

Приветственное слово



**Сергей
Кирюшин**

Главный редактор Учебника 4CIO,
член Совета Клуба 4CIO

Уважаемые коллеги!

В 2011 году вышла первая версия Учебника 4 CIO, в работе над которой приняло участие около 20 высокопрофессиональных CIO и известных экспертов рынка. В работе над версией 2.0, вышедшей на пару лет позже, приняло участие уже около 40 авторов, в результате получилось очень подробное профессиональное руководство для CIO, которое нашло широкое применение на рынке. Сейчас мы приступили к работе над третьей версией Учебника, актуализировали ряд глав, и сейчас мы представляем вашему вниманию специальную сокращённую версию, выпущенную к Конгрессу «Подмосковные вечера — 2017». Работа над Учебником продолжается, в ближайшее время будет подготовлена к выпуску версия 3.0, и мы ждём новых идей и новых авторов — присоединяйтесь к нам!

Приветственное слово



**Александр
Селютин**

Выпускающий редактор
специальной сокращённой версии
Учебника 4CIO

Уважаемые начинающие CIO!

Конечно, задачи ИТ-директора не ограничены теми вопросами, которые рассмотрены в этой редакции, и у нас есть ещё много того, чем опытные члены Клуба 4CIO и его друзья могут с вами поделиться. В эту специальную версию Учебника мы постарались включить ключевые, с нашей точки зрения, вопросы, которые обычно возникают у начинающего ИТ-директора.

Мы искренне надеемся, что Учебник будет вам полезен и поможет в освоении выбранного вами Пути. Ведь CIO – это уже больше, чем просто профессия.



Часть 1

Назначение CIO и его функции

Часть 1. Введение

Глава 1.1

Профессия CIO



Сергей
Кирюшин

Кто такой CIO

CIO (Chief Informational Officer) — главный руководитель компании в области информационных технологий.

Информационные технологии — мощный инструмент бизнеса, — являются средством интеграции, обеспечения эффективного взаимодействия различных бизнес-процессов и бизнес-подразделений, — поэтому для эф-

фективного применения информационных технологий CIO должен хорошо понимать бизнес компании, разбираться во всех её бизнес-процессах, — таким образом, профессия CIO — одна из самых сложных и интересных. И, конечно, профессия CIO чрезвычайно тяжела, так как CIO отвечает за бесперебойную работу всех ИТ-сервисов, и, если где-то проис-

Кто такой CIO?

«Тот, кто обладает соответствующими компетенциями. На мой взгляд, основные компетенции CIO следующие (в порядке важности):

- глубокое понимание бизнеса компании и глубинных причин принятия тех или иных решений в компании;
- умение коммуницировать, прежде всего с вышестоящим руководством, а также с коллегами функциональными руководителями;
- лидерство и умение повести за собой, причём лидерство не только как руководителя своей ИТ-команды, но и как топ-менеджера, способного отстаивать свою позицию перед руководством компании;
- понимание технологий, причём не столько самих технологий, сколько того, как и когда их принести в компанию;
- обучаемость и способность к изменениям».

Павел Пестряков

ходит сбой — виноват CIO и его команда. Кроме того, важнейшей составляющей деятельности CIO является реализация новых проектов по совершенствованию действующих и автоматизации новых бизнес-процессов, и любые возникающие здесь проблемы — также

«бьют» по CIO, хотя зачастую вызваны слабой мотивацией бизнес подразделений и пр. Таким образом, профессия CIO — тяжёлая и ответственная, но при этом она же — одна из самых интересных и многогранных!

Отличие CIO от IT-директора

CIO — один из ключевых топ-менеджеров компании, занимающийся не столько операционным, сколько стратегическим управлением ИТ, при этом ИТ в компании рассматриваются как мощный инструмент бизнеса.

В странах с развитой экономикой CIO является членом коллегиального исполнительного органа компании. Таким образом, он участвует в принятии важнейших решений компании и в целом в её управлении (см. раздел «Корпоративное управление ИТ»).

В России ИТ-директора исторически отвечали в основном только за поддержку ИТ-инфраструктуры и почти никогда не занимались стратегией — в результате очень редко входили в состав руководства компаний. Однако, с ростом роли ИТ в бизнесе, с активным внедрением мощных ИТ-платформ (ERP, информационными хранилищами данных и пр.), — роль ИТ-директора в компаниях существенно выросла, и сегодня многие крупные компании имеют в составе своих руководящих органов CIO.

Существует практика, когда CIO является непосредственным руководителем ИТ-директора, при этом ИТ-директор отвечает прежде всего за эксплуатацию, в то время как CIO — за общее руководство и стратегию. Кроме ИТ-директора, в подчинении CIO могут находиться:

- Подразделение, отвечающее за формирование стратегии развития компании (о связи бизнес-стратегии и ИТ-стратегии (см. раздел «Стратегическое планирование ИТ»);
- Единый проектный офис — в случае, если проектный офис контролирует все проекты компании, а не только в области ИТ;
- Бизнес-аналитики (методологи, технологи и пр.). Как правило, такое подразделение создаётся для упорядочивания бизнес-процессов, разработки стандартов и инструкций, формирования нормативно-методологической базы и пр. Если данное подразделение находится не в подчинении CIO, срок его эффективной работы недолог — не более года с момента создания, так как это подразделение, по сути, «прослойка» между ИТ и бизнес-подразделениями, и через какое-то время оно перестаёт понимать и тех, и других, и превращается в бюрократическую структуру. Если данное подразделение подчиняется CIO — есть возможность его правильного применения при реализации новых проектов и пр.;
- Подразделение информационной безопасности. Это намного более редкий случай, так как несмотря на тесную связь с информационными технологиями, работа со злоумышленниками, мошенниками и пр. — должна вестись службой безопасности компании. Поэтому, как правило, это подразделение, находящееся «на стыке» ИТ и безопасности, — подчиняется руководителю Службы безопасности компании. Но при этом CIO не должен утрачивать методологический и технологический контроль за деятельностью данного подразделения;
- Другие подразделения (в том числе функциональные) по решению руководства компании.

Принципы работы CIO

В данном разделе приводятся несколько базовых принципов работы CIO, без которых сложно достичь успеха в долгосрочной перспективе.

Принцип «Бизнес-ориентированности».

ИТ — это прежде всего инструмент бизнеса компании, поэтому важнейший принцип работы CIO — делать максимально все возможное, чтобы помочь бизнес-подразделениям решить их проблемы и задачи, в соответствии со стратегией развития и требованиями руководства компании.

Принцип «Архитектурной целостности».

Данный принцип гласит, что одной из важнейших задач CIO является «защита» и контроль целостности ИТ-архитектуры. Так, зачастую, идя навстречу различным запросам и «хотелкам» руководителей бизнес-подразделений, которые предлагают внедрить решения от различных производителей ERP-систем, либо ориентирующихся на различные «классы» и типы оборудования и т. п. — CIO тем самым закладывает «бомбу» в дальнейшее развитие компании, увеличивая сложность, стоимость и сроки внесения изменений в создаваемую ИТ-архитектуру.

Принцип «Системности». CIO необходимо решать огромное количество проблем и задач, но нужно не только «затыкать дыры» возникающих проблем, но построить и «запустить» такие механизмы преобразований ИТ в компании, которые позволят системно улучшить ситуацию в перспективе.

Принцип «Измеримости». Системные улучшения реализуются через изменения, которые инициирует и проводит CIO. При планировании изменений, необходимым становится привлечение различных ресурсов (финансовых, кадровых и пр.). Измеримые метрики качества результатов изменений позволяют обосновывать привлечение

необходимых ресурсов. Кроме того, применение «принципа измеримости» помогает доказывать, что изменения действительно приводят к улучшениям.

Принцип «Экспертизы». Принцип говорит о том, что для принятия правильных решений необходимо обладать корректной информацией. Очень важно иметь точную объективную оценку применяемой технологии, параметрам автоматизируемых бизнес-процессов и т. д. Поэтому очень важно прислушиваться к мнению экспертов требуемой области, и необходимо либо самому досконально разбираться в каждой предметной области (как правило, для CIO это непозволительная роскошь), либо иметь собственных экспертов, работающих в организации, либо при отсутствии собственной экспертизы (а это довольно часто, особенно при новых проектах), иметь возможность «взять» её у партнёров, на рынке, в профессиональных сообществах и т. п.

Принцип «Инновационности». Принцип говорит о том, что CIO, обладая знаниями о современных тенденциях в ИТ, появлении новых технологических решений и прочего может выступить в роли «локомотива» внедрения различных инноваций и современных методов обработки информации в компании. И именно активной позиции в вопросах развития очень часто ожидают от CIO генеральные директоры компаний.

Принцип «Публичности». Публичность необходима для визуализации результатов работы проектов, процессов и персонала, на основе чего можно построить механизм их объективной оценки, выявления и последующего поощрения/продвижения наилучших сотрудников, а также наказания наихудших. То есть должна заработать СИСТЕМА, работающая на рост эффективности процессов и персонала.

Принцип «Одного Ответственного». Данный принцип гласит, что каждая задача (процесс, либо проект, программа) должна иметь своего «владельца», который несёт персональную ответственность за её решение/выполнение:

- 1) не с кого будет спросить за результат;
- 2) мы не сможем собрать статистику результатов и понять: кто является хорошим менеджером (работником), а кто наоборот, и соответственно не сможем поощрять лучших и наказывать худших.

Кроме того, необходимо включить в данный принцип подход, при котором не происходит частой смены руководителей проектов, направлений и пр., без чётких и публичных обоснований. Каждый менеджер, завершающий

проект (или выходящий из проекта) — должен получить соответствующую оценку его деятельности в данном проекте, с применением «Принципа Публичности». И эта оценка результатов деятельности — должна приводить либо к поощрению, либо к наказанию, с информированием об этом (с соблюдением Принципа «Публичности»).

Принцип «Справедливости». Когда в результате применения всех вышеуказанных принципов создаётся система, при которой наиболее эффективные и успешные менеджеры (сотрудники) поощряются и продвигаются (в том числе только успешным поручаются новые сложные и ответственные задачи/проекты), а наименее успешные — отстраняются/увольняются.

Полномочия CIO

Чем шире спектр полномочий CIO — тем больше вопросов он может решать самостоятельно, — тем быстрее и эффективнее сможет он решать поставленные задачи. Определим «максимальный» перечень полномочий, необходимый для быстрого и эффективного решения CIO своих задач:

1. Возможность упрощённого доступа (на приём и пр.) к генеральному директору (либо его «ключевому» заместителю) с изложением своих проблем и предложений. Так как без существенной поддержки первых лиц невозможно эффективно заниматься развитием компании.

2. Возможность принятия кадровых решений по подчинённым сотрудникам (в «идеале» — в том числе в части возможности увеличения либо изменения подчинённой организационной структуры). С учётом того, что к назначению и отставке сотрудников нужно относиться с крайней осторожностью, — возможность набирать в штат, назначать, увольнять и перемещать сотрудников

внутри подразделения с минимальным объёмом согласований, — сильно поможет CIO в решении его задач, в формировании требуемой ему команды менеджеров и специалистов (подробнее об управлении персоналом см. главу 3.3. «Управление персоналом»).

