивляться этим обновлениям. При этом руководитель лишен возможности спросить с него повышение эффективности как в части ускорения разработки, так и в части роста надежности. Либо одна из функций будет страдать, либо он будет выполнять неудовлетворительно обе.

И наконец, люди различаются своими психологическими характеристиками. В литературе встречается деление психотипов на спринтеров, стайеров и марафонцев. Первый тип людей склонен за счет кратковременного усилия добиваться значительных результатов и перемежать работу обоснованными периодами отдыха. Второй тип склонен к долговременной рутинной деятельности. Совмещая администрирование и разработку, мы заставляем один тип (чаще это оказывается спринтер) совмещать рутину и творчество. Заставляя спринтера заниматься рутинной работой, мы лишаем его заслуженного отдыха, что неизбежно приведет к потере ценного сотрудника. Не говоря уже о том, что творческий

администратор критичной системы — это ночной кошмар любого руководителя; от него постоянно ожидают неприятного сюрприза, организованного из лучших побуждений. Он забывает народную мудрость, выраженную в поговорке «работающую систему не трогай».

Заставляя марафонца заниматься программированием, мы лишаемся возможности получать программы, основанные на новых подходах, современных инструментах и шаблонах.

Глава 12

ЗРЕЛОСТЬ ИТ-ПРОЦЕССОВ ПО МОДЕЛИ СММИ

Если сотрудники постоянно заняты «спасением» организации, если они систематически не могут уйти вовремя домой с работы и постоянно приходится премировать отдельных «героев» за сверхнормативные усилия, то, скорее всего, организация находится на низком уровне зрелости процессов¹.

Повторим несколько принципов теории качества, которые были рассмотрены в гл. 4:

1) качество предоставляемых ИТ-службой сервисов определяется качеством процессов этой службы;

2) невозможно предоставлять качественный ИТ-сервис и качественные автоматизированные продукты, имея некачественные неупорядоченные процессы;

3) не может быть улучшено то, что не измерено;

4) систему управления качеством нельзя выстроить за один шаг — это результат длительного эволюционного процесса, основанного на цикле постоянного улучшения.

На пересечении развития теории качества и информационных технологий родился ряд международных стандартов (например, *ISO 20000 — Information Technology Service Management, Control Objectives for Information and Related Technologies*), наборов рекомендаций (например, *Information Technology Infrastructure Library*) и модель зрелости (*Capability Maturity Model Integration*), которая включила не только описание того, как необходимо выстраивать процессы, но и в какой последовательности, и как оценить текущее состояние.

12.1. История и цели создания СММИ

В середине 1980-х гг. по инициативе Министерства обороны США была создана модель зрелости организации — *Capability Maturity Model*. Непосредственно работы по созданию модели выполнял *Software Engineering Institute (SEI)*, подразделение *Carnegie Mellon University*.

Поводом для начала работ над моделью стал ряд провалов крупных военных проектов, связанных с привлечением разработчиков программного обеспечения.

¹ *Norfolk D. Managing capability and maturity. L. : Bloor Research, 2006.*

Первичной целью создания модели было обеспечить предсказуемый уровень качества программного обеспечения, получаемого от сторонних разработчиков путем их ранжирования по уровню качества и эффективности внутренних процессов производства.

В основу разработки был положен базовый постулат теории качества: «Качественный продукт может производиться только при условии существования качественных производственных процессов».

На первом этапе создания модели зрелости использовалась работа Филипа Кросби «Качество — бесплатно»¹ с приведенной в ней матрицей зрелости организаций — *QMMG* (от англ. *Quality Management Maturity Grid* — матрица зрелости управления качеством): *uncertainty* (неопределенность), *awakening* (эпизодичность), *enlightment* (просвещение), *wisdom* (мудрость), *certainty* (определенность). В ноябре 2011 г. была выпущена очередная полная версия модели — *1.3 CMMI*, которая содержит другое наименование уровней:

- *initial* — начальный;
- *managed* — управляемый;
- *defined* — регламентированный;
- *quantitatively managed* — количественно управляемый;
- *optimizing* — оптимизирования.

После выхода нескольких последовательных уточняющих версий документа к 1991 г. модель получает широкое распространение в мире (*CMM v1.1*).

К 1997 г. модель получила развитие в виде ряда версий для проектирования систем (*SE-CMMI*), управления человеческими ресурсами (*PM-CMM*), оценки программного обеспечения (*SA-CMM*), конструирования/инжиниринга (*EMM*). Параллельно шло развитие международных стандартов *ISO* в области качества процессов управления, в том числе *ISO 9001 (Quality system requirements)*, *ISO15504 (Software Process Improvement and Capability Determination SPICE)* и др.

Результатом работы по обновлению версии *CMM* в 1997 г. стала объединенная на сменумодель — *CMM Integration*, учитывающая разработки в смежных областях. В настоящее время актуальной является версия *CMMI v1.3*, вышедшая в ноябре 2011 г. Однако ей на смену уже приходит новая версия 2.0. В рамках этого обновления уже выпущен раздел *Development* (разработка). Полное обновление модели, включая возможность прохождения официальной оценки, запланировано на 2019 г. На рис. 12.1—12.2 представлена статистика роста применимости модели в мире².

Применение *CMMI* в Российской Федерации по состоянию на 2018 г. представлено в табл. 12.1.

¹ Crosby P. B. *Quality Is Free: The Art of Making Quality Certain*. N. Y. : Penguin Books, 1980.

² URL: <https://cmminstitute.com/resource-files/public/quality/maturity-profiles>.



Рис. 12.1. Применение СММИ по континентам на 2017 г.

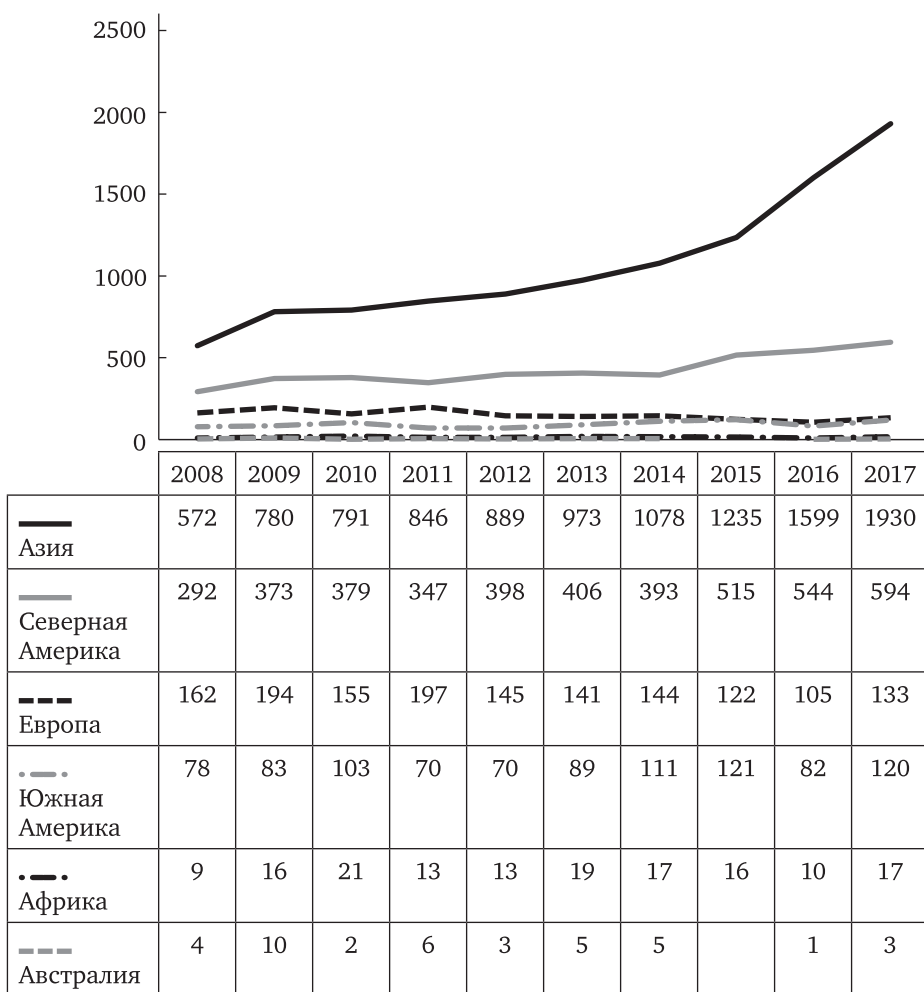


Рис. 12.2. Динамика роста применения СММИ по континентам

Применение СММИ в Российской Федерации¹

Компания	Уровень
<i>Accenture Tver Delivery Center application maintenance capability</i>	3
<i>Atos IT Solutions and Services LLC (RUSSIA) Consulting and System Integration</i>	3
<i>Luxoft Aerospace Delivery Center</i>	5
<i>National Settlement Depository (Russia) IT Division</i>	3
<i>NetCracker Technology Corporation Solution Delivery Department</i>	5
<i>Sberbank of Russia Sberbank — Technology Core Platform Development Center</i>	3
<i>Sberbank of Russia Sberbank — Technology Credit Factory Development Center</i>	3

Глобально в настоящее время для применения СММИ характерны следующие тенденции и параметры:

- с 2007 г. зарегистрировано 14 тыс. официальных оценок;
- в среднем количество оценок растет на 9 % ежегодно;
- 76 % организаций, прошедших оценку, имеют до 100 сотрудников;
- только 10 % организаций достигли высших уровней (4-го или 5-го);
- количество оцененных организаций, применяющих подходы *agile*, выросло с 30 % до 80 % за период с 2009 по 2017 г.;
- количество организаций, проходящих оценку периодически, выросло до 73 %;
- количество оценок выросло на 18 % в 2017
- наибольший рост количества оценок: Китай, США, Индия и Мексика.

12.2. Принципы и результативность применения СММИ

Рост зрелости организации предполагает возрастание качества услуг или производимых продуктов, снижение рисков и персональной зависимости, снижение внутренних и внешних конфликтов, а также переход от управления проектами к практике управления процессами.

На основе первичных описательных характеристик уровней модель предоставляет относительно простой инструмент для экспресс-оценки текущего состояния организации и выработки первичных рекомендаций по формированию стратегии развития.

Эволюционность, заложенная в СММИ, означает невозможность и нежелательность революционных преобразований. Модель позволяет

¹ <https://sas.cmminstitute.com/pars/pars.aspx>

обоснованно ставить стратегические цели развития и обеспечивать их достижимость.

Выгоды от знания и использования подходов *СММИ* на практике обеспечиваются эффективностью их применения для решения ряда типовых задач, включая принятие решения о возможности и результативности реализации сложных проектов в конкретной организации. Чем ниже уровень зрелости, тем более рискованным является инвестирование в крупные интеграционные проекты.

СММИ создавалась в качестве модели зрелости процессов разработки программного обеспечения, но в настоящее время ее применение данной областью не ограничивается. По всему миру данная модель находит все более широкое применение в качестве универсальной модели развития любых организационных процессов и хозяйственных форм.

Имеющаяся международная статистика демонстрирует преимущественное применение модели зрелости в компаниях, специализирующихся на предоставлении услуг (более 70 %).

В табл. 12.2 приведена официальная статистика результативности применения модели зрелости по данным 35 компаний (на 2006 г.). К настоящему времени не проводилось повторного аналогичного исследования.

Таблица 12.2

Результат применения *СММИ* по данным 35 компаний (СМУ/SEI-2006-TR-004)

Показатель деятельности	Среднее улучшение
Себестоимость	34 %
Соблюдение плановых сроков	50 %
Производительность	61 %
Качество	48 %
Удовлетворенность клиентов	14 %
Возврат инвестиций	4:1

Модель уделяет значительное внимание явлению «конфликта», проявляющемуся на различных этапах реализации проектов и выполнения процессов. Практически в каждой из приводимых в тексте *СММИ* процессных областей содержатся требования налаживать эффективное разрешение различного рода конфликтов и предотвращать их на основе излагаемых рекомендаций и требований. Конфликт рассматривается как неизбежный, часто — полезный, требующий эффективного управления.

Этот подход полностью соответствует положениям теории сложных систем, где конфликт рассматривается как естественный процесс взаимодействия систем в рамках процесса развития и взаимной адаптации, а также рассмотренной в гл. 3 экономической теории институционализма.

Наличие, величина и эффективность разрешения конфликтов непосредственно связываются с достижением определенного уровня зрелости. Чем выше уровень конфликтов и менее эффективно они разрешаются, тем ниже уровень зрелости.

Явно присутствуют значимые параллели между развитием экономической научной мысли и подходами, применяемыми в области высоких технологий: и там, и там процесс (или институция) являются базовым объектом, а его зрелость — показателем эффективности хозяйствующего субъекта; обе области значительное внимание уделяют анализу адаптационных процессов и конфликту, характеризующему интенсивность этих процессов.

При этом в отличие от научных разработок экономической теории, аналогичные подходы в области технологий доведены до уровня практического применения и уже успели продемонстрировать свою действенность. Это создает потенциал для дальнейшего развития и корректировки теории институционализма с учетом накопленных в данной области фактических данных. В табл. 12.3 приведено сравнение основных особенностей теории институционализма, общей теории качества и СММИ.

Таблица 12.3

Сравнительные особенности теории институционализма, TQM и СММИ

Институционализм	TQM	Модель зрелости СММИ
Институты	Процессы	Институционализация процессов
Уровень развития	Качество	Уровень зрелости
Оппортунистическое поведение	Лидерство, вовлеченность	Конфликты, персональная зависимость
Постоянная эволюция	Цикл постоянного улучшения	Процессы оптимизируемости
Размер издержек	Принятие решений на основе фактов	Методика оценки (SCAMPI)

Допустим, читатель в качестве нового руководителя в свой первый рабочий день пытается получить «стартовую» информацию о состоянии системы управления в новой для себя организации, или в рамках некоторого процесса обновления инициировать подготовку стратегии ее развития на ближайшую перспективу, или в качестве аудитора получить общую оценку о состоянии процессов управления и информационных технологий в организации. Другими словами — выяснить, «кто виноват и что делать?»

Ответ на этот вопрос обычно звучит следующим образом: «виноваты» процессы и надо их развивать, а для того, чтобы развивать, необ-

ходимо точно оценить текущую позицию организации и выработать стратегию дальнейшего развития.

Наиболее эффективный путь получения оценки, разработанный к настоящему времени, — это использование *СММІ*. На рис. 12.3 представлена сравнительная характеристика различных стандартов, рекомендаций и *СММІ* относительно их потенциального вклада в воображаемую «интегральную методiku» (экспертов попросили представить некоторую суммарную методiku развития процессов, основанную на всех основных «лучших практиках», и спозиционировать вклад каждой).

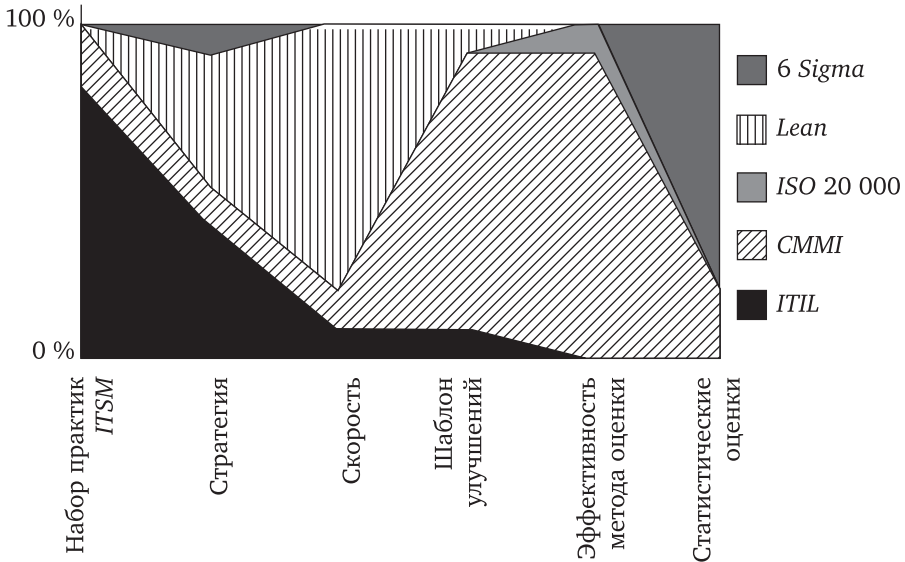


Рис. 12.3. Сравнительная характеристика *СММІ*¹

Из рис. 12.3 видно, что *СММІ* превосходит все остальные «стандарты» в области предоставления точной оценки организации с точки зрения ее «местоположения», а также с точки зрения выработки шаблона развития. При этом *СММІ* не претендует на детализированное представление о том, как именно необходимо реализовывать те или иные процессы. Это прерогатива *ITIL*.