3. Быть бюджетодержателем по «своим» статьям бюджета, — то есть иметь полномочия самостоятельного формирования и защиты ИТ-бюджета. Очень важно обеспечить ИТ необходимым финансированием — как деятельность по поддержке ИТ-инфраструктуры и информационных систем, — так и по реализации новых проектов, и чем выше полномочия и «независимость» CIO в данном вопросе — тем лучше будет финансовое обеспечение ИТ-деятельности. При этом не надо забывать, что объем финансирования ИТ должен быть не маленьким и не большим, — а адекватным задачам компании. То есть если Вы «выбили» большой ИТ-бюджет и не смогли его эффективно (то есть с максимальной отдачей для компании) потратить — это также проблема для CIO.

Полезна практика, когда бизнес-подразделения сами «защищают» бюджет на свои новые проекты (при условии, что бюджетодержатель — ИТ) — тогда больше шансов на выделение необходимых сумм на развитие, повышается заинтересованность бизнес-подразделений в достижении результатов проектов.

4. Доверенность на подписание ИТ-договоров, что позволяет более оперативно принимать решения — организовывать работу с подрядчиками, инициировать новые проекты и пр. Если CIO не имеет полномочий подписывать ИТ-договоры, то он не просто вынужден тратить больше времени и усилий на представление договоров вышестоящему руководителю, — он также и несёт меньшую ответственность за успешную реализацию данного договора.

5. Полномочия по инициации и управлению разработкой ИТ-стратегии, а также проведения различных аудитов в области ИТ и регламентации бизнес-процессов.

6. Полномочия по утверждению регламентирующих документов, касающихся ИТ. Наличие данных полномочий позволит оперативно менять различные параметры ИТ-сервисов, а также решать «внутрицеховые» задачи ИТ.

7. Назначение CIO руководителем Комитета (Совета) по ИТ компании. Это позволяет самостоятельно управлять деятельностью Комитета (Совета) по ИТ и проводить необходимые CIO решения. Однако зачастую «политически» для CIO правильно быть заместителем руководителя Комитета (Совета), в то время как руководителем будет являться генераль-

ный директор либо его первый заместитель. Особенно это актуально, если у CIO нет значительного опыта и/или авторитета в компании.

8. Наличие «Права на ошибку».

Полномочия CIO по сферам деятельности

Функционально и организационно ИТ появились уже после «связи и телекоммуникаций» и «производственных» систем (либо автоматизированных систем технологического управления, АСТУ). На практике сегодня эти «зоны» ответственности очень сильно пересекаются.

«Служба связи» исторически отвечала за телефонию и в ряде случаев — за технологическую производственную связь. После появления компьютеров и локальных сетей Служба связи зачастую стала отвечать и за компьютерную сеть — локальную вычислительную сеть (ЛВС) компании, и в перспективе также и за внешнюю кабельную сеть компании (ВКС). Однако, по мере развития информатизации, Интернета и пр., — все большие объёмы информации стали передаваться по сетям, появились удалённые ЦОД, появились IP-телефония и видеоконференцсвязь, — и получилось, что ИТ стала главным «потребителем» каналов связи. А с учётом того факта, что в целях оптимизации управления и затрат целесообразно все информационные потоки: файлы, «транзакции», «технологическую» информацию, а также телефонию — передавать по единым каналам связи, с использованием единой инфраструктуры и внешних провайдеров, — стала преобладать тенденция включения Служб связи в единое ИТ-подразделение компании. Это позволяет оптимизировать расходы и комплексно решать все телекоммуникационные вопросы.

АСТУ — информационные технологии, зародившиеся в недрах и под управлением производственных подразделений — как правило, под руководством тех-

Есть известный анекдот про «три конверта», когда уходящий руководитель вручает приходящему на его место три запечатанных конверта, которые рекомендует вскрывать последовательно в случае появления серьёзных проблем. В первом конверте было написано «Обвиняй во всем предыдущего руководителя», во втором — «Обещай», ну а в третьем — «Пиши три конверта»...

нического директора или главного инженера. Основная задача АСТУ — автоматизация производственных процессов «нижнего уровня», в основном управляющие работой производственного оборудования на промышленных предприятиях.

Как правило, системы АСТУ получают информацию от различных датчиков, установленных на оборудовании, либо с интеллектуальных систем, встроенных в оборудование. Однако, следует сказать, что АСТУ прежде всего — автоматизированные системы, использующие, как правило, те же персональные компьютеры и серверы, а нередко и каналы связи, — что и остальные ИТ-системы. У АСТУ — большое количество «пересечений» с ИТ-системами и в ряде случаев «грань» между АСТУ и ИТ — «размыта» (например, в области техобслуживания и ремонтов). В ре-

зультате, имея отдельное, не подчиняющееся СІО ИТ-подразделение (АСТУ) — сложно проводить единую ИТ-политику в компании.

Таким образом, с развитием ИТ, усложнением систем и ростом объёма (и значения!) передаваемой ИТ- информации через телекоммуникационные каналы, с растущим уровнем интеграции различных информационных систем в компании, необходимостью проведения комплексной взвешенной ИТ-политики — наиболее целесообразным является объединение трёх вышеуказанных направлений в одно, с подчинением их непосредственно СІО, — и это тот явный тренд, который имеется у компаний сегодня. В том числе, консолидация указанных подразделений в рамках единого ИТ-подразделения говорит об определённом уровне зрелости компании, её способности развиваться в современных условиях.

Ответственность СІО

Как нас учили «классики» — нет полномочий без ответственности, поэтому каждый СІО должен осознанно брать на себя только те полномочия, за которые он может и хочет нести ответственность, и делать это эффективно. Определим перечень вопросов, ответственность за решение которых должен брать на себя эффективный СІО:

1. Обеспечивать надёжную эксплуатацию информационных систем и сервисов компании. Как отдельную задачу, следует указать обеспечение высокого качества информационного обслуживания топ-менеджмента компании.

2. Быть надёжным партнёром топ-менеджмента компании и вопросах её развития. Вести эффективную проектную деятельность по внедрению новых информационных систем и бизнес-процессов.

3. Чётко обосновывать расходы на ИТ. Постоянно работать над сокращением расходов на ИТ там, где это возможно и рационально.

4. Формировать надёжную и эффективную ИТ-инфраструктуру.

5. Выстраивать эффективную ИТ-архитектуру информационных систем, не допускать «перекосов» в её формировании и развитии.

6. Быть «локомотивом» развития компании, постоянно предлагать инновации, информировать руководство и функциональных заказчиков о новейших трендах в ИТ и пр.

7. Осуществлять контроль за информацией и информационными потоками, решение вопросов информационной безопасности.

8. Иметь «прозрачную» систему отчётности для руководства компании — по расходам, выполнению проектов и поручений и пр.

9. Чётко выполнять поручения руководства компании.

В «идеале» — максимально зафиксировать документально те требования, которые

предъявляет Вам руководство (в том числе с использованием измеряемых показателей

эффективности, KPI). Однако на практике это трудноосуществимо.

С чего начать новому CIO

Вновь назначенный CIO как правило имеет «карт-бланш» на вхождение в компанию, с наделением «иммунитета» на срок, как правило — от полугода до года (далее будем указывать среднюю величину — 9 месяцев). Вместе с тем, очень важно то, как Вы себя проявите сразу, с самого начала.

В течение **первых двух недель** очень важно дать понять, что с Вашим приходом в компании произошли изменения. Вы должны быть на положительном «контрасте» с предыдущим CIO, в том числе во внешних проявлениях, — вплоть до формы одежды, перестановки мебели, изменения расписания совещаний и т. д., и т. п.

В течение **первых трёх месяцев** необходимо максимально ознакомиться с коллективом, бизнес-процессами и деловой атмосферой в компании, установить правильные отношения с коллективом и топ-менеджерами, дать «быстрый эффект» там, где это возможно, и инициировать ряд стратегических проектов и инициатив.

Первоначальный период вхождения в компанию надо провести максимально эффективно, чтобы правильно «влиться» в коллектив, стать полноправным членом управленческой команды. В противном случае, вероятнее всего, **через 9 месяцев** (или даже раньше) начнутся поиски нового CIO...

Так что нужно для этого сделать?

1. Выстроить хорошие рабочие отношения с Вашим непосредственным руководителем. Принять стиль его руководства и постараться оказаться максимально ему полезным. Необходимо регулярно заходить к руководителю за указаниями и с вопросами о его видении тех или иных аспектов деятельности ИТ и компании в целом (под-

робнее об отношениях с руководством см. в главе 2.5 «Управление отношениями»).

2. Незамедлительно приступить к решению тех задач, которые поставило перед Вами руководство компании при назначении на должность, и информировать руководство о ходе их решения.

3. Лично познакомиться и попытаться установить хорошие отношения со всеми ключевыми бизнес-руководителями. Ключевые руководители бизнес-подразделений — Ваши заказчики, и от Ваших отношений с ними будет зависеть (например, доверят ли они Вам решение своих проблем, либо будут искать «обходные» пути). Очень важно лично встретиться с ними, понять их основные проблемы — как в бизнесе, так и с ИТ, — с тем чтобы в дальнейшем предложить им их решение.

4. Создать или реорганизовать, а при необходимости и возглавить Комитет (Совет) по ИТ. Это даст Вам больший «политический» вес в глазах всей компании, позволит проводить через Комитет (Совет) необходимые вопросы, в том числе обсуждение результатов ИТ-аудита и согласование ИТ-стратегии. Однако, зачастую «политически» для CIO является правильным — быть заместителем руководителя Комитета (Совета), в то время как руководителем будет являться генеральный директор либо его первый заместитель. Особенно это актуально, если у CIO нет значительного опыта и/или авторитета в компании.

5. Сформировать в ИТ профессионально сильную и лояльную Вам команду. Для этого надо прежде всего познакомиться с действующей командой и определить: кто сильный менеджер (специалист), а кто нет; кто лоялен Вам, а кто нет и т. д.

Следует учесть, что раз Вы — CIO, Вы будете работать прежде всего с подчинёнными ИТ-менеджерами, и поэтому Вам прежде всего нужно определиться по ним, и затем уже, при помощи менеджеров, со специалистами. При этом лояльность к Вам Ваших заместителей имеет приоритет над их уровнем профессионализма. Однако в целом следует учитывать, что если перед Вами приоритетной поставлена задача развития, то Вам важнее найти сильных профессионалов — специалистов и менеджеров, а если задача — приоритетно сохранить и удерживать существующее положение — предпочтение следует отдавать лояльным Вам подчинённым... Но во всем нужен разумный компромисс.

CIO не становятся в одночасье. Это определённая вершина карьерного роста, предполагающая некий, для каждого свой путь. К определённому моменту у CIO появляется «своя» команда, «скамейка запасных», с которой новые ИТ-руководители зачастую приходят в новую компанию. Однако важно не «перегнуть палку», с тем, чтобы у базового коллектива это не вызвало сильного отторжения. И важно, чтобы новые, взятые Вами менеджеры сохранили ключевых специалистов в компании и не заменили бы их «своими» сотрудниками по принципу лояльности, но в ущерб интересам дела.