На чем основаны сильные стороны *СММІ*? Эта модель предполагает, что в своем развитии организации проходят несколько состояний «зрелости», четко распознаваемых с помощью формальных методов, изложенных в специальном разделе модели, который в настоящее время называется *Standard CMMI Appraisal Method for Process Improvement*. Общее описание уровней представлено в табл. 12.4.

¹ ITIL®V3.0 Compliance Benchmarking with CMMI SCAMPI, ACMMI®. Technology Conference and User Group. November, 18th, 2009.

Общая характеристика уровней зрелости СММІ

Уровень зрелости	Фокуси- ровка	Необходимые про- цессы	Эффектив- ность, каче- ство, про- дуктивность, управление процессами ↑
5. <i>Optimizing</i> (оптими- зированный). Посту- пательное улучшение процессов	Цикл постое- янного улуч- шения	<i>Causal Analysis and Resolution</i>	
		<i>Organizational Performance Management</i>	
4. <i>Quantitatively managed</i> (количественно управ- ляемый). Процессы и проекты полностью управляются на основе количественных метрик	Метрики и <i>KPI</i>	<i>Organizational Process Performance</i>	
		<i>Quantitative Project Management</i>	
3 <i>Defined</i> (регламен- тированный). Про- цесс регламентиро- ван, контролируется, устойчив к нештатным ситуациям, снабжен необходимым инстру- ментарием	Стандар- тизация и контроль процессов	<i>Integrated Project Management</i>	
		<i>Risk Management</i>	
		<i>Organizational Process Focus</i>	
		<i>Organizational Process Definition</i>	
		<i>Organizational Training</i>	
		<i>Decision Analysis and Resolution</i>	
		<i>Product Integration</i>	
		<i>Validation</i>	
		<i>Verification</i>	
		<i>Requirements Development</i>	
<i>Technical Solution</i>			
2. <i>Managed</i> (управляе- мый). Процесс планиру- ется, регламентирован, мониторится и кон- тролируется на уровне проектов. Регламенты исполняются не полно- стью	Базовое проектное управление	<i>Configuration Management</i>	
		<i>Project Planning</i>	
		<i>Project Monitoring and Control</i>	
		<i>Measurement and Analysis</i>	
		<i>Requirements Management</i>	
		<i>Process and Product Quality Assurance</i>	
		<i>Supplier Agreement Management</i>	

Уровень зрелости	Фокусировка	Необходимые процессы	
1. <i>Initial</i> (начальный). На этом уровне процессы, системы, обязанности персонала не документированы. Деятельность сотрудников является хаотичной и реактивной (под воздействием внешних событий)	Нет процессов		Риски, низкое качество, персональная зависимость, конфликты, управление проектами

Приведем пример неформализованной оценки, выполненной в соответствии с моделью.

Пример 12.1

В качестве анализируемой организации возьмем крупный банк.

На начальном этапе анализа попытаемся выявить наличие в ИТ-службе количественных оценок эффективности ее деятельности. Это можно выяснить путем анализа регламентной базы или опросом сотрудников.

Наиболее вероятный вариант: — присутствуют данные о количестве инцидентов, связанных со сбоями отдельных критичных для деятельности банка систем, а также сведения о плановом и использованном бюджете на текущий год. Других количественных метрик нет. Таким образом, основа уровня количественной управляемости не присутствует в необходимом объеме.

Далее проверяем наличие проектной и эксплуатационной документации. Пусть в нашем примере есть отдельные проектные документы в виде «Спецификаций требований пользователя», а также эксплуатационные документы в виде «Руководства пользователя» на отдельные автоматизированные системы. Руководства администраторов, программистов, детальные проекты сетевой архитектуры систем, схемы взаимодействия прикладного программного обеспечения и т. д. — отсутствуют. Следовательно, уровень ниже, чем у показателя регламентированности (не определены и не описаны основные процессы).

В нашем случае существует утвержденный внутренний документ, регламентирующий проектную деятельность, но он соблюдается не всегда, т. е. исполнительская дисциплина находится на низком уровне.

Большинство проектов реализуются с превышением сроков или завершаются неуспешно. Надежность функционирования отдельных систем и динамика их развития в определяющей степени зависят от сотрудников, занятых их сопровождением и развитием, присутствует высокий риск потери возможности поддержки функционирования и развития систем в случае увольнения отдельных сотрудников. Таким образом, уровень зрелости анализируемой организации — «начальный» или «управляемости».

Полученная в рассмотренном примере экспресс-оценка, к сожалению, типична, но, несмотря на простоту, проводится редко.

Следует отметить, что приведенная в примере оценка не соответствует требованиям к формальной оценке, включенным в *СММИ*. Но это не делает ее некорректной, а простота и наглядность соответствует изначальному смыслу модели. Формальная оценка проводится по более сложным правилам и дает более точный и повторяемый результат.

Определенный уровень не является непосредственно хорошей или плохой характеристикой организации. Отрицательной характеристикой может быть скорее отсутствие стремления к повышению уровня зрелости и содержательной работы в этом направлении. При этом следует принимать во внимание то, что в отдельных случаях (например, мелкий бизнес) повышенный уровень зрелости оказывается необоснованно затратным, не принося выгод в части повышения управляемости.

Эволюционность, заложенная в *СММИ*, означает невозможность и нежелательность революционных преобразований. Например, ваша организация находится на уровне «начальный», и в целом ее деятельность является успешной. При этом руководство полагается на ряд ключевых специалистов, даже не представляя в деталях, чем они заняты. Нет смысла пытаться перейти на 5-й уровень (оптимизируемости), хотя бы потому, что от коллектива будет невозможно добиться достаточного осознания процессов, выполняемых на этом уровне, их целей и характеристик.

Стратегическим направлением развития должен быть выбран уровень 2 или 3 («управляемый» или «регламентированный»). В этом случае руководство и коллектив сможет разобраться в том, что собирается получить, где и когда будут происходить действия, кто их будет контролировать¹.

Таким образом, модель позволяет обоснованно ставить стратегические цели и обеспечивать их достижимость.

Выгоды от знания и использования подходов *СММИ* на практике обеспечиваются эффективностью их применения для решения ряда типовых задач, включая принятие решения о возможности и результативности реализации отдельных ИТ-проектов в конкретной организации.

Как невозможно было бы внедрение конвейера в эпоху Троянской войны, так невозможно и эффективное внедрение *ERP*-систем в организации, находящейся на начальном уровне зрелости.

Анализ текущего уровня зрелости организации позволяет обоснованно делать заключение о потенциале реализации крупных ИТ-проектов и их рисках. Чем ниже уровень зрелости, тем более рискованно инвестирование в крупные интеграционные проекты — они становятся задачей для «героев» и не гарантируют результативность. Например, в работе *Managing capability and maturity*² приведено мнение одного из руководителей компании *ASG* Майк Ойцман (*Mike Oitzman*): «Необходимо достижение высокого уровня зрелости для успешной реа-

¹ *Norfolk D. Managing capability and maturity.*

² Там же.

лизации проекта внедрения базы данных компонентов информационной системы (*configuration management database*)».

В целом список потенциальных выгод от ориентации на *СММ* модель включает:

- поиск обоснованного решения по направлениям совершенствования деятельности ИТ-служб, исходя из отдельных бизнес-целей;
- возможность обоснованного формирования стратегии развития информационных технологий в организации с целью повышения эффективности деятельности организации в целом;
- анализ актуальности и рисков крупных проектов в области развития информационных технологий;
- получение экспресс-оценки деятельности ИТ-служб;
- возможность эффективного и сравнимого внешнего аудита процессов;
- демонстрацию конкурентного преимущества для клиентов и контрагентов в случае успешной аттестации на высокий уровень зрелости.

С ростом уровня зрелости происходит не только переход от управления проектами к управлению процессами, но также снижаются риски, связанные с деятельностью, персональная зависимость, возрастает качество предоставляемых продуктов и услуг.

Кроме того, в литературе, посвященной данной теме, приводятся данные о снижении уровня конфликтов и превалировании компромиссных и взаимовыигрышных (англ. *win-win*) стратегий при росте зрелости, что является вполне объяснимым.

Как уже говорилось выше, модель основана на принципе эволюционности развития. Этот принцип не только накладывает ограничения на попытки скачкообразных революционных изменений, но и предусматривает рассмотрение любой организации или ее части в качестве поступательно развивающейся системы.

Модель предполагает, что организация достигает повышения зрелости, первоначально добиваясь управляемости отдельных проектов, затем, на наиболее высоких уровнях — в рамках непрерывного улучшения, — используя объективные количественные показатели процессов.

Система организации состоит из людей, используемых ими инструментов и оборудования, методов и процедур работы. Эти составляющие поддаются замене — специалист может занять вакансию и покинуть ее, оборудование замещается в связи с износом или в связи с устареванием, отдельные процедуры и методы обработки продукта или предоставления услуги могут совершенствоваться, замещаться или исключаться. При этом функционирование организации, предоставление услуг и выпуск продуктов с ее стороны сохраняются за счет поддержания компоненты, которая является основополагающей и объединяющей остальные составляющие — процесса. Именно процесс находится в центре внимания модели зрелости.

Ориентация на процессный подход демонстрирует приверженность разработчиков модели принципам, заложенным в международные стандарты серии *ISO 9000*, методику 6 *Sigma*, рекомендации *ITIL* и т. п.

В публикациях по применению модели зрелости присутствуют многочисленные указания на взаимное дополнение модели и других стандартов управления качеством.

12.3. Описание СММІ

Таблица 12.4 достаточно информативна для «экспресс-диагностики» организации, но детализированное описание СММІ содержит множество дополнительных данных, позволяющих значительно повысить точность оценки, что в свою очередь позволит выработать меры корректирующего воздействия на существующие процессы.

В высокотехнологичной среде XX в. почти все организации столкнулись с необходимостью обеспечивать создание, поддержку и предоставление все более сложных продуктов и услуг. При этом составляющие данных продуктов и услуг компоненты частично производятся внутри организации, а частично предоставляются со стороны с последующей интеграцией в рамках конечного продукта или услуги.

Структура текущей версии модели зрелости (1.3) отражает полный набор описанных процессов и состоит из трех основных блоков, изложенных в отдельных книгах:

- разработка продуктов и услуг (*CMMI for Development model*);
- внедрение услуг, управление услугами, предоставление услуг (*CMMI for Services model*);
- приобретение продуктов и услуг (*CMMI for Acquisition model*).

Каждая из книг состоит примерно из 500 страниц. В дополнение к указанным книгам модель содержит требования к проведению обследования и оценки.

Справочный материал в каждой из книг предваряется описанием основных принципов и структуры справочной информации. В основном справочном блоке приводится описание процессных областей, которые должны исполняться на определенных уровнях зрелости. Описание каждой процессной области жестко структурировано и разбито на подразделы (табл. 12.5).

Таблица 12.5

Основные подразделы описания процессной области СММІ

Подраздел	Описание
1. Формулировка задачи (<i>Purpose Statements</i>)	Описание основной задачи процесса. Например, для процесса «Управление конфигурациями» (<i>Configuration Management</i>) — создание и поддержка целостности создаваемого продукта или предоставляемой услуги за счет описания конфигурации, ее контроля и аудита

Подраздел	Описание
2. Связанные процессы (<i>Related Process Areas</i>)	При выполнении процессы могут вызывать другие процессы и использовать их результаты. В данном подразделе приводится список взаимодействующих процессов. Например, в описании процесса «Планирование проектов» (<i>Project Planning</i>) присутствует ссылка на процессную область «Управление рисками» (<i>Risk Management</i>)
3. Специальные цели (<i>Specific Goals</i>)	Здесь приводится состав специфических для данной процессной области параметров, которые должны соблюдаться при выполнении процесса. Данные параметры подлежат проверке в случае проведения аудита уровня зрелости и проверки наличия, корректности и полноты выполнения конкретного процесса. Например, для процесса «Управление конфигурациями» (<i>Configuration Management</i>) определено три специальные цели: «Создание базиса идентифицированных продуктов и услуг» (<i>Establish Baseline</i>), «Протоколирование и контроль изменений» (<i>Track and Control Changes</i>) и «Обеспечение целостности» (<i>Establish Integrity</i>)
4. Специальные практики (<i>Specific Practices</i>)	В данном подразделе приводится описание действий, которые выполняются для достижения специальных целей. Например, одной из специальных практик для специальной цели «Создание базиса идентифицированных продуктов и услуг» является «Внедрение системы управления конфигурациями» (<i>Establish a Configuration Management System</i>), под которой понимается система для хранения соответствующих данных, набор процедур их обновления и предоставления. В целях уточнения обычно приведены «Подпрактики» (<i>Subpractices</i>) и получаемые результаты. Для приведенной выше присутствует подпрактика — «Обеспечить хранение архивных записей об изменении описания объектов» — и вырабатываемый результат — «Процедуры доступа к данным»
5. Общие цели (<i>Generic Goals</i>)	Общие цели составляют набор целей, типовых для большинства процессов и характеризуют уровень внедрения каждого процесса. Они же являются основными признаками зрелости процессной области при проведении аудита. Примером общей цели может служить наличие регламентирующей документации по процессу
6. Общие практики (<i>Generic Practices</i>)	Общие практики соответствуют деятельности, необходимой для реализации Общих целей и являются типовыми для большинства процессных областей. Пример общей практики для общей цели «Процесс существует в качестве управляемого» (уровень 2) — «Обеспечить наличие необходимых ресурсов для выполнения процесса, выработки соответствующих рабочих результатов»

В табл. 12.6 приведен пример описания процессной области.

Описание процесса «Управление конфигурациями» (*Configuration Management*)

Подраздел	Описание
Формулировка задачи (<i>Purpose Statements</i>)	Задача процесса состоит в создании и поддержании целостности результатов работы с помощью определения конфигурации, ее контроля и аудита
Связанные процессы (<i>Related Process Areas</i>)	<ul style="list-style-type: none"> • Планирование проектов (<i>Project Planning</i>) содержит дополнительную информацию о разработке планов, создаваемых структурах, которые позволяют определять конфигурацию и ее изменения; • мониторинг и контроль проектов (<i>Project Monitoring and Control</i>) позволяет получить дополнительную информацию об анализе параметров производительности и корректирующих действиях
Специальные цели (<i>Specific Goals</i>)	<ul style="list-style-type: none"> • Установка базиса (<i>Establish Baselines</i>) — определение настроечных параметров, компонентов, результатов работы и их параметров, подпадающих под управление конфигурациями, создание системы управления конфигурациями; • протоколирование и контроль изменений (<i>Track and Control Changes</i>); • обеспечение целостности (<i>Establish Integrity</i>)
Специальные практики (<i>Specific Practices</i>)	<p>Установка базиса <i>Идентификация учитываемых объектов (Identify the configuration items)</i> Определение и специфицирование следующего:</p> <ul style="list-style-type: none"> • продукты (услуги), предоставляемые клиентам; • связанные внутренние результаты работы; • приобретаемые услуги и изделия; • инструменты и другие составляющие производственной среды; • другие объекты, используемые при создании и описании данных результатов работы. <p>Учитываемые объекты также включают спецификации и описания интерфейсов в рамках требований к продукту. Другие документы, включая результаты тестирования, также могут быть включены в зависимости от их критичности для уточнения характеристик продукта. Учитываемые объекты должны выбираться на основе документированных критериев, например, «изменение объекта приводит к обязательному изменению связанного объекта». Частью учитываемого объекта может быть: описание процесса, требования, дизайн, методика испытаний, результаты испытаний, описание интерфейса, схемы, исходные тексты, компиляторы и т. д. Учитываемые объекты должны иметь уникальные номера, дату постановки на учет, указание ответственного и описание значимых характеристик, например, автор, наименование файла, язык программирования и т.д.</p> <p><i>Внедрение Системы управления конфигурациями (Establish a Configuration Management System)</i> Система управления конфигурациями должна включать соответствующую базу данных, процедуры и инструментарий доступа к ней, включая подсистему разграничения прав доступа пользователей. Система должна позволять протоколировать и отслеживать изменения учитываемых объектов.</p>

Подраздел	Описание
	<p><i>Установка базиса (Release Baselines)</i> Создание набора согласованных и утвержденных спецификаций результатов работы и продуктов для внутреннего использования и предоставления клиентам.</p> <p>Протоколирование и контроль изменений (Track and Control Changes) <i>Протоколирование запросов на изменение (Track Change Requests)</i> Запросы могут быть следствием не только изменений в требованиях к продукту или услуге, но и результатом сбоя или обнаруженного дефекта. Запросы подлежат согласованию с заинтересованными лицами, анализу на предмет влияния на взаимодействующие процессы и продукты, бюджет и планирование. Запросы должны отслеживаться по ходу исполнения.</p> <p><i>Контроль изменений учитываемых объектов (Control Configuration Items)</i> Контроль производится на соответствие заданному базису. В случае необходимости обеспечивается внесение изменений в базис (с указанием причин) по результатам необходимого согласования. Результатом реализации практики может являться протокол ревизии, протокол изменений базиса.</p> <p>Обеспечение целостности (Establish Integrity) <i>Протоколирование управления конфигурациями (Establish Configuration Management Records)</i> Возможные результаты практики включают: историю ревизий учитываемых объектов, протокол изменений, и запросов на изменения, статус учитываемых объектов, различия между версиями базиса.</p> <p><i>Аудит конфигурации (Perform Configuration Audits)</i> Проверка соответствия базиса и документации установленным критериям и стандартам. Аудит может проводиться в части соответствия функциональных и технических характеристик объекта конфигурации, проектной и эксплуатационной документации, в части наличия и комплектности эксплуатационной документации, а также наличия и комплектности записей в базе данных системы управления конфигурациями.</p>
Общие цели (Generic Goals) и общие практики (Generic Practices)*	<p>Достижение специальных целей (Achieve Specific Goals) Процесс обеспечивает достижение специальных целей путем преобразования входящих материалов и ресурсов в специфицированные результаты с применением специальных практик.</p>

Подраздел	Описание
	<p>Процесс внедрен в качестве управляемого (<i>Institutionalize a Managed Process</i>) При этом утверждены внутренние нормативные документы, регламентирующие процесс. Кроме того, существует утвержденный календарный план процесса, процесс снабжен необходимыми ресурсами, назначены ответственные за реализацию процесса, задействованные специалисты прошли обучение, заинтересованные лица и подразделения определены и привлечены к согласованию документов, процесс имеет внешний контроль, высший менеджмент контролирует количественные данные процесса.</p> <p>Для уровня зрелости 3 и выше: процесс внедрен в качестве регламентированного (<i>Institutionalize a Defined Process</i>) При этом на систематической основе обеспечивается сбор количественных характеристик процесса, а также статистика его развития и предложения по улучшению с формированием соответствующей управленческой отчетности, обеспечивается планирование количественных показателей процесса, выявляются и устраняются причины его сбоев и недостатков</p>

* В приводимом описании общие практики не выделены подразделами в целях сокращения текста. Полное описание см. в официальной документации.

Целью учебного пособия не является дублирование официальной документации *СММІ*. Приводимая информация должна дать общее представление о модели зрелости, ее задачах, возможностях и составе, а также облегчить задачу самостоятельного изучения в случае необходимости.

Модель в части разработки продуктов и услуг (англ. *СММІ for Development model*) содержит описание 22 процессов. В части внедрения услуг, управления услугами, предоставлением услуг (англ. *СММІ for Services model*) — 24, в части приобретения продуктов и услуг (*СММІ for Acquisition model*) — 22. Несмотря на то, что основной набор процессов дублируется (16 общих процессов считаются ядром модели), присутствует ряд расхождений (табл. 12.7).

Таблица 12.7

Процессы уровней зрелости для различных областей *СММІ*

№ п/п	Разработка продуктов и услуг (<i>СММІ for Development model</i>)	Внедрение услуг, управление услугами, предоставлением услуг (<i>СММІ for Services model</i>)	Приобретение продуктов и услуг (<i>СММІ for Acquisition model</i>)
Уровень зрелости — 2 (управляемости)			
1.	Управление конфигурациями (<i>Configuration Management</i>)	Управление конфигурациями (<i>Configuration Management</i>)	Управление конфигурациями (<i>Configuration Management</i>)

№ п/п	Разработка продуктов и услуг (CMMI for Development model)	Внедрение услуг, управление услугами, предоставлением услуг (CMMI for Services model)	Приобретение продуктов и услуг (CMMI for Acquisition model)
2.	Планирование проектов (Project Planning)	Планирование проектов и работ (Work Planning)	Планирование проектов (Project Planning)
3.	Мониторинг и контроль проектов (Project Monitoring and Control)	Мониторинг и контроль проектов и работ (Work Monitoring and Control)	Мониторинг и контроль проектов (Project Monitoring and Control)
4.	Количественные измерения и анализ (Measurement and Analysis)	Количественные измерения и анализ (Measurement and Analysis)	Количественные измерения и анализ (Measurement and Analysis)
5.	Управление требованиями (Requirements Management)	Управление требованиями (Requirements Management)	Управление требованиями (Requirements Management)
6.	Контроль качества процессов и продуктов (Process and Product Quality Assurance)	Контроль качества процессов и продуктов (Process and Product Quality Assurance)	Контроль качества процессов и продуктов (Process and Product Quality Assurance)
7.	Управление внешними договорами (Supplier Agreement Management)	Управление внешними договорами (Supplier Agreement Management)	Управление внешними договорами (Agreement Management)
8.	—	Предоставление услуг (Service Delivery)	Разработка требований (Acquisition Requirements Development)
9.	—	—	Запрос и разработка договора (Solicitation and Supplier Agreement Development)
Уровень зрелости — 3 (регламентированности)			
10.	Общее управление проектом (Integrated Project Management)	Общее управление работами (Integrated Work Management)	Общее управление проектом (Integrated Project Management)
11.	Управление рисками (Risk Management)	Управление рисками (Risk Management)	Управление рисками (Risk Management)

№ п/п	Разработка продуктов и услуг (CMMI for Development model)	Внедрение услуг, управление услугами, предоставлением услуг (CMMI for Services model)	Приобретение продуктов и услуг (CMMI for Acquisition model)
12.	Внесение изменений в процессы (Organizational Process Focus)	Внесение изменений в процессы (Organizational Process Focus)	Внесение изменений в процессы (Organizational Process Focus)
13.	Регламентированность процессов (Organizational Process Definition)	Регламентированность процессов (Organizational Process Definition)	Регламентированность процессов (Organizational Process Definition)
14.	Организационное обучение (Organizational Training)	Организационное обучение (Organizational Training)	Организационное обучение (Organizational Training)
15.	Управление принятием решений (Decision Analysis and Resolution)	Управление принятием решений (Decision Analysis and Resolution)	Управление принятием решений (Decision Analysis and Resolution)
16.	Интеграция продукта (Product Integration)	Управление доступностью и производительностью (Capacity and Availability Management)	Техническое управление (Acquisition Technical Management)
17.	Валидация (Validation)	Управление инцидентами и их предотвращение (Incident Resolution and Prevention)	Валидация (Acquisition Validation)
18.	Верификация (Verification)	Непрерывность услуг (Service Continuity)	Верификация (Acquisition Verification)
19.	Разработка требований (Requirements Development)	Разработка системы предоставления услуг (Service System Development)	—
20.	Техническое решение (Technical Solution)	Изменение системы предоставления услуг (Service System Transition)	—
21.	—	Стратегическое управление услугами (Strategic Service Management)	—
Уровень зрелости — 4 (количественной управляемости)			
22.	Организационный контроль эффективности (Organizational Process Performance)	Организационный контроль эффективности (Organizational Process Performance)	Организационный контроль эффективности (Organizational Process Performance)

№ п/п	Разработка продуктов и услуг (CMMI for Development model)	Внедрение услуг, управление услугами, предоставлением услуг (CMMI for Services model)	Приобретение продуктов и услуг (CMMI for Acquisition model)
23.	Количественное управление проектами (Quantitative Project Management)	Количественное управление работами и проектами (Quantitative Work Management)	Количественное управление проектами (Quantitative Project Management)
Уровень зрелости — 5 (оптимизируемости)			
24.	Анализ и разрешение проблем (Causal Analysis and Resolution)	Анализ и разрешение проблем (Causal Analysis and Resolution)	Анализ и разрешение проблем (Causal Analysis and Resolution)
25.	Управление эффективностью (Organizational Performance Management)	Управление эффективностью (Organizational Performance Management)	Управление эффективностью (Organizational Performance Management)

Часто для удобства прочтения процессные области дополнительно структурируют по направлению деятельности (табл. 12.8).

Таблица 12.8

Процессные области по направлениям для CMMI for development

Process	Project	Engineering	Support
Регламентированность процессов (L3 OPD)	Общее управление проектом (L3 IPM)	Интеграция продукта (L3 PI)	Анализ и разрешение проблем (L5 CAR)
Внесение изменений в процессы (L3 OPF)	Мониторинг и контроль проектов (L2 PMC)	Разработка требований (L3 RD)	Управление конфигурациями (L2 CM)
Управление эффективностью (L5 OPM)	Планирование проектов (L2 PP)	Техническое решение (L3 TS)	Управление принятием решений (L3 DAR)
Организационный контроль эффективности (L4 OPP)	Количественное управление проектами (L4 QPM)	Валидация (L3 VAL)	Контроль качества процессов и продуктов (L2 PPQA)
Организационное обучение (L3 OT)	Управление требованиями (L2 REQM)	Верификация (L3 VER)	Количественные измерения и анализ (L2 MA)
—	Управление рисками (L3 RSKM)	—	—
—	Управление внешними договорами (L2 SAM)	—	—

СММІ допускает два базовых подхода к ее применению: *непрерывное представление* и *представление на основе состояний* (рис. 12.4).

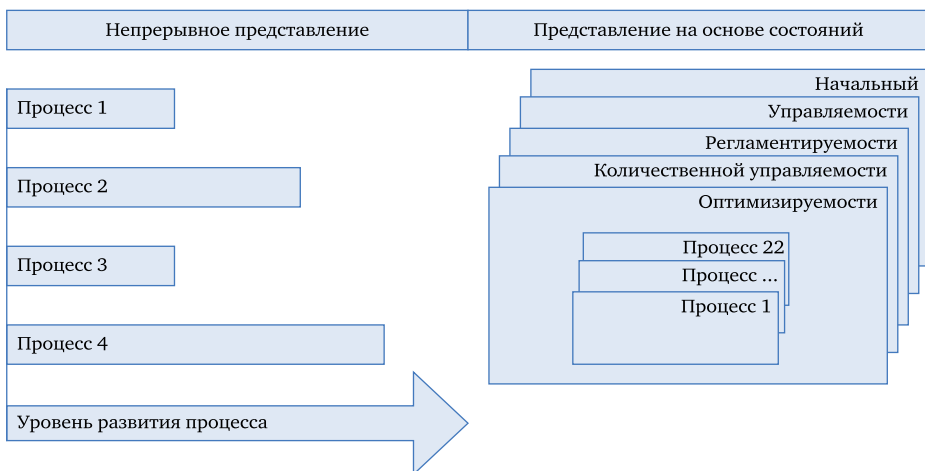


Рис. 12.4. Непрерывное представление и представление на основе состояний

Первый подход является относительно новым, более детализированным, применяется в целях анализа и совершенствования отдельных процессов, существующих в организации, на основе определения уровня развития этих процессов, например, в случае необходимости развития некоторого проблемного технологического участка или в целях повышения качества предоставления отдельной услуги. При этом, безусловно, приходится принимать во внимание наличие тесных связей между процессами, что иногда исключает возможность индивидуального развития отдельного процесса.

Данный подход, таким образом, является более гибким и в большинстве случаев, за счет возможности фокусировки на отдельных направлениях, менее затратным.

Вероятно, введение данного подхода и соответствующее дополнение к структуре модели явилось результатом развития конкурирующего стандарта качества процессов разработки программного обеспечения — *ISO 15 504 (SPICE, от англ. Software Process Improvement and Capability dEtermination)*, предлагающего многомерный взгляд на процессы.

Второй подход является наследником ранее применяемых моделей зрелости СММІ и используется в основном для анализа текущего уровня зрелости организации или ее части на основе комплектности определенного для заданного уровня набора процессов.

Официальная документация СММІ¹ рекомендует на начальном уровне применения модели использовать традиционное представление на основе состояний.

¹ URL: <http://www.sei.cmu.edu/cmми>.

Непрерывное представление требует наличия в организации развитого представления о процессной организации деятельности, что само по себе является признаком не начального уровня зрелости.

Оба подхода (представления модели) связаны с понятием уровня. В случае непрерывного представления — это четыре уровня развития (англ. *capability*¹) процесса в диапазоне от 0 до 3 (табл. 12.9).

Таблица 12.9

Описание уровней в непрерывном представлении (уровни развития процессов)

Уровень	Описание	Комментарий
0	Не полностью исполняемый (<i>Incomplete</i>)	Процесс не исполняется или исполняется частично, т. е. не реализуется одна или более специфических целей процесса
1	Выполняемый (<i>Performed</i>)	На данном уровне реализуются специфические цели процесса. Процесс не является устойчивым
2	Управляемый (<i>Managed</i>)	Процесс является стабильным, устойчивым к нештатным ситуациям, имеет необходимую инфраструктуру. Он выполняется в плановом порядке на основе регламентирующей документации, с привлечением необходимых исполнителей и контролирующих заинтересованных лиц. Обеспечивается мониторинг, контроль и оценка исполнения процесса в соответствии с документацией на него
3	Регламентированный (<i>Defined</i>)	Процесс регламентирован и соответствует требованиям уровня 2. Процесс реализуется в рамках единой процессной архитектуры организации. Присутствует набор стандартов и более подробных описаний процессов, включая: цель, вход, операции, роли, количественные параметры, критерии верификации, результаты, критерии успешности. На этом уровне процессы управляются более проактивно

В случае представления на основе состояний — это пять уровней зрелости (англ. *maturity*) организации или подразделения (от 1 до 5).

Вне зависимости от выбранного представления принцип определения уровней един. Уровни характеризуют совершенствование от «болезненного» неупорядоченного состояния к «здоровому» состоянию, при котором используется количественная информация для определения того, что необходимо улучшить, и выполнение этого в соответствии с бизнес-задачами.

Различие в количестве уровней объясняется тем, что нулевой уровень зрелости организации означал бы прекращение ее функциониро-

¹ В русскоязычной литературе встречается различный перевод термина. В данном пособии используется перевод «уровень эффективности» как наиболее соответствующий смыслу модели.

вания. При этом нулевой уровень развития процесса подлежит регистрации в целях определения уровня зрелости организации.

При выборе подхода к применению модели в том числе должны приниматься во внимание три категории факторов:

1) *бизнес-факторы*, которые подразумевают определенность бизнес-целей и их связей с некоторым набором процессов. Если цель — совершенствование отдельной услуги или продукта, то выбирается непрерывное представление, если цель — развитие организации в целом, то предпочтительным является представление модели на основе состояний;

2) *внутренняя культура организации* — если для конкретной организации процессный подход является привычным, то для нее, вероятно, более предпочтительным будет непрерывное представление;

3) *предыдущий опыт* — для организации, имеющей опыт применения модели зрелости, основанной на состояниях, инвестировавшей ресурсы ранее в применение подобной модели, данное представление СММІ будет менее затратным.

Описываемые подходы (таб. 12.10) к применению модели зрелости не являются взаимоисключающими. В большинстве случаев при разработке плана работ по совершенствованию процессов в реальных организациях эти подходы комбинируются. Например, некоторая организация может принять решение о выделении и развитии отдельных процессов на основе непрерывного представления, при этом предусматривая на завершающем этапе прохождения аудита на соответствие некоторому уровню зрелости, что является использованием представления на основе состояний.

Таблица 12.10

Сравнительная характеристика непрерывного представления и представления на основе состояний

Непрерывное представление	Представление на основе состояний
Предоставляет полную свободу выбора последовательности совершенствования деятельности организации, исходя из ее приоритетных бизнес-целей	Содействует организации в определении зарекомендовавшей себя траектории развития
Обеспечивает наглядность достигнутого приращения качества и производительности отдельных процессов	Позволяет определить возможности организации на основе выявления присутствующего набора процессов, характерного для определенного уровня зрелости
Развитие различных процессов может выполняться в разном темпе	Позволяет консолидировать результаты прогресса в едином широко распознаваемом значении — уровне зрелости

Непрерывное представление	Представление на основе состояний
Является новым подходом, который пока не предоставил возможности собрать достаточную статистику для подтверждения результативности	Имеет длительный опыт применения, который демонстрирует высокую результативность, в том числе в части возврата инвестиций
Позволяет оценивать совершенство процессов	Позволяет получать сравнительную оценку организаций и подразделений, в том числе с различными областями деятельности

Приведем пример применения непрерывного представления модели.