При реорганизации ИТ-подразделения следует избегать понижения сотрудников в окладах и позициях, если, конечно, Вы не хотите от них избавиться — в таких случаях они будут нелояльны и с большой вероятностью будут искать новое место. Очень негативно на человека также влияет лишение удобного рабочего места, кабинета (если это не общий переезд и пр. в компании). Таким образом, лучше уволить человека, нежели понизить ему зарплату и ухудшить условия труда (при условии их сохранения у других сотрудников), — иначе Вы получите обиженного, нелояльного Вам и незаинтересованного в результатах труда сотрудника.

Следует учесть, что сохранение и даже продвижение наиболее достойных представителей старой команды — будет позитивно встречено коллективом. Однако новые менеджеры в Вашей управленческой команде все же необходимы — как минимум для повышения уровня лояльности коллектива к Вам (так как Вы назначали их, а не прежний руководитель). Поэтому замену людей нужно начинать с Ваших непосредственных заместителей, — одновременно Вы снижаете риск последующей замены Вас Вашим заместителем... А Ваши заместители уже должны сформировать (под Вашим контролем!) эффективную команду в своих подразделениях.

6. Провести ИТ-аудит и разработать новую ИТ-стратегию. Аудит необходим Вам для более быстрого «вхождения» в дела, для понимания всех имеющихся проблем, «узких мест» и пр. Кроме того, Вы сразу выявляете и фиксируете ошибки и проблемы, имевшие место ещё до Вашего прихода, что позволит избежать обвинений в дальнейшем. Выработка ИТ-стратегии должна являться результатом ИТ-аудита. В крайнем случае, результатом должен стать перечень задач и инициатив, которые Вы вынесете на рассмотрение и утверждение руководством компании. Утверждённая ИТ-стратегия позволит Вам в дальнейшем приступить к реализации намеченных планов и проектов, получить для этого необходимое финансирование, полномочия и пр.

Для проведения ИТ-аудита и разработки ИТ-стратегии Вам необходима сильная **команда специалистов (экспертов)**. Силами собственного ИТ-подразделения такую работу провести сложно — как правило, не хватает собственной экспертизы, к тому же старый персонал может стремиться «сгладить» имеющиеся проблемы и оправдать принятые ранее решения. Кроме того, всегда полезно иметь «свежий» взгляд на вещи. Поэтому наилучший выход для проведения аудита и раз-

работки стратегии — привлечь внешнюю экспертную компанию на договорной основе.

Для проведения аудита и разработки ИТ-стратегии всегда очень важно получить авторитетное заключение, не «боящееся» критики со стороны Ваших противников. Лучше всего для этого к аудиту привлекать компании с мировым именем в области ИТ — такие как IBM, Accenture и пр. Наличие указанного «бренда» на Вашем заключении или предлагаемой ИТ-стратегии — не только свидетельство качества документов, но и возможность избегания критики в некомпетентности. При этом желательно выбрать компанию, которая ещё не имела договорных отношений с Вашей компанией (по крайней мере в течение значительного времени) — тогда на компанию-аудитора в меньшей степени возможно давление со стороны кого-нибудь «внутри» Вашей компании, к тому же, заключая такой контракт, Вы даёте новый бизнес компании-аудитору и таким образом можете рассчитывать на их лояльность.

Как проводить ИТ-аудит? Кроме проведения тесной работы с сотрудниками ИТ-подразделений компании, изучения имеющейся ИТ-инфраструктуры, приложений, договорной базы и пр., очень важно организовать встречи-обсуждения с ключевыми руководителями и специалистами бизнес-подразделений, с тем, чтобы их пожелания были максимально учтены, и, кроме того, чтобы они поддержали Ваши предложения в рамках ИТ-стратегии, так как она формировалась при их непосредственном участии (подробнее об аудите см. главу 3.1 «Управление ИТ-инфраструктурой»).

Тесная работа с бизнес-подразделениями в рамках ИТ-аудита и разработки ИТ-стратегии — также позволяет принять ИТ-стратегию в условиях отсутствия общей бизнес-стратегии компании (хотя по-хорошему она должна идти за общей бизнес-стратегией, либо быть составной её частью).

7. После утверждения ИТ-стратегии незамедлительно приступить к её реализации — инициировать новые проекты, реорганизацию инфраструктуры и пр. Это позволит Вам продемонстрировать свою активность и целеустремлённость, взять на себя реальную ответственность и проявить себя.

8. Важно инициировать один или несколько масштабных проектов для бизнеса компании, в рамках утверждённой ИТ-стратегии. Следует отметить, что реализация с Вашим участием масштабного проекта всегда будет свидетельствовать о доверии к Вам со стороны руководства, о том, что от Вас зависит продвижение бизнеса компании и т. д. К тому же, инициация Вами новых крупных проектов даёт определённый «иммунитет» на работу в компании, Вы гарантированно пройдёте «испытательный 9-месячный срок», так как в случае Вашего увольнения или смещения с должности ответственность за результаты проекта необходимо будет переложить на кого-то другого, а это всегда риски... И вообще — «по движущейся мишени тяжело попасть».

9. Кроме необходимости запуска масштабных проектов, для повышения авторитета CIO в глазах руководства компании, нужны «быстрые результаты». То есть нужно быстро найти те области/задачи, которые можно было бы решить легко и в результате получить хороший презентативный эффект. Скорее всего, данные задачи будут лежать в технической области и не связаны со сложным изменением бизнес-процессов и процедур. Например — организация видеоконференцсвязи, создание нового дизайна для Интернет-сайта и т. п. Часто таким «эффектом» может стать сокращение расходов на телекоммуникации — их проще всего «вычислить», а учитывая тенденцию, что цены на телеком постоянно снижаются — вероятность того, что определённое время никто не занимался их снижением, довольно высока.

Сложно переоценить **важность общения CIO с «коллегами по цеху»** — другими CIO, которые могут подсказать интересные решения, поделиться опытом, предложить помощь и т. п. Также важно знать о происходящих событиях на рынке, о новых процессах и экспертизах, появляющихся у вендоров и интеграторов и пр. Для решения этих вопросов предлагается вступить в профессиональное сообщество CIO, например, Клуб ИТ-директоров 4CIO 4cio.ru.

Вышеуказанные мероприятия имеет смысл проводить не только при первичном «вхождении» CIO в компанию, но и периодически акцентировать на них своё внимание. Более того, всегда правильно инициировать ИТ-аудит самому, а не дожидаться, когда этого потребуют другие. Так, обновлять ИТ-стратегию желательно каждые 3-5 лет, либо можно вносить в неё небольшие корректировки ежегодно.

Что же касается персонала, то в данном случае необходимо вести постоянную работу по поиску новых, замене, продвижению и пр. Ваших кадров, при этом возможно проведение регулярных (желательно не чаще одного раза в год — полтора) реорганизаций ИТ-подразделения, в целях отказа от ненужных Вам сотрудников, открытия вакансий для новых (прежде всего под новые задачи и проекты, утверждённые руководством компании) и т. д. Не спешите решить все кадровые проблемы «одним махом», — иногда может потребоваться несколько «транзакций» по реорганизации Вашего коллектива с интервалом в год-два, чтобы избавиться от ненужных подразделений и сотрудников и прийти к нужной Вам организационно-кадровой структуре.

Перспективы CIO

В современном мире идёт постоянный и все возрастающий рост объёмов обрабатываемой информации, растёт сложность информационных систем и их значение для

класса BPM (Business Process Management, системы управления процессами), развитием аутсорсинга и наметившейся тенденцией «ухода» в «облака», — ИТ-компетенция

Кто такой CIO?

С учётом вышеизложенного, и принимая во внимание важную и все растущую роль информационных технологий и руководителя в компании, отвечающего за информационные технологии, — термин CIO (Chief Informational Officer) все чаще трактуется как «главный руководитель компании по управлению информацией».

будет все сильнее концентрироваться в специализированных внешних ИТ-компаниях и поэтому функция ИТ в самой компании будет менее значима; ИТ будет более сопоставимы с utilities (потребляемыми ресурсами — электричество, вода и пр.). При этом роль CIO

бизнеса, — поэтому роль CIO в компании неизменно будет возрастать как минимум — в 5-10 летней перспективе.

А что будет дальше — пока вопрос открытый, так как существует мнение, что с развитием комплексных ИТ-решений типа ERP, систем

будет трансформироваться, — он будет все меньше заниматься инфраструктурными вопросами и управлением собственного персонала, но все больше — организацией информационных потоков и контролем внешних сервисных компаний.

Часть 1. Введение

Глава 1.2

Архитектура предприятия и ИТ-архитектура



Марина
Аншина



Константин
Зимин



Владимир
Ананьин



Сергей
Кирюшин

Цифровая трансформация прочно укрепилась в качестве направления развития нашей страны. 5 июля 2017 г. министр связи и массовых коммуникаций Российской Федерации Николай Никифоров в рамках заседания Совета при Президенте по стратегическому развитию и приоритетным проектам представил Владимиру Путину разработанную министерством программу «Цифровая экономика». Главная идея программы по словам министра состоит в том, чтобы создать в России определённый набор условий для запуска и ускорения цифровизации привычного жизненного и экономического уклада.

Таким образом на государственном уровне провозглашено, что ИТ приобретают важнейшую роль в развитии общества, государства и бизнеса. Переход от рассмотрения ИТ исключительно как поддерживающего деятельность организаций сервиса к такому взгляду на них, как на двигатель развития, требует се-

рьёзного пересмотра многих подходов к ИТ и связанным с ними областям деятельности. В частности, становится жизненно необходим системный подход, который сосредоточен в понятии архитектура предприятия и связанных с этим понятием методиках и практиках. Современным организациям, которые двигаются по пути цифровой трансформации, необходима достоверная, надёжная, безопасная и оперативная информация. Такая информация может быть предоставлена только в результате грамотного использования современных ИТ с учётом как потребностей организации, так и их возможностей, а также внешних условий и двигателей их развития. Только системный подход к получению, обработке и предоставлению информации позволит справиться с этой задачей. «Кусочная» автоматизация отдельных частей предприятий без учёта их связей и рассмотрения предприятия как единого целого принесёт ему вред, а не пользу, и

обычно существенно замедляет его развитие, а то и приводит к более угрожающим последствиям. Именно поэтому очень важно рассматривать предприятие целиком в концепции «архитектуры предприятия».

Однако распространение архитектурного подхода в нашей стране идёт очень медленно и далеко ещё не достигло того уровня, который необходим для цифровой трансформации предприятий и цифровой экономики. Такая ситуация опасна так же, как строительство современного здания, без расчёта прочности и создания соответствующего фундамента. Оно может рухнуть в любой момент и привести к большим потерям.

Начать эту главу мы хотим с краткого экскурса в возникновение и развитие понятия «архитектура предприятия».

Как область знаний архитектура предприятия родилась в 80-х годах прошлого века в ответ на вопросы, которые волновали как отдельные предприятия, так и целые государства: почему такой полезный инструмент, как ИТ, несмо-

тря на значительные усилия заинтересованных сторон и существенные инвестиции, не приносит ожидаемых выгод? Одна из причин неэффективности — оторванность автоматизации от бизнес-процессов предприятия и несогласованность усилий по автоматизации отдельных подразделений. Сейчас с появлением облачных технологий эта угроза вышла на новый уровень: возникают так называемые теневые облака, когда отдельные функциональные подразделения, привлечённые видимой простотой внедрения и относительной дешевизной облаков (ежемесячный платёж за аренду ПО может быть «невидим» в бюджете подразделения), внедряют их вне архитектуры предприятия, без согласования с подразделениями ИТ, которые об этих проектах зачастую даже не подозревают.