Пример 12.2

Разработчик электронных устройств периодически запаздывает с выпуском очередной версии устройств.

Учитывая необходимость ускорения выпуска очередной версии и в целях экономии ресурсов, сужаем набор совершенствуемых процессов. В данном случае из 22 процессных областей выбираем процессы, относящиеся к области проектирования и управления проектами (англ. *engineering and project management*): *Product Integration* (интеграция продукта — обеспечение взаимодействия продукта с подключаемыми устройствами), *Requirements Development* (разработка требований), *Requirements Management* (управление требованиями), *Technical Solution* (техническое решение), *Validation* (валидация — подтверждение требований пользователя), *Verification* (верификация — частное и комплексное тестирование на соответствие требованиям), а также *Project Planning* (планирование проекта), *Project Monitoring* и *Control* (мониторинг и контроль проекта).

Такое количество процессов потребует значительных усилий для параллельного совершенствования, поэтому пробуем найти наиболее проблемные (потенциально выигрышные) с точки зрения бизнес-целей участки и в итоге сосредотачиваемся на разработке требований (*Requirements Development*) и управлении требованиями (*Requirements Management*).

Далее необходимо принять решение об объеме улучшений:

- если приведенные выше процессы отсутствуют в организации, то обоснованным выбором будет создание процесса и достижение уровня 1.
- процессы выполняются в рамках каждого проекта, но при этом не управляемы в необходимой степени (регулирующие документы, обучение, инструментарий не внедрены) - решение — достижение уровня 2.
- процессы присутствуют, но в рамках каждого проекта выполняются не единообразно — достижимым уровнем будет — третий.
- процессы выполняются стандартным образом, но отсутствует объективная возможность контролировать их параметры — пытаемся добиться уровня 4.
- если есть понимание того, какие элементы данных процессов следует улучшать в соответствии с какими количественными характеристиками, то целью может быть выбран уровень 5.

Рассмотрим пример применение представления, основанного на состояниях.

Пример 12.3

Компания — разработчик программного обеспечения, недавно прошедшая аудит на соответствие уровню зрелости 2. Цель — повышение уровня зрелости до 3.

Составляем список процессных областей, входящих в требования к уровню 3 и не включаемых в требования уровня 2: *Requirements Development* (разработка требований), *Technical Solution* (техническое решение), *Product Integration* (интеграция продукта), *Verification* (верификация), *Validation* (валидация), *Organizational Process Focus* (организационный процесс), *Organizational Process Definition* (поддержка организационных процессов), *Organizational Training* (организационное обучение), *Integrated Project Management* (общее управление проектом), *Risk Management* (управление рисками), *Decision Analysis and Resolution* (анализ решений).

Далее выполняем анализ недостатков, включая соответствие элементов существующих процессов требованиям модели.

12.4. Оценка зрелости

Наличие развитой системы проведения оценки в соответствии с *СММИ* является одним из основных преимуществ модели перед родственными стандартами и методиками.

Присвоение организации в результате оценки некоего уровня в соответствии с *СММИ* означает ее рейтингование среди других компаний, прошедших оценку, включая прямых конкурентов, с возможностью последующей официальной публикации. Это накладывает жесткие требования и ограничения на методику проведения оценки, учитывая необходимость получения сравнимых, формально обоснованных и точных результатов.

Официальная документация *СММИ* включает «Требования к проведению оценки в соответствии с *СММИ*»¹. Данный документ содержит основные требования к методологии и проведению обследования и оценки.

В соответствии с данными требованиями *Software Engineering Institute* разработал Стандартную методику оценки для совершенствования процессов *СММИ* (*SCAMPI*, от англ. *Standard CMMI Appraisal Method for Process Improvement*), которая может быть непосредственно использована для выявления сильных и слабых сторон существующих процессов, анализа присутствующих рисков, а также определения текущего уровня зрелости организации. Данная методика включает описание этапа подготовки к оценке (например, выбор опций оценки), действий на территории оцениваемой организации, предварительного обследования, процедур сбора сведений, рейтингования, итоговой отчетности и последующих действий.

В соответствии с требованиями *SCAMPI* допускаются три уровня формализации при проведении оценки, которые соответствуют классам А,

¹ Требования к проведению оценки в соответствии с *СММИ*. *Software Engineering Institute*, 2010.

B и *C*. Наиболее формальная оценка (класс *A*) должна проводиться авторизованным *SEI* — лидером-оценщиком (англ. *lead appraiser*) в соответствии с методикой, изложенной в *SCAMPI A Method Definition Document (MDD)*. Только результаты данной оценки могут быть опубликованы как официальные и признаются Министерством обороны США в качестве формальной характеристики потенциального поставщика услуг или продуктов.

Оценка по классу *B* предоставляет возможность выбора в объеме оценки, при этом проводится анализ только действующих практик и на основе единой фиксированной шкалы.

Класс *C* обеспечивает максимальную гибкость в проведении оценки, включая анализ планируемых к внедрению процессов на основе шкалы, которую может предложить клиент.

В связи с тем, что оценка *CMMI* является достаточно ресурсоемкой, при выборе класса оценки необходимо максимально точно определить ее цель. Если оценка проводится в целях получения официального рейтинга, то, безусловно, единственным выбором будет класс *A*. В случае необходимости совершенствования конкретного направления деятельности достаточно будет класса *C*.

Во всех случаях оценка производится путем выявления процессов, признаков их соответствия формальным требованиям модели зрелости: достижение специфических и общих целей, выполнение специфических и общих практик, наличие типовых результатов их выполнения. При этом используется развитая методика проверки и подтверждения получаемых в ходе оценки сведений, которая позволяет гарантировать объективность, повторяемость и точность оценки. В целях обеспечения возможности разбора спорных ситуаций каждый шаг проведения обследования подробно документируется.

Оценка может проводиться для организации в целом или ее части. Обязательным требованием к проведению оценки является предварительное четкое описание и документирование области и ограничений обследования. По результатам оценки обеспечивается подготовка развернутой отчетности в установленном формате. На рис. 12.5 приведен пример фрагмента получаемого отчета (расшифровку сокращенных обозначений процессных областей см. в табл. 12.8).

При анализе достижения соответствующего уровня зрелости (класс оценки *A*) непременно проверяется достижение всеми обязательными для искомого уровня зрелости процессами уровня развития, равного или превышающего соответствующий уровень зрелости. Например, если оценивается достижение организацией уровня зрелости «регламентированный», то каждый из обязательных для этого уровня процессов должен иметь уровень не ниже «регламентированного».

При опубликовании результатов оценки детальный отчет размещается в базе данных *SEI* в целях сбора статистики по различным разделам оценки. Данная информация может использоваться в том числе для уточнении параметров модели при разработке новых версий.

ИТ-блок	Maturity Level 2 Process Areas							
	REQM	WP	WMC	SAM	MA	PPQA	CM	SD
Specific Goal 1 %	50 %	65 %	61 %	75 %	63 %	13 %	42 %	50 %
Specific Goal 2 %		61 %	58 %	75 %	63 %	75 %	75 %	42 %
Specific Goal 3 %		75 %					75 %	75 %
Generic Goal 3 %	54 %	50 %	60 %	50 %	54 %	44 %	50 %	60 %
Specific Practices	5	15	10	6	8	4	7	7
Generic Practices	12	12	12	12	12	12	12	12
Состояние процесса на дату оценки	52 %	63 %	60 %	67 %	60 %	44 %	60 %	57 %

ИТ-блок	Maturity Level 3 Process Areas											
	CAM	IRP	SCON	SST	STSM	OPF	OPD	OT	IWM	RSKM	DAR	SSD
Specific Goal 1 %	17 %	75 %	75 %	58 %	50 %	42 %	54 %	75 %	61 %	42 %	0 %	0 %
Specific Goal 2 %	42 %	75 %	75 %	75 %	50 %	25 %		75 %	58 %	75 %		0 %
Specific Goal 3 %		25 %	75 %			19 %				50 %		0 %
Generic Goal 3 %	50 %	50 %	56 %	50 %	56 %	35 %	35 %	67 %	50 %	50 %	0 %	0 %
Specific Practices	6	10	8	5	4	9	7	7	10	7	6	12
Generic Practices	12	12	12	12	12	12	12	12	12	12	12	12
Состояние процесса на дату оценки	36 %	56 %	70 %	61 %	52 %	30 %	44 %	72 %	56 %	54 %	0 %	0 %



Рис. 12.5. Пример фрагмента отчета по оценке *СММ* на соответствие третьему уровню зрелости

Глава 13

СТАНДАРТ *ISO 20 000 (ITSM)* И БИБЛИОТЕКА РЕКОМЕНДАЦИЙ *ITIL*

На рис. 12.3 представлен потенциальный вклад различных «лучших практик» в воображаемую идеальную методику организации работы ИТ-служб. При этом первенство в части детализации процессов, связанных с ИТ-сервисами, безоговорочно отдано библиотеке рекомендаций *ITIL*. Данная библиотека была разработана в Великобритании и легла в основу международного стандарта *ISO 20 000* (англ. *Information Technology Service Management*). В настоящее время встречается достаточно много критических замечаний как в адрес *ITIL*, так и в адрес *ISO 20 000*. Однако все «лучшие» и подтвержденные практики систематически утверждаются в качестве стандартов, а «самые лучшие» из них — в качестве международных стандартов. *ISO 20 000* является именно таким стандартом и, безусловно, любое учебное пособие в области ИТ-процессов не будет полным без раскрытия особенностей данного стандарта.

13.1. История стандарта *ISO 20 000* и *ITIL*

Исторически правительство Великобритании часто становилось источником международных стандартов в самых разных областях деятельности. В конце 1980-х гг., столкнувшись с необходимостью упорядочения деятельности в области информационных технологий, и не в последнюю очередь под влиянием развития общей теории качества, одно из подразделений британского правительства — Центральное агентство по вычислительной технике и коммуникациям (англ. *Central Computer and Telecommunications Agency, CCTA*) разработало набор рекомендаций, который в дальнейшем был назван *Information Technology Infrastructure Library (ITIL)*.

Разработанная библиотека базировалась на процессном подходе к управлению и контролю сервисов, предоставляемых со стороны ИТ-служб. Кроме того, в основу библиотеки был положен цикл постоянного улучшения Деминга — Шухарта (см. гл. 4).

Первичный набор (*ITIL* версия 1) был издан в период с 1989 по 1996 г. и состоял более чем из 30 книг. Вторая версия была напечатана в 2001 г.

в значительно измененном и реструктурированном виде, поскольку первая версия в силу объема была плохо доступной для широкого круга специалистов.

Третья версия была издана в 2007 г. и состояла уже из 5 томов, содержащих описание 26 процессов. В 2011 г. данная версия была переиздана с некоторыми изменениями. В процессе переиздания *ITIL* неоднократно изменялась организация, являющаяся основным издателем. Сегодня это *UK Cabinet Office*, т. е. правительство Великобритании.

В дальнейшем на основе *ITIL* был разработан британский стандарт *BS 15 000:2002* (2002 г.), который стал базой для создания в 2005 г. международного стандарта *ISO 20 000* практически без внесения значимых изменений. Здесь необходимо отметить, что между *ITIL* и *ISO 20 000* существует разница в целях и структуре, из-за чего формально различаются количество описываемых процессов и другие детали. И при этом *ITIL*, безусловно, не корректно было бы называть стандартом.

В России стандарт *ISO 20 000* был принят в качестве государственного (ГОСТ Р ИСО/МЭК 20 000-1-2013) без внесения изменений в текст (прямой перевод) в 2013 г. и затем с небольшими правками переиздан в 2015 г.

13.2. Состав стандарта *ISO 20 000*

Стандарт *ISO 20 000* (ГОСТ Р ИСО/МЭК 20 000-1-2013) состоит из двух частей: *ISO 20 000-1 Information Technology: Specification* (Требования к системе управления услугами) — содержит описание требований к системе менеджмента ИТ-сервисов из 10 разделов, 13 процессов (рис. 13.1), *ISO 20 000-2 Information Technology: Code of practice* (Кодекс практической деятельности) — представляет рекомендации по реализации процессов, требования к которым содержит первая часть.



Рис. 13.1. Структура процессов *ITSM*

В группу **процессов предоставления услуг** (англ. *service delivery process*) входят: управление мощностями, управление непрерывностью и доступностью услуг, управление уровнем услуг, формирование и предоставление отчетности по услугам, управление информационной безопасностью, бюджетирование и учет затрат на услуги.

В группу **процессов взаимоотношений** (англ. *relationship processes*) входят: управление взаимоотношениями с бизнесом и управление подрядчиками.

Группу **процессов разрешения** «Resolution processes» образуют: Управление инцидентами и запросами на обслуживание и Управление проблемами.

В группу **процессов контроля** (англ. *control processes*) вошли процессы: управление конфигурациями, управление изменениями, управление релизами и внедрением. В данной группе присутствует отклонение 20 000 ГОСТ Р ИСО/МЭК 20 000-1-2013 от предыдущей версии ГОСТ и ISO: управление релизами и внедрением не выделен в отдельную группу.

Как уже говорилось выше, стандарт *ISO 20 000* предполагает активное использование цикла постоянного улучшения Деминга — Шухарта — цикл *PDCA* (см. гл. 4):

- *plan* — спроектировать и спланировать улучшение процесса;
- *do* — выполнить пробное изменение;
- *check* — измерить и проверить результативность;
- *act* — принять решение об изменении процесса.

Кроме того, в стандарте заложено требование создания комплексной системы управления ИТ-услугами с выделением заказчика, ответственностью руководства за управление документацией, компетенциями и подготовкой персонала. То есть стандарт полностью соответствует основным положениям *TQM*.

Следует отметить, что, несмотря на полное сходство по тексту между *ISO 20 000* и ГОСТ Р ИСО/МЭК 20 000, цели и порядок сертификации совершенно различны. Органы сертификации ГОСТ Р ИСО/МЭК не могут подменять структуру сертификации *ISO*.

Целью сертификации может являться формализованное требование со стороны организации — потребителя услуг на внутриведомственном или государственном уровне, а также намерение продемонстрировать конкурентоспособное качество предоставляемых услуг для коммерческого потребителя.

13.4. Библиотека *ITIL*

Являясь основой для большинства стандартов в области управления ИТ-процессами, *Information Technology Infrastructure Library* установила несколько базовых принципов:

- ИТ-служба является поставщиком ИТ-услуг на платной основе;

- пользователь является потребителем и заказчиком ИТ-услуг, а не систем;
- ИТ-услуги должны иметь объективно измеряемые количественные показатели качества и стоимости;
- услуги должны предоставляться с повторяющимся уровнем качества и функциональности;
- внедрение ИТ-услуг возможно при высоких уровнях зрелости процессов;
- предоставление ИТ-услуг обеспечивается при делении на зоны персональной ответственности менеджеров процессов, которые обеспечивают поддержку услуг;
- процесс определяется как повторяющаяся последовательность операций, выполняемых ролями, имеющая вход, цель, результат, стоимость и менеджера.

Текущая версия *ITIL* (2011) описывает 26 процессов, скомпонованных в 5 книг (табл. 13.1):

1) *ITIL Service Strategy* (Стратегия предоставления услуг) — описывает понимание целей организации, портфеля проектов и потребностей потребителя¹;

2) *ITIL Service Design* (Проектирование услуг) — описывает, каким образом преобразовывать стратегию и потребности в услуги²;

3) *ITIL Service Transition* (Преобразование услуг) — описывает развитие и улучшение способностей для создания новых услуг в сопровождаемой среде³;

4) *ITIL Service Operation* (Управление услугами) — описывает каким образом управлять услугами в текущей деятельности⁴;

5) *ITIL Continual Service Improvement* (Непрерывное улучшение услуг): описывает внедрение инкрементальных и крупномасштабных изменений услуг⁵.

Обратите внимание на то, что количество процессов в таблице отличается от цифры, приведенной в начале раздела (их больше, чем 26). Здесь проявляется несогласованность в терминологии и изложении материалов *ITIL V3* различными авторами, принимавшими участие в разработке. В различных книгах *ITIL* отдельные процессы названы функциями и наоборот. По официальной точке зрения главного «архитектора» *ITIL* Шерон Тэйлор, библиотека содержит именно 26 процессов (это следует принимать во внимание при сдаче квалификационных экзаменов).

Список процессов *ITIL* позволяет получить дополнительную выгоду за счет того, что обеспечивает возможность эффективно выстраивать организационную структуру ИТ-службы путем контроля полноты состава процессов, выполняемых отдельными подразделениями иерархии.

¹ Cannon D. *ITIL Service Strategy*. L. : The Stationery Office, 2011.

² Hunnebeck L. *ITIL Service Design*. L. : The Stationery Office, 2011.

³ Rance S. *ITIL Service Transition*. L. : The Stationery Office, 2011.

⁴ Steinberg R. A. *ITIL Service Operation*. L. : The Stationery Office, 2011.

⁵ Lloyd V. *ITIL Continual Service Improvement*. L. : The Stationery Office, 2011.

Структура процессов ИТЛ

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (КРП)
Service strategy (Стратегия услуг)		<p>Определение основных приоритетов инвестиций в развитие сервисов. Стратегически развивает ИТ-организацию и подразделение, повышает их ценность для заказчиков в долгосрочной перспективе. Контроль направления жизненного цикла услуг. Ключевые темы: определение ценности услуг для заказчика, развитие направлений бизнеса, маркетинговый анализ. Играет основную роль в квалификационных экзаменах на ИТЛ</p>	
1.	<i>Strategy management for IT Services</i> (Стратегическое управление ИТ-услугами)	<p>Оценка набора предлагаемых услуг, их эффективности, конкурентов, текущих и потенциальных рыночных ниш для выработки стратегий обслуживания. Контроль реализации стратегии</p>	<ul style="list-style-type: none"> ● Количество запланированных новых услуг; ● количество незапланированных новых услуг; ● количество новых заказчиков; ● количество потерянных заказчиков
2.	<i>Service portfolio management</i> (Управление портфелем услуг)	<p>Обеспечивает привязку ИТ-услуг к бизнес-целям, контролирует достаточность списка и уровня услуг, их определенность для заказчиков при соответствии уровня инвестиций</p>	
3.	<i>Financial management for IT services</i> (Управление финансами для ИТ-услуг)	<p>Управление бюджетом, учетом и затратами провайдера услуг. Обеспечивает финансовую эффективность получения инфраструктуры (цель — эффективно, а не максимально дешево)</p>	<ul style="list-style-type: none"> ● Процент проектов, прошедших регламентированный бюджетный процесс; ● процент проектов, имеющих финансово-экономическое обоснование; ● процент проектов, в которых финансово-экономическое обоснование было верифицировано после завершения проекта; ● процент проектов, завершенных с превышением бюджета; ● суммарное превышение бюджетов проектов за период

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
4.	<i>Demand management</i> (Управление спросом)	Обеспечивает анализ и управление спросом заказчиков, контролирует влияние спроса на набор услуг. Обеспечивает балансировку спроса и мощностей	Количество планов мощности, проверенных относительно спроса
5.	<i>Business relationship management</i> (Взаимоотношение с бизнесом)	Обеспечивает позитивные отношения с заказчиком. Идентифицирует потребности существующих и потенциальных заказчиков, контролирует их удовлетворение	<ul style="list-style-type: none"> ● Количество жалоб заказчиков; ● количество подтвержденных жалоб заказчиков; ● количество опросов заказчиков; ● количество ответченных опросов удовлетворенности относительно количества услуг
<i>Service design</i> (Проектирование услуг)		Обеспечивает создание новых услуг и модернизацию существующих с привязкой к бизнес — среде и ИТ-инфраструктуре	
6.	<i>Design coordination</i> (Координация проектирования)	Координирует мероприятия по созданию новых услуг, включая ресурсы, метрики, цели бизнеса и возможности ИТ-инфраструктуры	Используются KPI всего набора процессов SD
7.	<i>Service catalogue management</i> (Управление каталогом услуг)	Обеспечивает создание и поддержку актуальности каталога услуг, который должен содержать ключевую информацию о: способах запроса услуг, ценах, контактах, предоставляемых материалах, текущем состоянии, взаимовлиянии услуг	Полнота каталога услуг
8.	<i>Service-level management</i> (Управление уровнем услуг)	Обеспечивает достижение соглашений об уровне услуг (англ. <i>Service Level Agreements, SLA</i>) с заказчиками и участие в проектировании услуг, исходя из SLA. Отслеживает исполнение SLA, их согласованность с внутренними операционными соглашениями (англ. <i>Operational Level Agreement, OLA</i>) и вспомогательными контрактами (англ. <i>Underpinning Contracts, UC</i>), а также контролирует создание и обновление необходимых планов обеспечения непрерывности деятельности	<ul style="list-style-type: none"> ● Количество услуг с SLA; ● количество услуг, где SLA подкреплены существующими OLA и UC; ● количество SLA, по которым непрерывно контролируются количественные контрольные метрики; ● количество SLA, которые пересматриваются на регулярной плановой основе; ● количество услуг, SLA по которым успешно исполняются

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
9.	<i>Risk Management</i> (Управление рисками)	Идентификация, оценка и управление рисками. Включает оценку услуг для бизнеса, поиск угроз для этих услуг и оценку уязвимости услуг перед угрозами	<ul style="list-style-type: none"> ● Оценки рисков по отдельным угрозам; ● суммарная оценка риска
10.	<i>Availability management</i> (Управление доступностью)	Определение, анализ, планирование и измерение доступности ИТ-услуг. Отвечает за то, чтобы ИТ-инфраструктура, процессы, инструменты, роли процесса присутствовали в необходимом объеме для предоставления услуг с соблюдением допустимого уровня затрат. Контролирует для всех ИТ-компонент: надежность, поддерживаемость, доступность сервисного обслуживания, резервирование, безопасность информации	<ul style="list-style-type: none"> ● Уровень фактической доступности относительно закрепленного в SLA и OLA; ● количество прерываний в предоставлении услуг; ● средняя длительность прерываний в предоставлении услуг; ● количество метрик контроля доступности
11.	<i>Capacity management</i> (Управление мощностями)	Обеспечение достаточности мощностей ИТ для соблюдения SLA вовремя и эффективно с точки зрения затрат. Контролирует и планирует все ИТ-ресурсы (человеческие, системные) на короткую, среднюю и долгосрочную перспективу	<ul style="list-style-type: none"> ● Количество инцидентов, связанных с недостаточной мощностью ресурсов; ● отклонение фактической мощности от планируемой; ● количество изменений мощности в связи с изменением спроса; ● среднее время разрешения проблем мощности; ● процент резерва мощности в периоды нормальной и повышенной загрузки; ● процент компонент, по которым обеспечен мониторинг загрузки

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
12.	<i>IT service continuity management</i> (Управление непрерывностью)	Обеспечивает наличие возможности соблюдения минимального уровня услуг в случае аварийных ситуаций и их восстановления. Включает предупреждающие мероприятия, планы реагирования на аварии и планы восстановления. В рамках процесса выполняется: приоритизация мероприятий на основе анализа влияния на бизнес (англ. <i>Business Impact Analysis, BIA</i>), анализ рисков для ИТ-услуг (включая анализ угроз, уязвимостей и контролер), оценку возможностей для восстановления, выработку аварийных планов, их регулярное тестирование и пересмотр	<ul style="list-style-type: none"> ● Процент бизнес-процессов, по которым присутствуют аварийные планы или соглашения об аварийных уровнях услуг; ● количество основных угроз, по которым не предусмотрены контролеры; ● средняя задержка между выявлением угрозы и реализацией контролер; ● количество недостатков аварийного планирования, выявленных при тестировании
13.	<i>Information Security management</i> (Управление информационной безопасностью)	Обеспечивает поддержку целостности, доступности и конфиденциальности (ЦКД) информационных ресурсов организации в соответствии с требованиями стандарта <i>ISO 27002</i> .	<ul style="list-style-type: none"> ● Количество реализованных контролер по отношению к выявленным угрозам; ● средняя задержка между выявлением риска и реализацией контролер; ● количество выявленных инцидентов, классифицированных по категориям критичности; ● количество инцидентов безопасности, которые привели к недоступности услуг; ● количество пройденных тестов защиты и соответствующих тренингов; ● количество недостатков, выявленных при прохождении тестов
14.	<i>Supplier management</i> (Управление поставщиками)	Получение необходимого уровня услуг и продуктов от поставщиков в соответствии с уровнем затрат и заключенными договорами. Обеспечение балансировки внешних договоров с <i>OLA</i> и <i>SLA</i> предоставляемых услуг при согласовании, контроль их исполнения и прекращение	<ul style="list-style-type: none"> ● Количество заключенных и проконтролированных контрактов; ● количество нарушенных условий контрактов

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
Service transition (Внедрение услуг)		Описывает процессы внедрения и изменения услуг. Часто рассматривается как «проектная» часть ITIL	
15.	<i>Change management</i> (Управление изменениями)	Контролирует жизненный цикл изменений услуг, выполняемых в ответ на запрос изменений (англ. <i>Request For Change, RFC</i>). Обеспечивает эффективное и низкозатратное внедрение изменений с минимальным риском для доступности услуг	<ul style="list-style-type: none"> ● Количество основных изменений, оцененных комитетом по изменениям (англ. <i>Change Advisory Board, CAB</i>); ● количество проведенных совещаний CAB; ● среднее время реализации RFC; ● отношение реализованных RFC к отвергнутым; ● количество чрезвычайных изменений, проверенных комитетом чрезвычайных изменений (<i>Emergency Change Advisory Board, ECAB</i>)
16.	<i>Change Evaluation</i> (Оценка изменений)	Оценка основных изменений услуг до их внедрения	<ul style="list-style-type: none"> ● Процент неавторизованных изменений; ● процент изменений, которые привели к инцидентам
17.	<i>Transition Planning and Support</i> (Переходное планирование и исполнение)	Обеспечивает планирование и исполнение проектов по внедрению модифицированных услуг в соответствии с заданными затратами, временными ограничениями и уровнем качества	<ul style="list-style-type: none"> ● Количество внедрений; ● процент проектов с первичным утверждением устава; ● количество изменений в уставах проектов в процессе реализации; ● относительное значение факта и плана затрат ресурсов; ● относительное превышение фактических сроков проектов к плановым

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
18.	<i>Service asset and configuration management</i> (Управление активами и конфигурациями)	Обеспечивает полную информацию о конфигурации компонент, необходимых для предоставления услуг, включая их взаимосвязи	<ul style="list-style-type: none"> ● Периодичность инвентаризационной проверки информации; ● количество инцидентов, связанных с некорректной информацией о конфигурации; ● средние трудозатраты на инвентаризацию; ● процент компонент, информация о которых хранится в системе управления конфигурациями (англ. <i>Configuration management data base, CMDB</i>); ● количество ошибок в <i>CMDB</i>, найденных в результате аудита и инвентаризации
19.	<i>Release and deployment management</i> (Управление релизами и поставками)	Планирование и контроль движения версий (англ. <i>leases</i>) на тестирование и установку для снижения рисков промышленной среды. Контроль корректности установки новых версий на промышленную среду, включая необходимые лицензии и компоненты инфраструктуры. Контроль качества в процессе разработки и установки	<ul style="list-style-type: none"> ● Количество установленных новых версий; ● среднее время установки новых версий после исправлений; ● количество версий, которые были возвращены после установки; количество новых версий, установленных автоматически
20.	<i>Service validation and testing</i> (Валидация и тестирование услуг)	Обеспечивает проверку новых или измененных услуг на соответствие ожиданиям заказчика и возможностям поддержки	<ul style="list-style-type: none"> ● Процент отвергнутых версий; ● количество выявленных при тестировании дефектов; ● среднее время исправления выявленных дефектов; ● количество инцидентов, связанных с новыми версиями

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
21.	<i>Knowledge management</i> (Управление знаниями)	Обеспечивает сохранение и полезное переиспользование накапливаемых в организации знаний для исключения необходимости их «пересоткрытия»	<ul style="list-style-type: none"> ● Количество разделов знаний, сохраненных за период; ● средняя частота использования накопленных разделов знаний
<i>Service operation</i> (Предоставление услуг)		Управление предоставлением услуг заказчиком, под которым подразумевается балансировка надежности услуг и их стоимости. Кроме процессов, книга включает четыре дополнительных функции	
22.	<i>Event Management</i> (Управление событиями)	Событие (англ. <i>event</i>) может означать, что какой-либо компонент услуги работает некорректно, т. е. присутствует инцидент (англ. <i>incident</i>). Однако событие может свидетельствовать и о нормальном прохождении процесса, например, выполнении процедур ежедневного резервного копирования данных. В любом случае событие несет информацию, которая должна быть обработана. Ее ценность информации не сводится к обнаружению инцидентов или проблем	<ul style="list-style-type: none"> ● Количество зарегистрированных событий по разным категориям; ● количество зарегистрированных событий различной критичности; ● количество или процент событий, требующих ручной обработки и наличие такой обработки; ● количество событий, связанных с инцидентами и изменениями; ● количество событий, в результате существующих проблем или известных ошибок; ● количество дублирующихся событий; ● количество событий, связанных с производительностью
23.	<i>Access Management</i> (Управление доступом)	Обеспечивает авторизованным пользователям доступ к использованию услуг и предупреждение доступа неавторизованных	<ul style="list-style-type: none"> ● Количество обработанных запросов доступа; ● количество доступов к отдельным услугам по различным департаментам, пользователям; ● количество инцидентов, требующих переустановки прав доступа; ● количество инцидентов, связанных с некорректной установкой доступа

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
24.	<i>Request Fulfilment</i> (Исполнение запросов)	Управляет исполнением запросов на обслуживание (англ. <i>Service Request, SR</i>), которые в основном являются стандартными, например, изменение пароля или предоставление информации. Стандартный запрос <i>SR</i> — предварительно согласованный, повторяющийся, безрисковый. Если эти критерии не совпадают, то запрос является запросом на изменение <i>RFC</i>	<ul style="list-style-type: none"> ● Количество выполненных запросов; ● количество отклоненных запросов; ● количество перенаправленных запросов
25.	<i>Problem Management</i> (Управление проблемами)	Обеспечивает устранение корневых причин инцидентов и предотвращает их повторение. Проблемой является неизвестная внутренняя причина одного или нескольких инцидентов, а «известная ошибка» является проблемой, которая была диагностирована и для которой было найдено временное обходное или постоянное решение. В рамках работы процесс обеспечивает регистрацию, классификацию, анализ проблем и выполненные мероприятия по проактивному предупреждению связанных инцидентов	<ul style="list-style-type: none"> ● Количество зарегистрированных проблем; ● процент проблем, решенных в установленные сроки; ● количество проблем не решенных в установленный срок; ● средняя стоимость устранения проблем; ● количество зарегистрированных, закрытых, открытых проблем за период; ● количество известных ошибок
26.	<i>Incident Management</i> (Управление инцидентами)	Обеспечивает восстановление услуг в соответствии с <i>SLA</i> в минимальный срок с минимизацией воздействия на бизнес. Инцидентом является незапланированное нарушение ИТ-услуги или снижение ее качества. Сбой инфраструктуры, который не привел к деградации услуги, также является инцидентом. Более широко: инцидент — это событие, не являющееся частью стандартного операционного процесса, которое привело или может привести к нарушению качества услуг	<ul style="list-style-type: none"> ● Количество зарегистрированных инцидентов; ● статистика по открытым, закрытым и зарегистрированным инцидентам за период; ● среднее время устранения инцидентов; ● процент инцидентов, устраненных в запланированное время (в соответствии с <i>SLA</i>); ● средний ущерб на инцидент; ● количество переоткрытых инцидентов; ● количество некорректно переназначенных инцидентов;

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
			<ul style="list-style-type: none"> ● количество инцидентов, некорректно категоризированных; ● количество инцидентов, закрытых на первой линии; ● количество инцидентов, разрешенных удаленно; ● статистика инцидентов по периодам рабочего дня для управления ресурсами
	Service Desk (функция)	Задачи: первичная обработка инцидентов и запросов, предоставление интерфейса заказчику к основным процессам, включая контроль жизненного цикла запросов, поддержку информированности заказчика	—
	Technical Management (Техническое управление — функция)	Обеспечивает контроль, проектирование и поддержку ИТ-инфраструктуры. Принимает участие в большинстве основных процессов, где затрагивается инфраструктура	—
	Application Management (Управление приложениями — функция)	Управляет качеством разработки и сопровождения прикладного ПО с основным акцентом на сбор и соблюдение потребностей бизнеса. Кроме того, обеспечивает полную сборку и обновления информации об имеющихся лицензиях с целью минимизации рисков, связанных с ростом затрат на ПО, перегрузкой человеческих ресурсов, с нарушением авторских прав	—

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
IT Operations management (Операционное управление — функция)		Мониторинг и контроль ИТ-услуг, инфраструктуры и рутинных операций на ежедневной основе	—
27	<i>Continual service improvement</i> (Непрерывное улучшение услуг)	<p>Обеспечивает непрерывное улучшение сервиса для заказчиков, повышение его эффективности и качества в соответствии с принципом цикла постоянного улучшения Деминга — Шухарта. Для обеспечения улучшения обеспечивает необходимый уровень контроля и замеров процессов.</p> <p>Жизненный цикл улучшений обычно состоит из 7 шагов:</p> <ol style="list-style-type: none"> 1) определение стратегии улучшений; 2) определение метрик; 3) сбор данных; 4) обработка данных; 5) анализ данных; 6) использование данных; 7) внедрение улучшений 	
28.	<i>Service Review</i> (Пересмотр услуг)	Пересмотр услуг на периодической основе для поиска возможностей повышения качества с принятием во внимание экономической целесообразности улучшений и продолжения предоставления услуг	<ul style="list-style-type: none"> ● Количество формально выполненных пересмотров услуг за период; ● количество выявленных недостатков услуг, которые могут стать источником инициатив по улучшению
29.	<i>Process Evaluation</i> (Оценка процессов)	Оценка процессов на регулярной основе на соответствие контрольным параметрам и для сравнения с конкурентами	<ul style="list-style-type: none"> ● Количество аудитов, оценок зрелости и контрольных сравнений с конкурентами, выполненных для процессов за период; ● количество выполненных оценок услуг; ● количество выявленных недостатков процессов; ● количество инициированных и выполненных улучшений процессов за период

Окончание табл. 13.1

№ п/п	Процесс	Описание	Ключевые показатели эффективности процесса (KPI)
30.	<i>Definition of CSI Initiatives</i> (Определение инициатив улучшений)	Определение инициатив улучшений на основе пересмотра услуг и оценки процессов	<ul style="list-style-type: none"> • Количество выявленных и обоснованных инициатив; • количество реализованных инициатив
31.	<i>Monitoring of CSI Initiatives</i> (Мониторинг инициатив улучшений)	Проверка выполнения инициатив в соответствии с утвержденными планами, включение необходимых метрик и контрольных точек	Количество инициатив улучшений под контролем

В библиотеке *ITIL* содержится следующая референсная организационная модель (рис. 13.2).