В начале 80-х годов компания IBM начала разрабатывать метод Business System Planning (BSP), предназначенный для анализа и описания архитектуры программных систем. BSP определяет 13 этапов внедрения информа-

Джон Захман окончил Химический факультет Университета Northwestern, затем несколько лет прослужил офицером в Армии США. В 1964 пришёл в IBM, где занимал ряд маркетинговых позиций в Чикаго, Нью-Йорке и Лос-Анжелесе. В 1970 г. стал заниматься методологией BSP. На основании анализа BSP Джон Захман, создал свою модель, ставшую стандартом де-факто моделирования архитектуры предприятия. Он описал эту модель в ряде статей, первая из которых появилась в 1987 г. В ней он одним из первых употребил термин «архитектура предприятия» (Enterprise Architecture).

Джон Захман в течение 26 лет проработал в IBM и был так же, как и многие, не удовлетворён результатами проектов внедрения корпоративных информационных систем, в которых принимал участие. BSP также не принёс ожидаемых улучшений. Захман задумался над тем, как объединить опыт консультантов и знание предприятия, которым обладают его руководство и сотрудники, и для этой цели разработал свою модель архитектуры предприятия. «Схема архитектуры позволяет концентрироваться на отдельных аспектах системы и, в то же время, не терять ощущение общего контекста или «холистической» перспективы (то есть взгляда на предприятие как на целое). Именно потеря такой перспективы, в частности, разработка систем субподрядчиками, находящимися «вне контекста», уже около пятидесяти лет составляет причину появления неинтегрируемых и не поддерживающих предприятие систем, которые к тому же весьма дорого заменять».

Джон Захман
«A Framework for Information» System Architecture.
IBM System Journal, vol. 26, no. 3, 1987

ционных систем и основан на построении и постепенном уточнении ряда матриц, в частности, матрицы руководители — процесс, определяющей вовлеченность руководителей в основные бизнес-процессы, матрицы

информационные системы — руководители, отражающей потребность руководителей в программных системах, матрицы информационных системы — процессы, матрицы информационных системы — файлы данных.

Роль CIO — архитектор предприятия

Для моделирования архитектуры предприятия появилась новая роль (должность) — архитектор предприятия. Он отвечает за создание, развитие, актуализацию корпоративных архитектурных моделей. Он также следит за тем, чтобы архитектура внедряемых программных систем с одной стороны соответствовала архитектуре предприятия, а с другой — способствовала её развитию. На больших предприятиях в связи со сложностью архитектурных моделей существуют отдельные подразделения, объединяющие таких архитекторов. Архитектор предприятия

должен обладать системным подходом, аналитическим мышлением, широкой эрудицией в области ИТ, корпоративного управления, бизнес-процессов, владеть средствами моделирования архитектуры. Архитектор предприятия должен уметь понимать проблемы и предметную область бизнеса и объяснять их техническим специалистам, а также владеть принципами современных технологий и объяснять их возможности представителям бизнеса. Области компетенции архитектора предприятия показаны на Рис. 1.2.1.

Рис. 1.2.1. Роль CIO – архитектор предприятия.



Акт Клингера-Коэна (Clinger-Coen Act — CCA) — это федеральный закон США 1996 г., который предписывает государственным предприятиям соблюдать принципы управления ИТ, основанные на измерении производительности и архитектуре предприятия. CCA требует, чтобы каждое государственное агентство США разрабатывало модель архитектуры предприятия. CCA послужил источником значительных изменений в ролях и ответственности различных федеральных агентств в области управления развитием и приобретением ИТ. В частности, с ним связано появление позиции CIO во всех государственных предприятиях. Появление закона инициировало внедрение концепции архитектуры предприятия не только в государственных агентствах, но и во всех организациях США. Для улучшения использования информационных ресурсов и выполнения акта Клингера-Коэна было образовано межагентское объединение американского правительства CIO Council. В 1998 г. CIO Council начал развивать Federal Enterprise Architecture Framework (Структура Федеральной Архитектуры Госпредприятий) — один из Фреймвоков архитектуры предприятия.

- инвестиции в ИТ должны поддерживать стратегию развития ИТ, согласованную со стратегией развития организации и основываться на архитектуре предприятия;

- в каждой государственной организации должен быть руководитель — CIO, показатели деятельности которого должны быть чётко определены;

- одна из основных функций CIO — развитие архитектуры предприятия.

Таким образом, акт Клингера-Коэна провозгласил, что моделирование архитектуры предприятия является основой процесса планирования затрат и управле-

ния инвестициями правительства США в области ИТ, и это область ответственности CIO. Хотя в России нет пока аналога акта Клингера-Коэна, нельзя забывать, что одно из самых зрелых с точки зрения ИТ государств считает этот процесс основной обязанностью CIO.

Архитектурный процесс на предприятии должен осуществляться под руководством CIO. В акте Клингера-Коэна, который регулирует политику в области ИТ для государственных организаций США, к инвестициям в ИТ предъявляются следующие требования:

Общие понятия и определения

Архитектура предприятия (Enterprise Architecture) — это область знаний об организации, её компонентах, их взаимодействии, с учётом развития. Термин «архитектура предприятия» является сужением термина «архитектура системы», который в соответствии с IEEE определяется следующим образом:

Архитектура — это базовая организация системы, воплощённая в её компонентах, их отношениях между собой и с окружением, а также принципы, определяющие её проектирование и развитие. (ANSI/IEEE Std 1471-2000) Предприятие является, несомненно, одной из самых сложных систем окружающего нас мира и для него полностью правомерно ис-

пользование понятия «архитектура». И чем больше и сложнее предприятие, тем важнее для него архитектурный подход.

ИТ-архитектура — это составная часть архитектуры предприятия. К сожалению, до этого термин «архитектура» относился, прежде всего, к программным системам и в основном решал вопрос, на каком оборудовании и каким образом размещать компоненты приложений. Возможно, именно поэтому до сих пор существует иллюзия того, что архитектура предприятия относится только к ИТ, хотя подобный подход ведёт к неэффективной работе как ИТ, так и всего предприятия. Определение из акта Клингера-Коэна чётко

перечисляет, какие области охватывает архитектура предприятия:

«Архитектура предприятия» — это управленческая инженерная дисциплина, представляющая исчерпывающий обзор предприятия, включая стратегическое планирование, организационное планирование, управление взаимодействиями, улучшение бизнес-процессов, управление информацией и знаниями, и операциями».

Важно понимать, что вне зависимости от нашего желания архитектура объективно существует у каждого отдельного предприятия, а также холдинга, объединяющего разные предприятия. Причём, как правило, во втором случае архитектура холдинга не является

простым объединением архитектур входящих в него предприятий, т. к. существуют сквозные процессы, общие данные и корпоративные системы. Система «предприятие» в полной мере обладает таким свойством систем как синергия (эмерджентность, холизм, системный эффект) — появление у системы свойств, не присущих элементам системы; принципиальная несводимость свойств системы к сумме свойств составляющих её компонентов (неадитивность). Возможности системы превосходят сумму возможностей составляющих её частей; общая производительность или функциональность системы лучше, чем у простой суммы элементов.

Архитектурные модели и фреймвоки

Архитектура предприятия может быть представлена различными моделями, которые аналогичны проекциям и видам в архитектуре зданий. Каждая модель отражает архитектуру предприятия с определённой долей глубины и соответствия.

Выбор модели определяется целью применения архитектурного подхода. Ниже мы подробно расскажем о нескольких самых распространённых моделях и фреймвоках архитектуры предприятия — модели Захмана, Стивена Спивака и TOGAF, а также стандарте на описание архитектуры предприятия

Archimate. Кроме них существуют и другие важные архитектурные модели: FEA (Federal Enterprise Architecture, наиболее интерес-

ная её часть — справочная модель производительности (Performance Reference Model, PRM), DoDAF (Department of Defence Architecture

Существует множество подходов к построению архитектурных моделей предприятия. Обычно их называют фреймвоками (framework). На русский язык это слово часто переводят как логическая структура, рамочная модель или каркас. Однако точного перевода этого понятия не существует. **Фреймвок** — это набор методов, средств, моделей, рекомендаций, направленных, в данном случае, на развитие архитектуры предприятия.

В области архитектуры предприятия наиболее известны следующие фреймвоки:

FEAF — Federal Enterprise Architecture Framework

DoDAF — Department of Defence Architecture Framework

TOGAF — The Open Group Architecture Framework

Framework), GERAM (Generalised Enterprise Reference Architecture and Methodology) и MOF (Microsoft Operations Framework).

Четыре традиционных уровня описания архитектуры предприятия

Большинство моделей описания архитектуры предприятия в той или иной степени используют «перспективы», уровни взгляда на предприятие или слои. Как правило, в качестве основных уровней (слоёв) архитектуры предприятия рассматриваются четыре архитектуры:

1. Архитектура бизнеса (бизнес-архитектура).
2. Архитектура данных.
3. Архитектура приложений.
4. Технологическая архитектура (инфраструктура).

Именно такие уровни (слои) описания архитектуры использованы в моделях TOGAF, FEA, Стивена Спивака, MOF и других. Опишем подробнее, какие элементы включаются в описание каждой из архитектур.

1. Архитектура бизнеса описывает, как работает бизнес. Существует множество подходов к описанию того, как работает бизнес, и нет единого устоявшегося мнения о том, какие элементы следует включить в описание архитектуры бизнеса. Однако, принято считать, что наиболее значимыми элементами в этой об-

ласти являются «процессы и информация», «организация» и «производительность».

Элемент «процессы и информация» наиболее важен, так как описывает и классифицирует бизнес-структуры, бизнес-процессы и потоки деятельности, которые составляют бизнес-модель организации. Элемент «организация» описывает организационную структуру и методы работы, продукты и услуги, которые производит бизнес, бизнес-единицы их размещение и т. д. Элемент «производительность» описывает показатели, которые измеряют эффективность работы предприятия (производительность, бизнес-риски и др.). Один из возможных вариантов отображения бизнес-архитектуры приведён на Рис. 1.2.2.

2. Архитектура данных описывает данные, используемые предприятием. Принято выделять три составляющих элемента архитектуры данных:

- Структура данных.
- Требования к переносу данных.
- Требования по управлению данными.

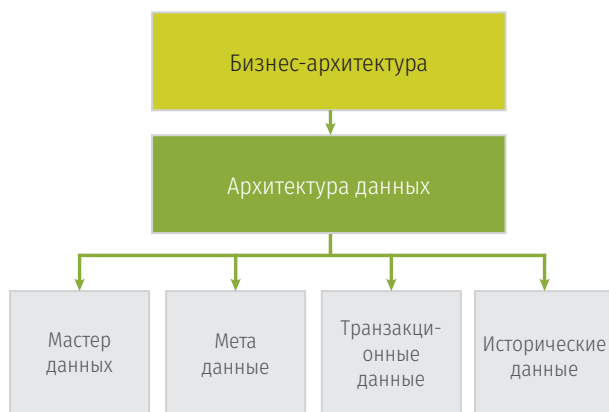
2.2. Структура данных, в свою очередь, со-

Рис. 1.2.2. Вариант отображения бизнес-архитектуры.