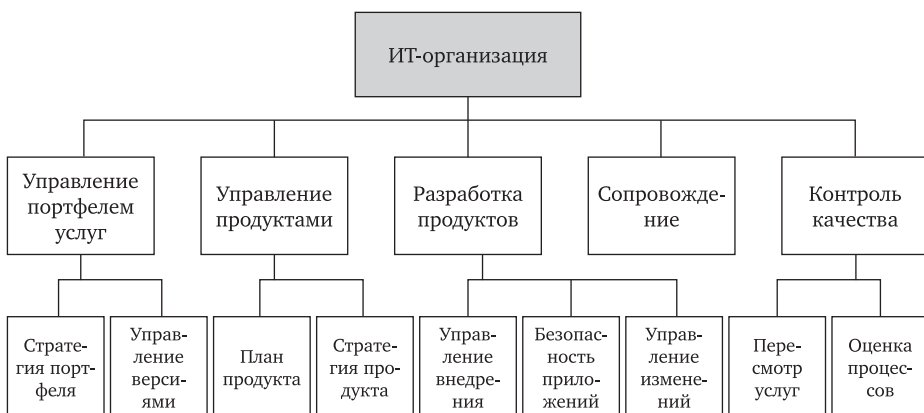


Рис. 13.2. Референсная организационная структура небольшой ИТ-компании

13.5. Метрики ИТ-процессов

В табл. 13.1 приведены ключевые показатели эффективности для различных процессов. Здесь рассмотрим основные требования к системе метрик и *KPI*.

Метрики и *KPI* не являются синонимами. Основное различие в том, что обычно метрики, являющиеся объективно собираемыми и рассчитываемыми количественными значениями, не имеют целевого значения, непосредственно не связаны с системой мотивации сотрудников и могут быть недоступны для контроля извне процесса (внешним наблюдателем). В то время как *KPI* строятся на основе метрик, но имеют целевое значение, часто встраиваются в систему мотивации сотрудников и могут быть объективно проконтролированы внешним по отношению к процессу наблюдателем.

Метрики могут собираться внутри процесса для внутреннего или внешнего контроля его исполнения и поиска возможности улучшений. Набор метрик обычно шире, чем набор *KPI*.

Целью сбора метрик является определение текущего уровня эффективности процессов, включая оценку удовлетворенности потребителей его результатов (а также взаимодействующих процессов). В соответствии с общей теорией качества нельзя улучшить то, что нельзя измерить. Это определяет значение метрик в развитии ИТ-услуг.

Набор метрик должен в основном соответствовать целям бизнеса, целям развития ИТ-услуг и целям процессов. Если внедряются метрики, не привязанные к целям основной деятельности, это может привести к негативным результатам в связи с искажением целей. Примером подобных метрик является внедрение «палочных» показателей в дея-

тельности правоохранительных органов, когда их эффективность измерялась количеством выявленных правонарушений. Тот же результат получим, если подобные метрики будут внедрены в работе внутренней службы безопасности банка или службы операционных рисков — полезная деятельность организации будет в значительной степени заблокирована в связи с непрерывной «охотой на ведьм». Корректной метрикой деятельности для подобных служб является, например, размер ущерба, полученного банком за отчетный период в результате инцидентов безопасности или операционных сбоев. Деятельность этих служб должна быть направлена в итоге на снижение ущерба в рамках своей компетенции, а не на поиск «воображаемых» угроз или завышенных рисков.

Если процессы недостаточно развиты, то сбор объективных количественных метрик затруднен — процессы не имеют контрольных точек. Если процессы не носят повторяющийся характер (на низких уровнях зрелости), то метрики либо невозможно собрать, либо они не имеют информационной ценности.

Часть метрик может представлять собой «сырые» данные, непосредственно собираемые, например, системами мониторинга. Основное количество метрик является расчетными на основе первичных данных. Они более репрезентативны и облегчают анализ.

Оценка процесса или подразделения на соответствие модели зрелости *СММІ* является допустимой и ценной метрикой, поскольку обеспечивает возможность сравнения эффективности деятельности подразделений или процессов с разнородными функциями и целями. Единственной альтернативной сравнительной метрикой является опрос удовлетворенности заказчиков или менеджеров взаимодействующих процессов, например, по шкале Лайкерта¹.

Внедрение метрик часто положительно изменяет поведение сотрудников само по себе, даже без привязки к системе мотивации. Например, в рамках одного из крупных проектов, реализованных автором, группой тестирования была предоставлена информация о значительной разнице в эффективности деятельности разработчиков. Руководством проекта было принято решение о еженедельной публикации в свободном доступе на внутреннем портале двух показателей для каждого разработчика: 1) количество разработанных и предоставленных на тестирование функций; 2) количество найденных ошибок в предоставленном на тестирование ПО.

В результате регулярного раскрытия информации повысилась эффективность деятельности всех разработчиков — «осторожные» разработчики, поставлявшие более качественное программное обеспечение, но с меньшей скоростью (англ. *time to market*), нарастили скорость поставки без потери качества. «Торопливые» разработчики, поставлявшие обновления с высокой скоростью, но большим количеством дефектов (что приводило к частым возвратам на исправление), значительно

¹ Орлов А. Организационно-экономическое моделирование. Экспертные оценки.

повысили качество поставок практически без потери скорости. Здесь обратите внимание на то, что было внедрено две встречные взаимодополняющие метрики. По отдельности они не являются репрезентативными и не привели бы к повышению эффективности.

Кроме того, приведенный пример иллюстрирует требование объективного сбора данных и расчета метрик незаинтересованной стороной. Показатели характеризовали деятельность разработчиков, но контролировались группой тестирования.

Часто допускаемой ошибкой при формировании набора метрик является их использование для «наказания», а не мотивации сотрудников. Неэффективность подобного подхода многократно доказывалась на практике¹.

Классический пример негативного эффекта от внедрения «карающих» систем метрик — так называемая итальянская забастовка, которая заключается в предельно строгом исполнении сотрудниками своих должностных обязанностей и правил, что обычно приводит к резкому снижению продуктивности или блокировке деятельности в связи с наличием процессных несоответствий на любом предприятии или в рамках его внешнего взаимодействия.

При разработке и внедрении системы метрик необходимо соблюдать баланс между их достаточностью для контроля эффективности процессов и возможностью рассчитывать большое количество показателей, что бывает достаточно трудоемко.

Еще одна балансировка требуется при работе с системой метрик в ходе ее развития. Невозможно за один шаг разработать и внедрить эффективную систему метрик. Она должна совершенствоваться и изменяться непрерывно в соответствии с изменением стратегических целей. С другой стороны, если метрики начинают изменяться в рамках отчетного периода, то их эффективность резко снижается — они перестают служить ориентирами в деятельности сотрудников. Таким образом, метрики должны изменяться, но быть достаточно стабильными.

Различные метрики могут предоставляться различным аудиториям. Не все метрики должны быть общедоступны. Обычно аудитории для предоставления метрик подразделяются на следующие группы: клиенты, пользователи, сотрудники ИТ, руководство. Например, если мы намерены информировать клиентов о росте качества и надежности наших ИТ-услуг, то нежелательно предоставление в открытом доступе информации непосредственно о количестве зарегистрированных инцидентов, даже если оно снижается из года в год, но при этом допустимо размещение информации о процентной динамике снижения количества инцидентов.

Ошибочное мнение о том, что отдельные процессы или подразделения не поддаются внедрению метрик. Обычно такая ошибка допускается в отношении контролируемых подразделений, например, финансового

¹ Брукс П. Метрики для управления ИТ-услугами. М. : Альпина Бизнес Букс, 2008.

мониторинга. Любая деятельность поддается внедрению эффективных метрик, например, для приведенного подразделения могут быть внедрены следующие метрики: средняя задержка в предоставлении финотчета (любая деятельность должна быть плановой и иметь устойчивый повторяющийся характер), степень достоверности финпрогноза за период (достоверность финансового планирования должна расти со временем) и др.¹

В заключение раздела отметим, что обычно попытки внедрения системы *KPI* в организациях, не имеющих сколько-нибудь развитой и опробованной системы метрик, не бывают успешными или занимают значительное время. Следует помнить о том, что система *KPI* базируется на эффективной системе метрик.

Для более широкого рассмотрения темы метрик и *KPI* рекомендуем обратиться к работе П. Брукса «Метрики для управления ИТ-услугами».

13.6. Соглашение об уровне услуг — *SLA*

Соглашение об уровне услуг (англ. *Service Level Agreement, SLA*) является одним из ключевых понятий *ITIL*. Жизненный цикл каждой ИТ-услуги включает составление *SLA*. Что это такое?

SLA — это договор между заказчиком и провайдером услуг, который может фиксироваться как в виде непосредственно договора в случае, если заказчик и провайдер находятся в разных организациях, так и в виде внутреннего регламента, если они представляют одну организацию.

SLA составляется для каждой предоставляемой услуги и в любом случае имеет ряд обязательных пунктов (табл. 13.2).

Таблица 13.2

Структура *SLA*

Раздел	Описание
1. Стороны и сроки действия	Указываются заказчик и провайдер услуг, а также ответственные лица, подтверждающие своей визой заключение соглашения
2. Полное описание услуг	Приводится описание предоставляемой услуги на языке заказчика, например, «автоматизированная обработка конверсионных сделок с формированием требований и обязательств по сделкам для передачи в расчетную систему». Здесь же указываются параметры взаимодействия с <i>service desk</i> провайдера услуг (это часть услуги). Следует обратить внимание на то, что <i>ITIL</i> (за очень редким исключением) не предусматривает предоставление услуги в виде конкретного оборудования или системы. Услуга не должна формулироваться как, например, «предоставление принтера <i>HP Laser Jet 1020</i> »

¹ Брукс П. Метрики для управления ИТ-услугами.

Раздел	Описание
3. Стоимость услуги	ITIL предусматривает обязательную оценку стоимости предоставляемых услуг, даже в случае взаимодействия внутри одной организации. В разделе приводится стоимость услуги за период или за экземпляр предоставленной услуги
4. Количественные характеристики качества услуг	Обязательно приводятся количественные объективно измеряемые и протоколируемые метрики качества услуг, например, «черная печать формата А4 до 30 страниц в минуту, с разрешением до 300 точек на дюйм»
5. Временные характеристики предоставления услуг	Указывается временной режим предоставления услуги в течение года, месяца, недели и рабочего дня. Например, «автоматическая обработка конверсионных операций обеспечивается с 8-00 до 21-00 по дням, официально объявленным Московской биржей рабочими
6. Число и размещение пользователей	Поскольку услуга предоставляется заказчику в лице потребителей услуги, то необходимо точно указать их количество и размещение, включая адрес и комнату. Детализация адреса важна в том числе в связи с необходимостью обеспечить доступ представителей провайдера услуг к рабочему месту потребителя (пользователя) услуги в случае сбоя или для обучения. В случае массового обслуживания или отсутствия привязки к физическому расположению пользователя детальная информация может быть исключена, но это должно быть дополнительно оговорено
7. Предоставление услуг в нештатных ситуациях	В зависимости от уровня воздействия нештатной ситуации может быть предусмотрено несколько уровней предоставления услуги. Например, «в случае недоступности помещения по адресу... услуга обработки конверсионных сделок может быть предоставлена по адресу резервного дилингового зала по адресу ..., при этом количество доступных рабочих мест операторов снижается до ... В случае разрыва канала связи с основным вычислительным центром обеспечивается переход на резервный вычислительный центр, расположенный в г. Волгограде, и скорость обработки снижается до ... количества операций в час»
8. Ответственность провайдера	Поскольку услуги предоставляются на платной основе, то здесь приводится информация о размере компенсации со стороны провайдера в случае нарушения нормального предоставления услуг в соответствии с разделом 7
9. Ответственность потребителя	Потребитель сознательно или неосознанно может своими действиями нанести ущерб материалам, предоставленным ему для оказания услуги, например, залить кофе предоставленный принтер. Здесь приводится сумма возмещения возможного ущерба, нанесенного потребителем провайдеру или порядок принятия решения о таком возмещении

Раздел	Описание
10. Процедуры улучшения соглашения	Невозможно разработать сразу идеальное соглашение. Это крайне сложная задача, особенно если учесть разнообразие и количество предоставляемых внутри организации ИТ-услуг. Система <i>SLA</i> должна поступательно развиваться в соответствии с набором процессов <i>Continual service improvement</i> (непрерывное улучшение услуг) и циклом постоянного улучшения. В разделе приводится порядок и периодичность пересмотра соглашения
11. Терминология	Услуга описывается на языке, понятном заказчику. Данная терминология может быть незнакома сотрудникам провайдера. Возможна и обратная ситуация, когда, например, при описании аварийных ситуаций используются технические термины. В разделе приводится описание основных используемых терминов для исключения двойного толкования

Глава 14

УПРАВЛЕНИЕ НЕПРЕРЫВНОСТЬЮ, РИСКАМИ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТЬЮ

Темы управления непрерывностью, рисками и информационной безопасностью часто рассматриваются обособленно, что совершенно некорректно.

Стандарт ГОСТ ИСО 20 000-1 в разделе 6.3 «Управление непрерывностью и доступностью услуг» однозначно указывает на необходимость оценки и документирования рисков. Тот же стандарт в разделе 6.6 «Управление информационной безопасностью» указывает на обязательное управление рисками.

То есть взаимная увязка непрерывности, анализа рисков и информационной безопасности не только является «лучшей практикой» — она стандартна на международном уровне. При этом на практике в большинстве случаев сталкиваемся с тем, что, например, контрмера «электронная подпись» применяется не там, где выявлен и количественно оценен как «высокий» риск получения ущерба в результате отказа от авторства или внедрения ложных сообщений в результате реализации угрозы «маскарад» со стороны внутренних нарушителей¹, а везде, где, по мнению службы защиты информации, могут быть внедрены соответствующие программно-технические средства.

Автору неоднократно приходилось встречаться с ситуацией, когда заказчик или служба защиты информации необоснованно требовали внедрить средства электронной подписи в канал автоматического обмена данными между прикладными системами, расположенными во внутренней сети. При этом в одном из характерных случаев сотрудник, необоснованно настаивающий на внедрении соответствующих средств, оказался в группе, проходившей обучение у автора по данной теме. В рамках практической работы специалист выполнил количественный расчет рисков, который показал крайне низкое значение соответствующего риска, после чего требование внедрить электронную подпись было снято.

Любая контрмера является затратной как с точки зрения прямых расходов, так и с точки зрения косвенных затрат, например, на повышение мощности оборудования. Электронная подпись накладывает

¹ Полная формулировка риска.

очень высокие требования на производительность оборудования; это приводит к тому, что для высокопроизводительных систем, например, систем биржевой торговли от нее отказываются, принимая соответствующие риски.

Часто стоимость внедрения контрмер значительно превосходит не только величину возможного ущерба, но и общие затраты по проекту. Например, в одном из случаев общие затраты на закупку и внедрение системы составили около 5 млн руб. При этом после дополнения задания требованиями по информационной безопасности сумма проекта выросла на 15 млн руб.

Каждая контрмера должна быть обоснована путем количественного анализа соответствующих рисков, а затраты на ее применение сбалансированы относительно предотвращаемого вероятного ущерба. Ключевое слово здесь — «вероятного». Например, ущерб от ядерной бомбардировки вычислительного центра банка является крайне высоким, но, учитывая ее относительно низкую вероятность, большинство банков воздерживаются от построения противоатомных бункеров и систем противоракетной обороны.

При высоких затратах на реализацию контрмер и относительно низкой оценке риска он должен приниматься. Международными стандартами *ISO 20 000* и *ISO 27 000* («Системы менеджмента информационной безопасности») предусмотрен обязательный порядок принятия рисков, который не возможен без его количественного анализа.

Во многом аналогичная ситуация складывается с аварийным планированием и обеспечением непрерывности деятельности. Типовой является ситуация, когда выдвигаются завышенные требования к времени восстановления ИТ-сервисов без проведения соответствующего анализа рисков (англ. *Business Impact Analysis, BIA*). Например, по торгово-информационным сервисам обычной практикой является требование восстановления при аварии в течение 30 минут. При этом для крупного банка это время после квалифицированного анализа рисков обычно увеличивается до 3—4 часов. Разница в стоимости реализации проекта технического резервирования торгового комплекса банка между вариантами 30 минут и 3 часа может достигать нескольких миллионов долларов.

14.1. Аварийное планирование

Основной смысл аварийного планирования заключается в том, чтобы в случае реализации редких катастрофических масштабных событий оперативно обеспечить выполнение мероприятий, позволяющих снизить потери организации и ускорить восстановление ее нормальной деятельности.

Под **аварией** будем понимать редкие события катастрофического и чрезвычайного характера, крупные технические аварии.

Хорошую иллюстрацию предоставляет сравнительная диаграмма деятельности организаций в случае наступления аварийной ситуации (АС), одна из которых имеет аварийный план, а вторая таким планом не располагает (рис. 14.1).

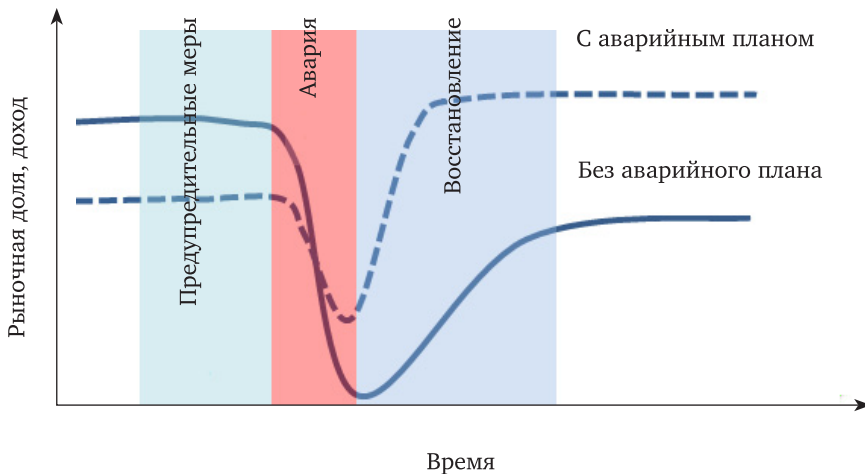


Рис. 14.1. Сравнительные показатели деятельности организаций с наличием и отсутствием аварийного плана

Схема демонстрирует основные принципы аварийного планирования:

- организация, имеющая аварийный план, в меньшей степени несет потери в эффективности деятельности и других показателях при наступлении аварии;
- организация, имеющая аварийный план, быстрее начинает выходить из аварийной ситуации;
- организация, имеющая аварийный план, быстрее восстанавливает нормальную деятельность и располагает конкурентным преимуществом по отношению к тем, кто не располагает аварийным планом.

В терминах Н. Талеба¹ организация с аварийным планом — «антихрупкая» и способна повернуть масштабные катастрофические события себе на пользу.

На рис. 14.1 явно представлены три периода: период выполнения предупредительных мероприятий, снижающих потенциальное воздействие аварии, период аварии и период восстановления. Аварийное планирование должно приоритетно касаться периода аварии. При этом в соответствии с рядом стандартов аварийные планы должны включать информацию о периодическом тестировании, что увеличивает объем непосредственно планов и с точки зрения автора не является вполне обоснованным — это предмет внутренних регулирующих документов. То же относится и к периоду восстановления после аварии — обычно для выработки мер по восстановлению требуется провести деталь-

¹ Талеб Н. Антихрупкость.

ный анализ последствий и спланировать отдельно необходимые меры по восстановлению (на это достаточно времени и могут быть подключены дополнительные эксперты и руководители).

В аварийном планировании приходится сталкиваться с рядом типичных ошибок. Основной является возложение функции старта аварийного плана на некий коллегиальный орган или высшее должностное лицо. Аварийные ситуации принципиально краткосрочны и требуют максимально быстрой (желательно автоматической) реакции. Представьте, что включение системы пожаротушения было бы возложено на председателя банка. Срок непоправимых последствий для жизни и здоровья людей в случае пожара обычно составляет минуты и секунды. Скорость задействования высоких должностных лиц составляет десятки минут в лучшем случае. То же относится ко всем основным видам чрезвычайных ситуаций.

Таким образом, классификация и первичное реагирование на аварийные ситуации должны быть возложены на специализированных дежурных. Хорошим примером данного подхода является армия. Военные постоянно имеют дело с чрезвычайными ситуациями и давно возложили функции первичного реагирования на исполнительский уровень. В случае пожара дежурный прапорщик имеет право отдать указание генералам генштаба переместиться из опасного помещения.

Второй типичной ошибкой является подготовка объемных детальных классификаторов аварийных ситуаций и аварийных планов. Нередко объем подобных документов достигает нескольких сотен страниц. В случае чрезвычайной ситуации дежурный должен принять обоснованное решение о старте определенного аварийного плана за секунды и минуты. Он должен сделать это обоснованно и не боясь последствий для себя. Ему некогда изучать сотни страниц. Таблица, которая позволит ему быстро и обоснованно принять решение (в случае отсутствия соответствующей автоматической системы) не должна превышать листа формата А 4, заполненного достаточно крупным шрифтом. Для такого «укрупненного» представления придется свести все разнообразие чрезвычайных ситуаций к обобщенным типам. Пример подобного классификатора приведен в табл. 14.1.

Каждая ячейка таблицы, по сути, является своеобразной кнопкой, стартующей отдельный аварийный план.

Приведенный классификатор не идеален, но имеет ряд полезных особенностей.

Основное достоинство — классификатор достаточно лаконичный и доступен для быстрого прочтения. Разнообразные сценарии чрезвычайных ситуаций сведены в девять строк по типам последствий для отдельных типов активов, клиентов и сотрудников.

Второй особенностью является наличие временной шкалы. Аварийные сценарии имеют различную критичность в зависимости от времени возникновения.

Классификация аварийных ситуаций

Ресурсы под угрозой	Уровень последствий		
	Низкий — нет нарушения критических бизнес-операций, минимальные нарушения других бизнес-процессов	Средний — присутствует временное нарушение основных бизнес-процессов не в рамках нижеуказанного периода времени	Высокий — присутствует нарушение основных бизнес-процессов в рамках нижеуказанного периода времени
Телекоммуникационные каналы	Потеря до 1 часа доступа к сети Биржи, сети Интернет, телефонной связи	Потеря до 2 часов доступа к сети Биржи, к сети Интернет, телефонной связи	Потеря доступа к сети Биржи, к сети Интернет, телефонной связи основного здания и основного вычислительного центра
Аппаратные ресурсы критичных АС	Устранимые в течение 2 часов нарушения работы	Устранимые в течение 2 часов нарушения работы	Неустраняемые нарушения работы основного вычислительного центра
Целостность данных критичных АС	Устранимая в течение 2 часов потеря данных	Устранимая в течение 2 часов потеря данных	Восстановление данных возможно только с привлечением сторонних источников
Конфиденциальность данных критичных АС	Угроза распространения и коммерческого использования отсутствует	Присутствует угроза коммерческого использования, угроза широкого разглашения, использования конкурентами отсутствует	Присутствует угроза широкого разглашения, использования конкурентами
Целостность архива документов	Присутствуют копии документов	Неустраняемая частичная потеря документов, не используемых в оперативной деятельности	Масштабная потеря документов, используемых в операционной деятельности
Клиенты	Нарушение финансовых интересов клиентов отсутствует	Нарушение финансовых интересов отдельных клиентов	Масштабные нарушения финансовых интересов клиентов

Ресурсы под угрозой	Уровень последствий		
	Низкий — нет нарушения критических бизнес-операций, минимальные нарушения других бизнес-процессов	Средний — присутствует временное нарушение основных бизнес-процессов не в рамках нижеуказанного периода времени	Высокий — присутствует нарушение основных бизнес-процессов в рамках нижеуказанного периода времени
Сотрудники	Отсутствуют ранения и угроза здоровью сотрудников, минимальный стресс	Присутствуют ранения и угроза здоровью сотрудников, нет угрозы жизни	Присутствует угроза жизни сотрудников
Электропитание ВЦ или рабочих помещений	Отключение электропитания на период не более 2 часов	Отключение электропитания на период не более 2 часов	Отключение электропитания основного здания и вычислительного центра на непрогнозируемый период времени
Помещения	Устранимые разрушения, перемещение сотрудников не требуется	Разрушение отдельных помещений, операционная деятельность может быть восстановлена в течение 2 часов путем перемещения сотрудников внутри основного здания	Масштабные разрушения основного здания
Период времени возникновения	С 10-00 до 16-00	С 8-00 до 10-00 или с 16-00 до 22-00	С 8-00 до 10-00 или с 16-00 до 22-00

Возвращаясь к теме типичных ошибок аварийного планирования, необходимо отметить часто встречающееся ошибочное отсутствие включения в аварийные планы информирования заинтересованных лиц об авариях различного уровня критичности. Более того, при авариях часто скрывают чувствительную информацию от клиентов, руководства и контролирующих органов. На практике нередко приходится сталкиваться со случаями, когда руководитель организации узнает о том, что его компания не предоставляет необходимые услуги в течение целого дня, от крупнейших клиентов или регулятора, что, безусловно, не допустимо и часто стоит карьеры.

Следует отметить, что в случае предоставления услуг внешним клиентам, особенно в массовом корпоративном или ритейл-сегменте, информирование клиентов о наступлении аварийной ситуации определенного типа в организации-провайдере является строго обязательным и часто предусматривается договорными обязательствами или действующим законодательством.

Более подробно информация по теме аварийного планирования и обеспечения непрерывности деятельности представлена в специализированной литературе¹.

14.2. Анализ рисков

Обычно тема «Анализ рисков» рассматривается в рамках более широкой — «Управление рисками». В данном параграфе не будем повторять материал, изложенный в гл. 7, — в основном рассматриваемые вопросы и подходы аналогичны. При этом методология анализа рисков к настоящему времени в большей степени проработана и отлажена именно для вопросов информационной безопасности и может быть в значительной степени заимствована смежными областями деятельности.

Методик анализа рисков в рамках информационной безопасности к настоящему времени разработано достаточно много. Они отличаются моделью расчета рисков (двухфакторная, трехфакторная), форматом результирующего отчета, списками угроз и уязвимостей, подходами к расчету величины возможного ущерба, методами проведения опроса и сбора статистики. Их основное сходство заключается в том, что все они направлены на получение максимально точного повторяемого количественного результата — величины риска для различных угроз. Этот результат позволяет обоснованно выбирать приоритеты защиты, вырабатывать эффективные и сбалансированные с точки зрения стоимости контрмеры, позволяющие снизить совокупный риск или застраховать его, на основе объективных расчетных данных и принятых политик провести процедуры принятия риска.

¹ *Rima Ch. Business Continuity and Disaster Recovery Planning for IT Professionals, Syngress, 2014.*

Без проведения расчета риска мы вынуждены предпринимать дорогостоящие контрмеры на основе предыдущего своего или чужого опыта, интуиции, подсказки которой часто бывают ошибочными.

Для демонстрации склонности интуиции к ошибкам в части анализа рисков попробуйте ответить на вопрос — кто в крупном банке должен иметь более широкие права доступа к информации и автоматизированным системам: операционист бэк-офиса (обработка операций и контроль взаиморасчетов) или председатель банка?¹

В рамках длительной практики преподавания темы анализа рисков (более 12 лет, свыше 2000 слушателей) и проведения практических работ по расчету автор на систематической основе собирал результат сравнения первичной интуитивной оценки максимального риска (выбор приоритета защиты: целостность, конфиденциальность, доступность) и последующего расчета по методике *CRAMM*.

В абсолютном большинстве случаев (более 90 %) первичная оценка давала ошибочный результат. При этом в состав слушателей курса МБА и второго высшего образования в Банковском институте «Высшей школы экономики» входило большое количество высших должностных лиц банков, сотрудников информационной безопасности, ИТ-специалистов. Это означает, что при отсутствии в организации систематического применения методик количественного расчета рисков большинство контрмер на практике предпринимается недостаточно обоснованно, путем избыточного покрытия всех возможных уязвимостей. При этом вполне возможна ситуация, когда наиболее высокие риски оказываются без достаточного внимания.

Как было сказано выше, существует достаточно большое количество методик анализа рисков. Под анализом понимается выявление и оценка величины риска. Рассмотрим возможности анализа рисков на основе широко применяемой и, возможно, наиболее развитой методики *CRAMM* (от англ. *CCTA Risk Analysis and Management Method*).

Данная методика была разработана в 1987 г. (примерно в одно время с *ITIL*) Центральным агентством по вычислительной технике и телекоммуникациям Великобритании (англ. *Central Computer and Telecommunications Agency, CCTA*) и в дальнейшем получила развитие путем выделения специальных профилей для государственных, коммерческих и финансовых организаций. В настоящее время пятая версия методики применяется, кроме правительства Великобритании, например, Северо-Атлантическим Альянсом, армией Нидерландов, компанией «Сименс» и многими другими.

Работы по методике проводятся в три основных этапа:

- на первом этапе описывается обследуемый объект и определяются его границы;

¹ Правильный ответ — операционист. Председатель банка не нуждается в правах доступа к продуктивным системам. Часто высшие должностные лица становятся основным источником утечки данных о параметрах доступа в критичные системы.

- на втором этапе производится количественная оценка возможного ущерба информационным активам организации и оценка рисков по трехфакторной модели;
- на третьем этапе вырабатываются контрмеры по крупнейшим оцененным рискам.

Работа методики основывается на использовании специализированного ПО, которое позволяет обеспечить сбор информации, последовательное заполнение опросников и подготовку итоговых отчетов¹.

Остановимся подробнее на втором этапе.

Под **угрозой** понимается действие или событие, способное нанести ущерб безопасности.

Под **уязвимостью** понимается слабое место актива или меры и средства контроля и управления, которое может быть использовано угрозой.

Эти определения соответствуют международному стандарту *ISO 27 000*, который был принят в России в качестве государственного².

Списки угроз и уязвимостей совпадают. Фрагмент списка угроз *CRAMM* приведен в табл. 14.2. Как можно заметить, приведенный список угроз покрывает практически все типы операционных рисков. Поэтому *CRAMM* вполне может быть рекомендована для обеспечения работ по управлению операционными рисками.