стоит из классификаторов и характеристик данных. Выделяются четыре типа данных: мастер-данные, метаданные, транзакционные и исторические данные (Рис. 1.2.3).

Рис. 1.2.3. Типы данных в описании архитектуры данных.



Мастер-данные — это ключевая информация для реализации бизнес-функций компании. Как правило, это данные о клиентах, продуктах, сотрудниках, поставщиках, материальных объектах компании и т. д. Мастер-данные обычно используются различными функциональными подразделениями компании.

Метаданные — это данные о данных, то есть форматы представления данных. Формат метаданных представляет собой стандарт, предназначенный для формального описания некоторой категории ресурсов (объектов, сущностей и т. п.). Такой стандарт обычно включает в себя набор полей (атрибутов, свойств, элементов метаданных), позволяющих характеризовать рассматриваемый объект.

Транзакционные данные — это данные, непосредственно описывающие событие, операцию или транзакцию. Транзакционные данные, как это следует из названия, характеризуют параметры транзакций и в общем случае содержат информацию о времени, статусе, характере выполнения операции. Примером транзакционных данных могут

служить данные о платежах, заказах и т. п. Исторические данные — это накопленные данные о транзакциях за определённый промежуток времени. Примеры исторических данных: история по продажам, финансовые операции, данные по прибыли и потерям. Количество и уровень детализации исторических данных могут варьироваться в зависимости от функций системы и внешних требований.

2.3. Требования к переносу данных. При интеграции и замене программных приложений возникает необходимость переноса данных. Требования для переноса данных формируют архитектуру данных. Также на этом архитектурном уровне должны определяться системы преобразования и очистки данных, необходимые для представления данных в формате, который отвечает требованиям и ограничениям целевого приложения. Именно этот элемент архитектуры данных отвечает за качество данных.

2.4. Требования по управлению данными. Этот элемент описания архитектуры данных описывает ресурсы, которые есть на предприятии для проведения преобразований данных. Прежде всего, это:

- **структура** — организационная структура и органы стандартизации управления преобразованием данных;
- **люди** — навыки и роли, которые нужны для преобразования данных;
- **системы управления данными на протяжении их жизненных циклов.** В некоторых случаях можно выделить ещё один элемент архитектуры данных — требования по безопасности данных. Обеспечение безопасности данных можно определить, как сохранение данными свойств целостности, конфиденциальности и доступности. Этот элемент архитектуры данных описывает требования к различным категориям данных.

3. Архитектура приложений (прикладная архитектура), описывает приложения, с помощью которых осуществляется автоматизация деятельности организации (бизнес-архитектура) и обработка потоков информации (архитектура данных). Прежде всего, прикладная архитектура описывается через классы и типы приложений без привязки к конкретным решениям и технологиям. Классы приложений достаточно стабильны и не слишком сильно изменяются во времени, в то время как технологии реализации выделенных классов приложений могут быстро изменяться.

Как правило, прикладная архитектура содержит следующие элементы:

- классификацию приложений или предоставляемых ими сервисов;
- выделение и описание основных классов приложений или групп сервисов, используемых на предприятии;
- привязку классов приложений к основным бизнес-процессам организации;
- описание взаимодействия и взаимозависимости корпоративных прикладных программ;
- определение функциональных требований к классам приложений или группам сервисов: уровень использования и критичность для компании, обрабатываемые данные и поддерживаемые протоколы, требования по надёжности и интенсивности использования и т. д.
- приоритеты для совершенствования и развития существующих программных средств и приобретения/создания новых средств.

Кроме того, прикладная архитектура может быть описана не только в виде классов приложений, но и через конкретные системы (например, см. рис. 8).

4. Технологическая архитектура описывает физическое воплощение ИТ-инфраструкту-

ры предприятия и опирается на требования, полученные в ходе описания архитектуры приложений. Существуют различные способы категоризации технологий. Согласно Gartner, технологическая архитектура состоит из шести элементов:

- сервисы данных (СУБД, хранилища данных и т. д.);
- прикладные сервисы (почта, системы коллективной работы, средства разработки и т. д.);
- ПО промежуточного слоя (сервера приложений и другие средства интеграции);
- вычислительная инфраструктура (серверное и пользовательское оборудование, СХД и т. д.);
- сетевые сервисы (локальная и глобальная сетевая инфраструктура, технологии доступа), сервисы безопасности.

Архитектурный подход при внедрении бизнес-приложений — модель Захмана

Самой популярной архитектурной моделью, которая активно используется при описании не только архитектуры предприятия, но и архитектуры других систем, является модель Захмана. Первый вариант своей модели Джон Захман создал в 1987 г. для использования, прежде всего, при внедрении крупных бизнес-приложений.

Он предложил отображать архитектуру предприятия в виде матрицы, строки которой отражают взгляд на предприятие различных групп сотрудников предприятия и тех, кто используют результаты его деятельности: от инвесторов и высшего руководства до рядовых сотрудников. Так как владелец компании и рядовой сотрудник видят её с разной степенью детальности, можно сказать, что это взгляд на организацию с разной высоты (Захман назвал эти строки перспективами). **В модели Захмана выделяются следующие перспективы:**

1. Хозяин или планировщик (Planner) — тот, кто устанавливает направление развития организации: прогнозирует изменения внешней среды, область функционирования организации, цели её деятельности. Это могут быть инвесторы или собрание акционеров компании.

2. Руководитель (Owner) — тот, кто отвечает за функционирование предприятия: производство продуктов и/или предоставление услуг, управление затратами, выполнение персоналом своих обязанностей. Это топ-менеджер компании и владелец процессов. Руководитель должен понимать все основные компоненты предприятия как сложной системы, например: внешнюю среду, технологии, культуру, поставщиков, клиентов, законы, конкурентов, возможности, риски, деньги и т. д.

3. Проектировщик (Designer) — конструктор или системный архитектор, который является промежуточным звеном между желаемым состоянием организации (с точки зрения «владельца») и технически/физически возможным. Сюда относятся аналитики и проектировщики. Применительно к ИТ, это поставщики задач и бизнес-аналитики.

4. Подрядчик (Builder) — генеральный подрядчик, который обеспечивает производство конечного продукта или услуги. В рамках проекта — это руководитель проекта или системный аналитик.

5. Субподрядчик (Subconstructor) — тот, кто отвечает за построение и компоновку части конечного продукта или услуги, проектировщик и разработчик части конечного продукта или услуги. Например, это внешние компании, которые поставляют оборудование или программное обеспечение, или разрабатывают подсистему.

6. Пользователь — тот, кто использует продукты или услуги работающего предприятия. Применительно к ИТ — это рядовые сотрудники компании.

Для того, чтобы лучше запомнить аспекты (столбцы) матрицы Захмана, можно использовать детский стишок (Р. Киплинг, перевод С.Я. Маршака):

Есть у меня шестёрка слуг,
Проворных, удалых,
И все, что вижу я вокруг, —
Все знаю я от них.
Они по знаку моему являются в нужде.
Зовут их: **Как** и **Почему**,
Кто, **Что**, **Когда** и **Где**.

Столбцы же матрицы отражают различные стороны (или аспекты) деятельности системы, или в конкретном случае, предприятия, которые сгруппированы по ответам на вопросы: Кто, Что, Где, Когда, Как, и Почему. Для пояснения приведём примеры того, что относится к разным столбцам модели Захмана для предприятия.

Кто — это участники функционирования предприятия, люди и организации, в этом столбце отражаются организационная структура, потоки работ, участники и роли, инструкции;

Что — это объекты предприятия, включая используемые данные (оборудование, материалы, элементы и базы данных);

Где — это места выполнения процессов и функций предприятия, то есть пространственное расположение компонент системы (география, связи, сети и оборудование);

Когда — это временные характеристики работы предприятия (жизненные циклы, события, переходы состояний, графики, календари и расписания);

«Архитектура, что бы это ни было, это именно то, что наводит мосты между стратегией и её реализацией».

Джон Захман

Как — это процессы и функции предприятия, от общего перечня основных бизнес-процессов до конкретных программных модулей (в этом столбце отражаются спецификации, преобразования и ПО);

Почему (или Зачем) — это мотивация и порядок трансляции миссии и стратегии на нижние уровни (стратегии, желаемые результаты и параметры достижений).

Пример матрицы Захмана показан в Табл. 1.2.4.

Рис. 1.2.4. Модель Захмана.

	Данные ЧТО	Функции КАК	Размещение, сеть ГДЕ	Люди КТО	Время КОГДА	Мотивация ЗАЧЕМ	
Владелец	Основные компоненты организации	Список основных бизнес-процессов	Территориальное размещение организации	Ключевые заинтересованные стороны: партнеры, клиенты	Важнейшие события	Бизнес-цели и стратегия	Сфера действия контекст
Топ-менеджер	Концептуальная модель данных	Концептуальная модель бизнес-процессов	Система логистики	Оргструктура, роли в процессах, внешние стороны	Мастер-план	Бизнес-план	Модель предприятия
Бизнес-аналитик	Логистическая модель данных	Логистическая модель бизнес-процессов	Архитектура предприятия	Логистическая ролевая модель процессов	Расписания	Бизнес-правила	Логистическая модель программной системы
Системный аналитик, проектировщик	Физическая модель данных	Технический проект	Архитектура ИТ	Роли в программной системе и их права	Циклы, ожидания, временные параметры	Физическая модель правил	Техническая модель программной системы
Разработчик, программист	База данных	Программный код, готовые компоненты	Интерфейсы	Авторизация, аутентификация	Программирование и настройка циклов	Реализация бизнес-логики	Программная система
Пользователь	Ввод и вывод в нужном виде	Работающая программная система	Способы доступа к программной системе	Логин, пароль	Время ожидания	Улучшение производительности, удобство	Пользователь, работающее предприятие

Захман писал: «Вопросы что, как, где, кто, когда и зачем существуют тысячи лет. И они будут существовать ещё тысячи лет. Требования к системам, процесс проектирования, производства или концептуальный, логический и физический уровень описания также существуют в течение тысяч лет и будут существовать ещё тысячи лет. Логическая структура классификации и схема — неизменны».

Каждая клетка матрицы содержит соответствующее описание конкретной стороны деятельности предприятия (аспекта) на конкретном уровне (в конкретной перспективе) в виде определённой модели или, возможно, набора соответствующих документов и других материалов. Клетки каждой строки вместе образуют полное описание системы на выбранном уровне (с выбранной перспективы).

При этом, порядок следования столбцов не существен. А вот заполнение клеток рекомендуется проводить последовательно

«сверху вниз», попытка пропуска одной из строк почти всегда приводит к ошибкам. Выбор перспектив (строк) и аспектов (столбцов) для модели зависит от цели её построения и не является жёстко заданным. Например, к перспективам могут быть добавлены клиенты и служба поддержки.