Таблица 14.2

Угрозы в методе *CRAMM*

№ п/п	Угрозы в методе <i>CRAMM</i>
1.	Использование чужого идентификатора сотрудниками организации («маскарад»)
2.	Использование чужого идентификатора поставщиками услуг («маскарад»)
3.	Использование чужого идентификатора посторонними («маскарад»)
4.	Несанкционированный доступ к приложению
5.	Внедрение вредоносного ПО
6.	Использование телекоммуникаций для несанкционированного доступа сотрудниками организации
7.	Ошибки при маршрутизации
8.	Неисправность сетевого сервера
9.	Неисправность накопительного устройства

¹ Описание методики и терминология сознательно упрощены. Более подробно вопросы информационной безопасности рассматриваются, например, в работе: Щеглов А. Ю., Щеглов К. А. Защита информации: основы теории : учебник для бакалавриата и магистратуры. М. : Юрайт, 2018.

² ГОСТ Р ИСО/МЭК 27000-2012 Информационная технология (ИТ). Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности, 2013.

№ п/п	Угрозы в методе CRAMM
10.	Неисправность печатающих устройств
11.	Неисправность сетевых шлюзов
12.	Неисправность средств сетевого управления или управляющих серверов
13.	Неисправность электропитания
14.	Неисправность кондиционеров
15.	Сбои системного и сетевого ПО
16.	Сбои прикладного ПО
17.	Ошибки операторов
18.	Ошибки при профилактических работах с оборудованием
19.	Ошибки пользователей
20.	Затопление
21.	Природные катаклизмы
22.	Нехватка персонала
23.	Кражи со стороны посторонних
24.	Преднамеренное вредительство со стороны сотрудников
25.	Терроризм

Риск в методике рассчитывается как произведение трех величин: вероятности реализации угрозы, вероятности реализации уязвимости и оценки величины возможного ущерба.

Для понимания соотношения угроз и уязвимостей рассмотрим пример.

Пример 14.1

Пусть вычислительный центр организации располагается в одном здании с прачечной. Прачечная с вероятностью 100 % подвергается затоплению раз в год в результате аварий различного рода. В результате затопления вычислительного центра основной сервер организации выходит из строя с вероятностью 50 %.

В данном примере вероятность реализации угрозы «затопление» равна 100 % (или 1), вероятность реализации уязвимости «затопление» равна 50 % (или 0,5).

Основным результатом работы методики является заполненная карта рисков (рис. 14.2)¹.

¹ Информация о компонентах методики не приводится полностью из-за ее объема и в связи с необходимостью защиты авторских прав.

$$\text{РИСК} = \text{Р угрозы} \cdot \text{Р уязвимости} \cdot \text{Ущерб}$$

Угрозы безопасности		15 М	1 Н	3 Н	12 Н	← Типы физического ущерба
		«Маскарад» со стороны внутренних нарушителей				
	Угроза	1	1	1	1	← Оценка вероятности угрозы
	Уязвимость	2	2	2	2	← Оценка вероятности уязвимости
	Ущерб	1	1	1	3	← Оценка ущерба
	Величина риска	2	2	2	6	← Оценка риска

Рис. 14.2. Фрагмент карты рисков в методике **CRAMM**

Более сложным является определение ущерба. В *CRAMM* ущерб определяется в баллах для приведения к единой шкале различных видов ущерба. При этом используется три шкалы, облегчающие экспертам определение размера ущерба (K_1 , K_2 , K_3). На рис. 14.3 приведены фрагменты шкал.

Ущерб определяется по каждой шкале в баллах от 1 до 10. При этом финансовый ущерб может накладываться на репутационный и конкурентный. Следовательно, оценки по разным шкалам должны суммироваться.

Следует обратить внимание на то, что ущерб не является результатом реализации угрозы непосредственно. Внутренний или внешний нарушитель может проникнуть в систему с использованием чужого логина («маскарад»), при этом не нанеся ущерба ее доступности, конфиденциальности и целостности или нанеся незначительный ущерб. Поэтому в методике предусмотрена детализация возможного физического ущерба, по отношению к которому уже и производится оценка ущерба в баллах. В графе типов физического ущерба карты рисков (см. рис. 14.2) указан неполный список последствий реализации угроз или типов физического ущерба, в расширенном составе приведенный в табл. 14.3.

Таблица 14.3

Последствия реализации угроз (типы физического ущерба) *CRAMM* (фрагмент)

Обозначение	Последствия
15 М	Недоступность — менее 15 минут
1 Н	Недоступность — 1 час
12 Н	Недоступность — 12 часов
1 D	Недоступность — 1 день
1 W	Недоступность — 1 неделя
1 M	Недоступность — 1 месяц

Обозначение	Последствия
<i>B</i>	Разрушение данных с момента последнего резервного копирования
<i>T</i>	Разрушение всех данных, включая резервные копии
<i>I</i>	Несанкционированное раскрытие данных внутренними нарушителями
<i>C</i>	Несанкционированное раскрытие данных провайдером услуг
<i>O</i>	Несанкционированное раскрытие данных внешними нарушителями
<i>SE</i>	Небольшие ошибки пользователей и администраторов
<i>WE</i>	Широкомасштабные ошибки пользователей и администраторов
<i>In</i>	Внедрение ложных сообщений
<i>Or</i>	Отказ от подтверждения авторства сообщения
<i>Mr</i>	Ошибки маршрутизации

В табл. 14.4 представлен фрагмент шкал определения ущерба в методике *CRAMM*.

Таблица 14.4

Определение ущерба в методике *CRAMM*

Оценка ущерба	Описание ущерба
K₁ — ущерб коммерческим (экономическим) интересам организации	
1	Представляет интерес для конкурентов, но не приносит коммерческой выгоды
2	Представляет интерес для конкурентов и приносит коммерческую выгоду на сумму менее 15 тыс. долл.
3	Представляет интерес для конкурентов и приносит коммерческую выгоду на сумму от 15 тыс. до 150 тыс. долл. или приводит к финансовым потерям либо потере платежеспособности
...	...
K₂ — финансовые потери либо дезорганизация деятельности	
1	Финансовые потери в размере менее 15 тыс. долл.
2	Финансовые потери в размере от 15 тыс. до 50 тыс. долл.
3	Финансовые потери в размере от 50 тыс. до 150 тыс. долл.
...	...
K₃ — ущерб репутации организации	
1	Нет
2	Локальные затруднения внутри организации
3	Окажет негативное влияние на отношения с клиентами, коммерческими или государственными организациями и приведет к нежелательной известности на локальном (региональном) уровне
...	...

Оценка вероятностей реализации угроз и уязвимостей по методике производится путем заполнения специальных вопросников (рис. 14.4). Работа по методике является коллективной, в ней принимают участие разные специалисты организации. Сотрудники бизнес-подразделений оценивают величину ущерба, сотрудники службы информационной безопасности и ИТ-специалисты отвечают на вопросы, касающиеся количества пользователей, статистики инцидентов, и технические вопросы.

После завершения расчета рисков по всей карте производится выбор максимальных значений рисков и выработка контрмер по отношению к ним.

Для точного позиционирования контрмер необходимо сохранять по проектным документам точное и полное наименование риска с указанием угрозы и последствий ее реализации (типа физического ущерба), например, «риск с величиной N в результате реализации угрозы “маскарад”, которая привела к потере данных с момента последнего копирования». В этом случае появляется возможность вырабатывать контрмеры как для снижения вероятности реализации угрозы или снижения уязвимости, так и воздействовать на возможные последствия, минимизируя получаемый ущерб.

Оценка вероятности угроз

Угроза: Использование чужого идентификатора сотрудниками организации (“маскарад”)

Ответьте на вопросы

1. Сколько раз за последние 3 года сотрудники организации пытались с использованием прав других пользователей получить несанкционированный доступ к хранящейся в системе (сети) информации?

Варианты ответов

- | | | |
|---|----------------------------------|----|
| a | Ни разу | 0 |
| b | Один или два раза | 10 |
| c | В среднем раз в год | 20 |
| d | В среднем чаще одного раза в год | 30 |
| e | Неизвестно | 10 |

10. Сколько раз за последние 3 года сотрудники пытались получить несанкционированный доступ к информации, хранящейся в других подобных системах в вашей организации?.....

Степень угрозы при количестве баллов:

- | | |
|-------------|---------------|
| До 9 | Очень низкая |
| От 10 до 19 | Низкая |
| От 20 до 29 | Средняя |
| От 30 до 39 | Высокая |
| 40 и более | Очень высокая |

Оценка уязвимостей

Угроза: Использование чужого идентификатора сотрудниками организации (“маскарад”)

Ответьте на вопросы:

1. Сколько людей имеют право пользоваться системой (сетью)?

Варианты ответов

- | | | |
|---|----------------|----|
| a | От 1 до 10 | 0 |
| b | От 11 до 50 | 4 |
| c | От 51 до 200 | 10 |
| d | От 200 до 1000 | 14 |
| e | Свыше 1000 | 20 |

Степень уязвимости при количестве баллов:

- | | |
|-------------|---------|
| До 9 | Низкая |
| От 10 до 19 | Средняя |
| 20 и более | Высокая |

Рис. 14.4. Фрагменты опросников для оценки вероятности угроз и уязвимостей в методике CRAMM

Проверочные тесты к разделу III

1. В соответствии с моделью зрелости *СММІ* уровень оптимизируемости предполагает:

- а) документированность, количественную оценку эффективности ИТ, наличие и выполнение задач по оптимизации процессов ИТ;
- б) оптимизацию распределения персональной ответственности специалистов ИТ и функциональных сотрудников за отдельные участки автоматизации.

2. Какой из стандартов (методик) располагает наиболее развитой системой оценки:

- а) *ITSM*;
- б) *6 Sigma*;
- в) *СММІ*?

3. Более зрелой организацией ИТ в соответствии с моделью *СММІ* по отношению к уровню «количественной управляемости» считается:

- а) уровень «регламентированности»;
- б) уровень «управляемости»;
- в) уровень «оптимизируемости».

4. Какая оценка по модели *СММІ* признается министерством обороны США:

- а) А;
- б) С;
- в) экспертное заключение?

5. Количественные оценки работы процессов ИТ в соответствии с моделью *СММІ* не должны наличествовать на уровне:

- а) начальном;
- б) управляемости;
- в) регламентированности.

6. Более зрелым является подход к управлению информационными технологиями:

- а) проектный;
- б) процессный.

7. *ITIL* — это:

- а) набор стандартов организации ИТ-службы компании или государственного учреждения;
- б) набор рекомендаций по организации работы ИТ-подразделения и его взаимодействия с функциональными подразделениями компании.

8. *ITIL* рекомендует:

- а) договориться ИТ-службе с функциональными подразделениями о наборе предоставляемых в пользование систем автоматизации;
- б) обеспечить процессы предоставления услуг по передаче и обработке информации с обязательной оценкой стоимости предоставления этих услуг (работать «как бизнес»).

9. Процесс в рамках *ITIL* — это:

- а) документ, описывающий полный технологический цикл обработки информации системой автоматизации;
- б) технологический порядок сопровождения АС предприятия;
- в) повторяющаяся последовательность операций, ролей, имеющая вход, цель, результат, стоимость и менеджера.

10. Выберите ошибочное утверждение:

- а) внедрение любой метрики приводит к положительному результату;
- б) система метрик должна последовательно совершенствоваться;

в) система метрик должна быть достаточно стабильна.

11. Метрики должны быть:

- а) общедоступными;
- б) доступными только для руководства;
- в) различные метрики доступны для различной целевой аудитории.

12. Выберите некорректную метрику:

- а) качественная оценка — «эффективно/неэффективно»;
- б) аудит на соответствие международному стандарту;
- в) суммарное время простоя автоматизированных систем.

13. Возможность сравнивать деятельность разнородных подразделений обеспечивают:

- а) любые количественные метрики;
- б) оценка взаимодействующих подразделений по пятибалльной шкале или оценка уровня зрелости *СММІ*.

14. Service Level Agreement предполагает:

- а) договоренность о плановом росте вычислительных ресурсов (план по росту вычислительной мощности, например, на 25 % в год);
- б) заключение договора с функциональными подразделениями на предоставление права использования конкретной марки системы автоматизации;
- в) предоставление набора услуг по автоматизированной обработке информации со стороны подразделения ИТ.

15. Основная цель разработки аварийного плана:

- а) предотвратить аварийные ситуации;
- б) исключить возможность ущерба здоровью и угрозы жизни персонала;
- в) снизить потери в результате наступления аварийной ситуации.

16. В функции аварийного комиссара (дежурного) не входит:

- а) классификация аварийной ситуации;
- б) запуск аварийного плана;
- в) устранение последствий аварийной ситуации.

17. К основным защищаемым свойствам информационных ресурсов относят:

- а) целостность, надежность, доступность;
- б) конфиденциальность, надежность, целостность;
- в) конфиденциальность, доступность, целостность.

18. CRAMM — это:

- а) международный стандарт по защите информации;
- б) методика анализа рисков и выработки контрмер;
- в) уровень зрелости организации защиты информации по международному стандарту.

19. Методика CRAMM обеспечивает:

- а) количественную оценку, сравнение и суммирование величины репутационного и финансового ущерба, возникшего при реализации какой-либо угрозы;
- б) экспертную оценку риска как величины возможного ущерба.

20. Финансовые потери брокерской компании возникают в результате:

- а) сбоя электропитания торговой системы;
- б) простоя торговой системы в случае реализации какой-либо угрозы, включая сбой электропитания.

21. Выберите корректную формулировку описания риска:

- а) риск возникновения финансовых и репутационных потерь в случае отказа от авторства в результате реализации угрозы «маскарад» внешним пользователем;
- б) риск потери данных с момента последнего копирования.

22. При анализе рисков не производится оценка:

- а) вероятности реализации отдельных угроз;
- б) вероятности реализации уязвимостей;
- в) аудит на соответствие стандартам защиты информации.

23. В случае репутационных потерь биржи, возникших в результате недоступности торговой системы в течение двух часов при сбое электропитания, уязвимостью является:

- а) репутационные потери;
- б) недоступность в течение двух часов;
- в) сбой электропитания.

24. В случае финансовых потерь, возникших в результате потери данных при затоплении вычислительного центра брокерской компании, угрозой является:

- а) потеря данных;
- б) финансовые потери;
- в) затопление.

Рекомендуемая литература

О'Коннор, Дж. Искусство системного мышления / Дж. О'Коннор. — М. : Альпина Пабlishер, 2018.

Стеллман, Э. Постигаая *agile* / Э. Стеллман, Дж. Грин. — М. : МИФ, 2018.

Флиvbьорг, Б. Мегапроекты и риски. Анатомия амбиций. — М. : Альпина Пабlishер, 2014.

Халл, Э. Инженерия требований / Э. Халл, К. Джексон, Дж. Дик. — М. : ДМК Пресс, 2016.

CMMI® for Development, Ver. 1.3. — Pittsburgh : Software Engineering Institute Carnegie Mellon University, 2010.

Software Engineering Institute. Appraisal Requirements for CMMI, 2010. — Pittsburgh : Carnegie Mellon Institute, 2010.

URL: <http://www.cmmiinstitute.com>.

Новые издания по смежным дисциплинам

Зуб, А. Т. Управление проектами : учебник и практикум для академического бакалавриата / А. Т. Зуб. — М.: Издательство Юрайт, 2018.

Кузес, Д. М. Наследие лидера / Д. М. Кузес, Б. З. Познер. — М. : Издательство Юрайт, 2016.

Кузнецова, Е. В. Управление портфелем проектов как инструмент реализации корпоративной стратегии : учебник для бакалавриата и магистратуры / Е. В. Кузнецова. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018.

Рейнор, М. Е. Стратегический парадокс / М. Е. Рейнор. — М. : Издательство Юрайт, 2009.

Томас, Р. Д. Испытание лидерства: опыт ведущий к мастерству / Р. Д. Томас. — М. : Издательство Юрайт, 2017.

Управление проектами : учебник и практикум для академического бакалавриата / А. И. Балашов [и др.] ; под общ. ред. Е. М. Роговой. — М. : Издательство Юрайт, 2018.

Федотова, М. А. Проектное финансирование и анализ : учеб. пособие для бакалавриата и магистратуры / М. А. Федотова, И. А. Никонова, Н. А. Лысова. — М. : Издательство Юрайт, 2018.

Ключи к тестам

Раздел I: 1в, 2б, 3б, 4в, 5в, 6а, 7в.

Раздел II: 1в, 2б, 3б, 4б, 5б, 6а, 7г, 8г, 9г, 10в, 11б, 12б, 13а, 14б, 15а.

Раздел III: 1а, 2в, 3в, 4а, 5а, 6б, 7б, 8б, 9в, 10а, 11в, 12а, 13б, 14в, 15в, 16в, 17в, 18б, 19а, 20б, 21а, 22в, 23в, 24в.