Архитектура как процесс

У термина «архитектура» существуют две стороны, которые дополняют друг друга. Одна — архитектура как описание, статический слепок некоторой сложной системы. И вторая — архитектура как процесс, набор руководств и правил, которые определяют построение новых подсистем сложной системы, а, следовательно, её развитие. Здесь уместно вспомнить определение Gartner Group: «Архитектура — это глагол, а не существительное». Архитектура как процесс — это один из важнейших процессов предприятия. Архи-

тектура предприятия находится в постоянном развитии от текущей архитектуры предприятия к целевой. Разовое создание набора архитектурных моделей, которые непонятно как использовать, не принесёт компании выгоды. Именно поэтому все больше появляется не только архитекторов, но и целых подразделений, которые на постоянной основе занимаются моделированием и развитием архитектуры предприятия.

Развитие архитектуры предприятия осуществляется в выбранном стратегическом направлении и на основании видения и принципов развития организации. Институт разработки архитектуры предприятий (Institute for Enterprise Architecture Development, IFEAD) обобщает основные руководящие принципы архитектуры предприятия следующим образом: «нет стратегических прогнозов — нет архитектуры предприятия». То есть архитектура предприятия — это целостный взгляд, который объединяет элементы бизнеса и технологии, исходя из общего стратегического прогноза развития предприятия.

Процесс планирования архитектуры Стивена Спивака. Спивак, как и Захман, работал в IBM, был озабочен теми же проблемами и обратился к архитектурным подходам лишь на год позже Захмана. Он соединил модель Захмана с методикой планирования и формирования целевой архитектуры. В основе метода Спивака лежит процесс планирования архитектуры предприятия (Enterprise Architecture Planning, EAP), направленный на разработку архитектуры, оптимально использующей информацию для поддержки бизнеса. Процесс EAP направлен на определение того, какие данные, приложения и технологии наиболее полно отвечают потребностям конкретного предприятия. Основные этапы EAP модели Спивака

приведены на рис. 4. Однако в полной мере процесс планирования и изменения архитектуры предприятия нашёл отражение в методологии TOGAF.

Практические шаги по моделированию архитектуры предприятия. На практике процесс работы с архитектурой предприятия состоит из пяти шагов:

- 1. Определение общих контуров (видения) архитектуры** — какой должна быть архитектура предприятия и зачем мы её описываем, а также рамки разработки архитектуры, заинтересованные лица и план работ.
- 2. Построение модели архитектуры «как есть».**
- 3. Построение модели архитектуры «как должно быть».**
- 4. Определение шагов, которые позволят перейти от «как есть» к «как должно быть».** План изменений архитектуры чаще всего содержит более одного предлагаемого решения, в результате чего может возникнуть несколько вариантов изменения архитектуры.
- 5. Формирование путей дальнейшего развития архитектурной модели,** т. е. определение правил дальнейшей детализации и актуализации модели, дополнения её конкретными архитектурными шаблонами, описаниями типовых элементов (схем процессов, схем баз данных, программ), каталогами и базами реализаций типовых элементов,

Рис. 1.2.5. Семь шагов архитектурного процесса Спивака.



примерами применения типовых элементов. Эта последовательность шагов показана на Рис. 1.2.5.

Кроме того, необходимо сделать архитектурную модель общим инструментом бизнеса и ИТ, используя её для подготовки как стратегических, так и оперативных решений. Это предполагает:

- её доступность и информирование всех заинтересованных лиц о её наличии;
- её понятность — важно уже на начальном этапе архитектурного процесса согласовать словарь и формулировать принципы создания модели предприятия, которые должны быть понятны всем топ-менеджерам компании;
- её адекватность ситуации на предприятии — поддержка её актуального состояния.

Модель TOGAF

Один из наиболее популярных и развитых фреймворков описания архитектуры принадлежит международному консорциуму The Open Group и так и называется: The Open Group Architecture Framework (TOGAF), то есть «Архитектурный фреймворк, созданный The Open Group». TOGAF является промышленным стандартом на описание архитектуры предприя-

тия, который может бесплатно использоваться любой организацией, разрабатывающей собственную архитектуру. TOGAF постоянно развивается с середины 90-х годов, в настоящее время актуальна его 9-я версия.

Примечательно определение архитектуры предприятия, данное Open Group «Архитектура предприятия — это способ понимания различных элементов, которые в совокупности составляют предприятие, и то, как эти элементы взаимосвязаны».

Фреймворк TOGAF в качестве основных компонентов архитектуры предприятия рассматривает уже знакомые нам четыре типа архитектур: архитектуру бизнеса, архитектуру данных, архитектуру приложений и технологическую архитектуру. Архитектура данных и архитектура приложений объединены в архитектуру ИС. И этап разработки модели ИС состоит из 2-х частей: разработки модели архитектуры данных и модели архитектуры приложения.

Основное отличие TOGAF от других архитектурных фреймворков заключается в том, что он фокусируется на процессе разработки архитектур, и кроме информационной базы стандартных архитектурных моделей, его основу составляет подробно описанный по-

шаговый итеративный метод разработки архитектуры, который называется Architecture Development Method (ADM). Основные его компоненты и этапы приведены на Рис. 1.2.6.

ADM — циклический, итеративный процесс, состоящий из 9 фаз. На каждой итерации должны быть выбраны следующие решения:

- Широта охвата предприятия
- Уровень детализации
- Временной горизонт
- Архитектурные активы организации, включая:

The Open Group — некоммерческий консорциум, сформированный при объединении X/Open с Open Software Foundation в 1996 году, для разработки и продвижения вендору-независимых открытых технологических стандартов в области ИТ. Сейчас в него входит более 500 крупных организаций из разных стран, как покупателей, так и производителей информационных технологий, а также правительственные агентства, в частности, Capgemini, Fujitsu, Sun Microsystems, Hitachi, Hewlett-Packard, IBM, NEC, US Department of Defense, NASA и другие. The Open Group наиболее известен как сертифицирующий орган для торговой марки UNIX. Он является автором Single UNIX Specification, расширяющей стандарты POSIX и являющейся официальным определением UNIX. The Open Group считает своей задачей обеспечить достижение целей бизнеса с помощью ИТ-стандартов. Девиз этого консорциума — «Безграничный поток информации»

- то, что было создано на предыдущих итерациях;
- активы доступные на других предприятиях отрасли (другие фреймвоки, системные модели).

Эти решения должны быть сделаны на основе практического анализа ресурсов и доступных компетенций и выгод, которые ожидаются от такого архитектурного описания. ADM показателен своей цикличностью, повторяемостью шагов, характеризующих постоянное совершенствование архитектуры предприятия, связью между компонентами и центральным местом, которое уделяется требованиям бизнеса. Как видно из Рис. 1.2.5, центральное место в процессе TOGAF занимают требования. Управление требованиями ADM работает с любыми видами требований, в частности, с требованиями новых функциональных возможностей и изменений бизнеса. И поскольку требования постоянно изменяются, управление требованиями ведётся на протяжении всего жизненного цикла архитектуры предприятия, который совпадает с жизненным циклом самого предприятия. В соответствии с последней версией TOGAF процесс разработки архитектуры (ADM) состоит из девяти фаз:

1. Подготовительная фаза: подготовительная деятельность, направленная на выявление бизнес-требований для целевой архитектуры предприятия («как должно быть»), включая основные принципы, адаптацию методики под особенности предприятия и выбор средств описания архитектуры.

2. Фаза А: Архитектурное видение — начальная фаза цикла разработки архитектуры. Здесь описываются рамки процесса разработки архитектуры, определяются заинтересованные лица, формируется видение того, какой должна быть архитектура предприятия, утверждаются архитектурные принципы, видение и план работ.

Рис. 1.2.6. Фазы ADM TOGAF.



3. Фаза В: Архитектура бизнеса — разработка бизнес-архитектуры предприятия, основанная на согласованных на предыдущем шаге, принципах и видении архитектуры. Описание существующей бизнес-архитектуры и формирование целевой.

4. Фаза С: Архитектура информационных систем — разработка архитектуры данных и архитектуры приложений. Описание существующих архитектур данных и приложений и формирование целевых.

5. Фаза D: Технологическая архитектура — описание существующей технологической архитектуры и формирование целевой.

6. Фаза Е: Проверка возможностей реализации решений, предложенных для построения целевой архитектуры предприятия. Это база для начального планирования реализации и выявления двигателей (стимулов) процесса построения целевой архитектуры, определённой на предыдущих фазах, выра-

женная в возможностях её реализации и решении основных вопросов.

7. Фаза F: Планирование перехода к целевой архитектуре — формирование последовательности подробных переходных архитектур и разработка плана миграции.

8. Фаза G: Управление построением целевой архитектуры и её контроль — формирование системы руководства преобразованием архитектуры предприятия (Implementation governance), которая предполагает создание «Совета по архитектуре» и стратегии соответствия архитектуре, которая, в свою очередь, определяет правила оценки проектов в части соответствия целевой архитектуре.

9. Фаза H: Управление изменениями — процедуры для управления изменениями новой архитектуры.

Каждая из девяти фаз разбивается на подэтапы, отдельные работы, которые необходимо выполнить, и содержит перечень входных

и выходных документов.

Кроме процесса разработки архитектуры, модель TOGAF включает коллекцию компоновочных блоков (шаблонов), названную «континуум предприятия» (Enterprise Continuum). TOGAF подразумевает, что эта коллекция предоставляет коллективам, занимающимся архитектурой предприятия, соответствующие шаблоны архитектур, модели и процессы, из которых можно собирать готовые решения, как в LEGO. Кроме этого, модель TOGAF содержит описание типичных выходов архитектурного процесса, средств для оценки этих выходов и самого архитектурного процесса, и требования к персоналу, участвующему в моделировании и построении архитектуры.

В целом модель TOGAF является цельным и подробным методом формирования и развития архитектурного процесса на предприятиях и используется во многих организациях в разных странах мира.

Стандарты архитектуры предприятия

Стандарты архитектуры предприятия определяют понятия, концепции, основные подходы, требования, применяемые для анализа и развития предприятия, как сложной системы, включающей социальные, экономические и технологические элементы, которые объединены общим предназначением. Важнейшими стандартами, относящимися к архитектуре предприятия, являются стандарты ISO 14258:1998 (с изменениями от 2000 г.) и ISO 15704:2000.

Стандарт ISO 14258 — «Промышленные автоматизированные системы. Концепции и правила для моделей предприятия» (Industrial automation systems— Concepts and rules for enterprise models) появился в 1999 г., а в 2008 г. его перевод был принят как ГОСТ Р. В стандарте содержатся концепции и правила для моделирования производственных пред-

приятий с целью обеспечения более эффективной интеграции элементов предприятия, в частности, производственных процессов. Стандарт отмечает необходимость системного подхода к предприятию, определяя его как «социально разнородную систему, определяемую свойствами и характеристиками людей и машин». В соответствии со стандартом ISO 14258 архитектура и методики уровня предприятия должны включать в своё содержание роли людей, описание процессов (функции и поведение) и представление всех вспомогательных технологий на протяжении всего жизненного цикла предприятия.

Стандарт ISO 14258 определяет наиболее общие правила для моделирования и построения архитектуры предприятия. Любая система, в том числе предприятие, имеет жизненный цикл, который, следуя стандарту,

Табл. 1.2.2. Связь между стадиями жизненного цикла системы и видами деятельности.

Наименование стадии	Вид деятельности, предусматривающий определение того, что делать	Вид деятельности, предусматривающий определение того, как делать	Вид деятельности, предусматривающий выполнение
Стадия планирования и создания (например, до передачи права на продажу/покупку)	• Разработка целей. • Определение стратегии. • Определение потребностей в продукте.	• Разработка требований. • Определение концепции. • Проектирование продукта. • Планирование производства продукта. • Планирование обеспечения продукта.	• Закупка частей. • Производство продукта. • Испытание продукта. • Отгрузка продукта.
Стадия эксплуатации и деятельности (например, после передачи права на продажу/покупку)	• Определение потребностей в обеспечении. • Определение использования.	• Определение требований к эксплуатации. • Определение требований к поддержке.	• Эксплуатация продукта. • Поддержка продукта.
Стадия рециклинга и утилизации (например, после истечения полезного срока службы продукта)	Определение потребностей в рециклинге/утилизации.	Определение требований к рециклингу/утилизации.	• Рециклинг продукта. • Утилизация продукта.

можно разделить на три фазы: планирование/создание, использование/эксплуатация и повторное использование/утилизация. На каждой фазе жизненного цикла стандарт предлагает выполнить три действия:

1. Ответить на вопрос, что надо сделать — **W (What)**.
2. Ответить на вопрос, как надо сделать — **H (How)**.
3. После этого сделать то, что надо — **D (Do)**.

В Табл. 1.2.2 отображена связь между фазами жизненного цикла системы (продукта, проекта, программной системы, предприятия и т. д.) и этими действиями. В каждой клеточке указана информация, которая должна поддерживать каждую фазу/действие. Кроме того, стандарт ISO 14258 также выделяет статические и динамические модели. Например, такие модели как TOGAF несомненно

относятся к динамическим, а модель Захмана — к статическим. Стандарт выдвигает требования к моделям, к их информационному обеспечению и взаимодействию.

Стандарт ISO 15704 «Промышленные системы. Требования к стандартным архитектурам и методологиям предприятия» (Industrial automation systems. Requirements for enterprise-reference architectures and methodologies). Стандарт ISO 15704 появился в 2000 году и закрепил основные положения современного подхода к архитектуре предприятия. В 2008 г. перевод этого стандарта был принят как ГОСТ Р. С учётом его положений в 2006 году был выпущен стандарт ISO 19439 «Enterprise integration — Framework for enterprise modelling» (Интеграция предприятия — Рамочная структура для моделирования предприятия). В 2005 году в стандарт ISO 15704 включены «Дополнительные представ-

ления с точки зрения пользователя». Его цель — определить требования к архитектурной модели и архитектурному процессу, обеспечивающие полноту архитектурной модели для текущих и стратегических целей предприятия и интеграции его частей в единое целое. Хотя стандарт ISO 15704 предназначен для определения требований к архитектурам и методологиям промышленного предприятия, в нем особо оговаривается возможность применения положений стандарта для всех предприятий.

Стандарт ISO 15704 создан комитетом по автоматизации предприятий (рабочей группой по архитектуре, связям на предприятии и его интеграции), поэтому в центре внимания стандарта предприятие как система, состоящая из отдельных интегрируемых частей.

Рис. 1.2.7. Этапы жизненного цикла GERA для любого предприятия или производственного объекта.



Стандарт ориентирован как на людей, так и на технологии и предостерегает от рассмотрения интеграции только на уровне информационных систем и систем управления. Проблемы интеграции связаны с определением миссии компании, производственной деятельностью, производством продукции и оказанием услуг, человеческим фактором и организационной структурой.

В стандарте определены **два класса функций предприятия**:

1. Функции, связанные с выполнением миссии, то есть процессами производства продукции или оказания услуг.
2. Функции управлением выполнением миссии для обеспечения жизнеспособности и успешной работы предприятия.

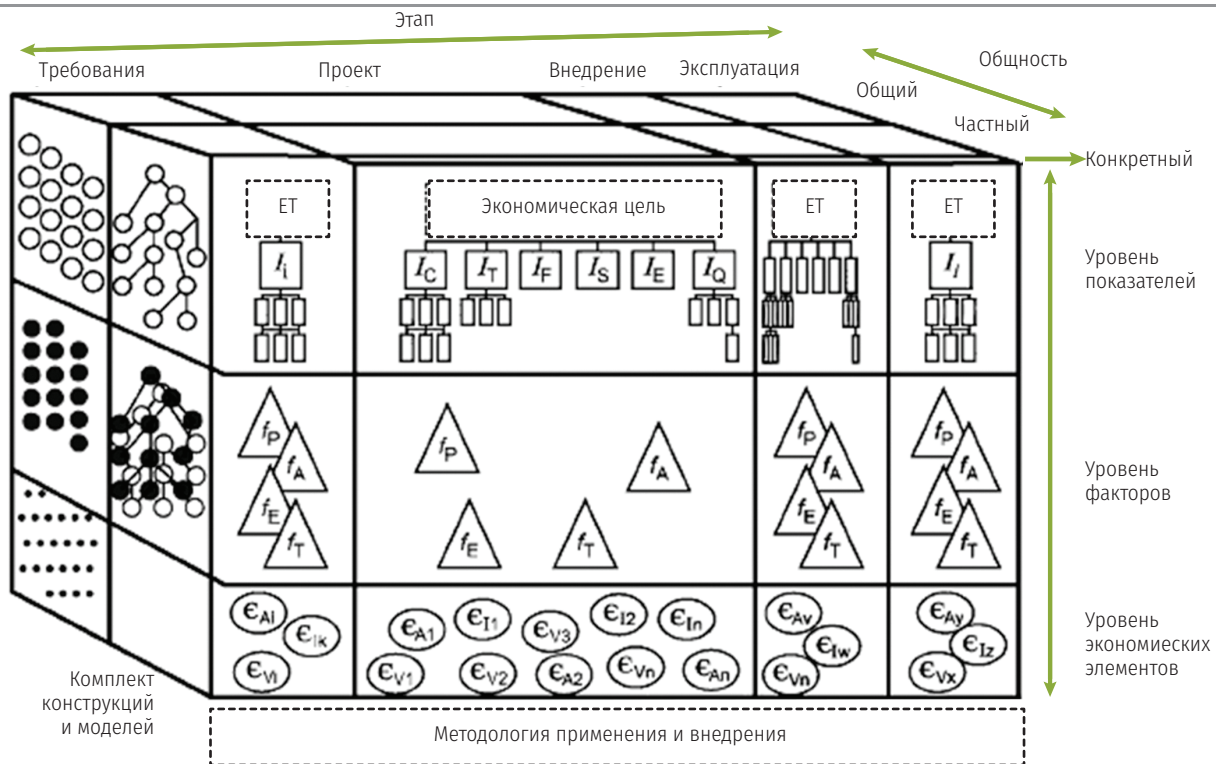
В соответствии со стандартом, каждому предприятию необходим план действий для достижений стратегических и оперативных целей, стоящих перед ним, включающий:

- a) описание необходимых задач;
- b) определение необходимого объёма информации;
- c) взаимодействие между персоналом, процессами и оборудованием с учётом рассматриваемой интеграции;
- d) определение вопросов менеджмента;
- e) принятие во внимание соответствующих экономических, культурных и технических факторов;
- f) подробное определение необходимой степени компьютерной поддержки;
- g) моделирование процессно-ориентированной поддержки, способной моделировать всю историю жизни предприятия.

Использование этого стандарта позволяет проверить соответствие архитектуры предприятия стоящим перед ним целям и задачам и выработать эффективную методику управления предприятием.

В приложении к стандарту приведена модель GERAM (Generalised Enterprise Reference

Рис. 1.2.8. Основа экономического изображения.



Architecture and Methodology) — обобщённая стандартная архитектура предприятия и методологии, разработанная целевой группой IFIP/IFAC по архитектурам интеграции предприятия. Основным компонентом GERAM является GERA (Generic Enterprise Reference Architecture) — общая стандартная архитектура предприятия.

На Рис. 1.2.7 приведены этапы жизненного цикла любого предприятия в соответствии с GERA. Кроме того, в другом приложении к стандарту описано Экономическое представление в архитектуре системы CIM (Computer-Integrated Manufacturing) — модель архитектуры информационных систем, основанная на системах автоматического проектирования и производства. С точки зрения стандарта, экономическое представление содержит систему моделей экономических компонентов и их связей, описанных различными способами. Для улучшения интеграции компонентов предприятия и его успешной работы пред-

приятия предлагается трёхуровневая среда, основанная на методах моделирования предприятия и эталонных моделях в общей эталонной архитектуре предприятия, как показано на Рис. 1.2.8.

Примечательна также схема иерархии перспективных инвестиций, приведённая в этом приложении (Рис. 1.2.9), показывающая вклад передовых информационных технологий в непрерывный рост доходов предприятия. И ещё в одном приложении приведена модель представления принятия решений, основанная на выделении 3 доменов: продукция, ресурсы и время и отражающая важнейшую роль в этом процессе информации.

Стандарты эталонных архитектурных моделей для новых технологий

Отдельно стоит обратить внимание на стандартизацию таких относительно новых, но стремительно развивающихся областей как облачные технологии и Интернет вещей. Примечательно,

Рис. 1.2.9. Иерархия перспективных инвестиций.



что для первой из них ещё в 2014 году появились стандарты ISO/IEC: 17788 «Информационные технологии — Облачные вычисления — Общие положения и словарь», ISO/IEC 17789 «Информационные технологии — Облачные вычисления — Эталонная архитектура», определяющие основные роли и подроли облачных вычислений, связи между ними, а также сквозные аспекты облачных вычислений, которые должны обеспечиваться всеми участниками.

Аналогичная работа по созданию модели эталонной архитектуры идёт сейчас в области Интернета вещей. В ближайшее время должны появиться стандарты ISO/IEC 30141. Information technology — Internet of Things Reference Architecture (IoT RA) и ISO/IEC 20924, Information technology — Internet of Things — Definition and Vocabulary. Таким образом, в области новых технологий основополагающие стандарты связаны с эталонной архитектурной моделью.

Общие принципы построения архитектурной модели

Построение архитектурной модели — архитектурный процесс — подчиняется определённым принципам. Важнейшие из них:

1. Принцип постепенной детализации. Следует начинать с взгляда на предприятие с большой высоты (перспектива «хозяин» в модели Захмана или архитектура бизнеса в модели TOGAF и др.), а не с подробного описания одной или нескольких нижележащих элементов. Это связано с тем, что, описывая какой-либо из нижележащих элементов общей модели предприятия, мы не получим единого системного взгляда на предприятие, которое призвана дать его архитектурная модель. По мере необходимости детализацию можно углублять, но это должно определяться тем, как будет использоваться архитектурная модель.

2. Принцип согласованности слоёв. В случае, если за основу взято наиболее популярное представление архитектуры предприятия как «слоёного пирога» (четыре уровня в модели TOGAF или «перспективы» в модели Захмана), необходимо добиваться согласованности слоёв. Ведь цель построения архитектурной модели — получить единую и взаимосвязанную картину предприятия. Поэтому не стоит описывать слои по отдельности, поручая эту работу различным подразделениям. Лучше выделить процесс или на первых порах — проект

по описанию архитектуры предприятия и создать единую команду.

3. Принцип независимости слоёв. Вместе с тем, слои (уровни, «перспективы») должны быть независимы. Возможно выделение любых необходимых слоёв или уровней (например, архитектура интеграции, определяющая принципы взаимодействия и интеграции приложений, данных и бизнес-процессов в распределённой среде компании в модели Framework (бывший NGOSS), или архитектура безопасности). Поэтому при их выделении применяют следующие условия:

- в случае если нижний слой вышел из строя, верхний работать не может;
- неработоспособность верхнего слоя не влияет на работоспособность нижнего;
- работоспособность элементов внутри одного слоя может влиять на работоспособность других элементов этого же слоя, а может и не влиять.

4. Принцип полноты. Модель архитектуры предприятия должна описывать предприятие с требуемой полнотой. В то же время, не следует увлекаться большим количеством архитектур и уровней, так как это усложняет модель. Количество архитектур и уровней должно определяться конкретными задачами, для решения которых строится архитектурная модель.

5. Принцип непротиворечивости. Элементы архитектурной модели не должны противоречить друг другу.

6. Принцип отсутствия дублирования. Элементы архитектурной модели не должны дублировать друг друга.

Принцип постоянной трансформации текущей архитектуры предприятия. Не следует забывать, что любое предприятие находится в постоянном развитии.

А значит, его архитектурная модель полезна только тогда, когда она актуальна и постоянно приводится в соответствие с реальным

состоянием предприятия.

С чего начать?

Если CIO ограничен во времени и ресурсах, то не стоит сразу браться за сложные архитектурные фреймвоки. Намного лучше для этого подходит модель Захмана. В самом начале процесса надо выбрать те уровни взгляда на предприятие (слои), которые для вас наиболее важны. В этом случае вы сосредоточитесь на основных проблемных областях, где использование архитектурного подхода может принести наибольшую пользу, при этом не распыляя свои усилия.

Нередко у того, кто приступает к описанию архитектуры предприятия, возникает соблазн ограничиться только уровнем архитектуры приложений. Однако, важно понимать, что описание только уровня архитектуры приложений приведёт к неверным результатам. Для корректной привязки приложений к основным бизнес-процессам организации необходим их перечень и категоризация, что относится к уровню бизнес-архитектуры, а также логическая структура данных (какие бизнес-объекты есть и связи между ними) и информационные потоки, что описывается на уровне информационной архитектуры.

Кроме того, в дополнение к четырём традиционным уровням архитектуры мы рекомендуем выбрать уровни взгляда на предприятие (слои), которые важны для вашего предприятия. Например, в случае холдинга полезны взгляды на уровнях корпоративного центра, филиалов или территориальных объединений и отдельных предприятий. Как может выглядеть описание архитектуры приложений, для этих уровней показано на Рис. 1.2.10.

Такая простейшая модель уже позволяет обсуждать с руководством компании стратегию развития ИТ и наметить, чего хочется достичь за период планирования. Прежде всего, расставить приоритеты заполнения «белых пятен» неавтоматизированных бизнес-процессов. Или, например, добиться того, чтобы

одна ERP-система покрывала большую часть потребностей в автоматизации. Или чтобы все компании использовали для бухгалтерского учёта одну и ту же систему, и её версию. Если у компании есть филиалы, то в общем случае стоит задать руководству два вопроса, которые относятся к архитектуре бизнеса:

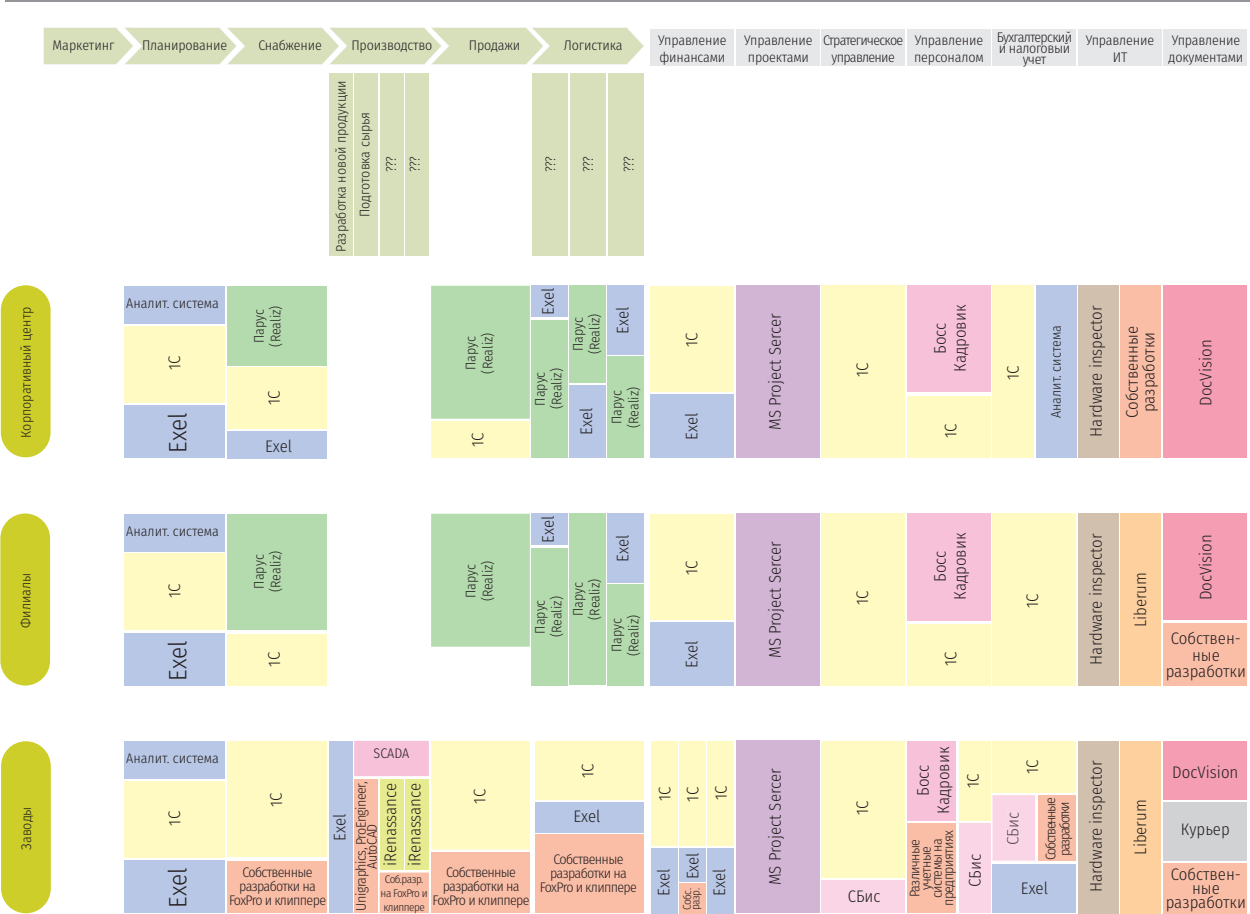
- 1. **Степень централизации** — насколько тесно связаны между собой подразделения (филиалы) организации.
- 2. **Степень унификации** — насколько одинаковы должны быть ИТ-системы, используемые для схожих бизнес-процессов. Это не такой простой вопрос, как кажется. Ведь если унификация позволяет сократить затраты на ИТ, то специализация позволяет получить конкурентные преимущества и повысить гиб-

кость организации. Поэтому далеко не всегда цветовая однородность Рис. 1.2.10 — это цель, к которой должна стремиться компания. Ответы на эти вопросы существенно повлияют на выбор целевой архитектурной модели.

Инструменты построения архитектурной модели

Традиционно для моделирования архитектуры предприятия использовали методы описания бизнес-процессов. Например, семейство языков моделирования IDEF (Integrated Computer Automated Manufacturing Definition), которое возникло в середине 70-х годов в ВВС США, как решение проблемы повышения производительности и эффективности информационных технологий. Часть стандартов этого семейства может быть использо-

Рис. 1.2.10. Пример архитектурной модели Холдинга.



вана при построении моделей архитектуры бизнеса, архитектуры данных и прикладной архитектуры.

Язык, использованный в инструментах линейки ARIS (Architecture of Integrated Information Systems) также позволяет описывать бизнес-процессы. Кроме того, в эту линейку входит отдельное решение, которое называется Enterprise Architecture Management (Управление архитектурой предприятия) и предназначено для создания архитектурных моделей предприятия и поддержки их в актуальном состоянии.

BPML (Business Process Modeling Language) — язык моделирования бизнес-процессов был разработан для описания архитектуры Бизнеса. BPML представляет бизнес-процессы как комплекс взаимодействий управляющих потоков, потоков данных и потоков событий и средств моделирования бизнес-правил, ролей их взаимодействий. Существует много программных систем, которые предоставляют возможность описания архитектуры бизнеса на этом языке, в частности, Popkin System Architect.

Язык UML подходит для описания всех слоёв архитектуры предприятия. Эту нотацию поддерживают многие инструментальные средства от ARIS до Microsoft Visio и Rational Rose. В 2008 г. консорциум Open Group приобрёл права на развитие языка описания архитектуры предприятия Archimate, который разрабатывался в Нидерландах по заказу голландского правительства, ряда государственных и крупных частных компаний, а также образовательных организаций. Первая версия международного стандарта появилась в 2009 г., стандарт постоянно развивается, с июня 2016 г. актуальна его 3-ья версия. Archimate полностью совместим с TOGAF. Так же, как и в TOGAF, в нем выделяются 4 уровня описания архитектуры предприятия: архитектура бизнеса, архитектура данных, архитектура программных приложения и технологическая

архитектура. Archimate представляет собой зрелый стандарт моделирования архитектуры предприятия, поскольку основан на большой аналитической и научной работе представительной группы экспертов и учёных и опирается на вышеописанные средства моделирования, существовавшие до его появления. Archimate прост для понимания и освоения. Множество программных продуктов позволяют моделировать архитектуру предприятия на этом языке, включая бесплатно-распространяемое ПО. Мало того, даже те инструменты моделирования, которые появились до Archimate, дают возможность импортировать и экспортировать модели в этот стандарт.

Основу Archimate составляют три типа элементов:

- активный структурный элемент, который способен выполнять определённые действия. Это могут быть бизнес-исполнители, компоненты приложений или устройства, которые исполняют те или иные действия. По аналогии с естественным языком это — подлежащее.
- элемент поведения, который определяется как некоторые действия, выполняемые одним или несколькими активными структурными элементами. Элементами поведения являются процессы, функционалы, сервисы и события. В естественном языке это — глаголы.
- пассивный структурный элемент, который определяется как объект, на котором или с которым выполняются действия. Обычно это информационные объекты или объекты данных, но также они могут быть использованы для представления физических объектов, над которыми выполняются те или иные действия. По аналогии с естественным языком это — дополнения.

Кроме того, можно выделить такие элементы как сервис — функциональность, которую система предоставляет своему окружению,

скрывая при этом внутренние операции, и интерфейс — точка доступа, в которой сервис становится доступным внешнему окружению, присутствующие на разных архитектурных уровнях.

Варианты архитектур по определённым признакам группируются в стили также как в архитектуре зданий. Остановимся на двух современных архитектурных стилях: сервисная архитектура и архитектура, управляемая моделями. Заметим, что к прикладным архитектурным стилям также часто относят два противоположных подхода: все на базе «единой системы» или набор «лучших в своём классе» приложений.

Сервисно-ориентированная архитектура (Service Oriented Architecture, SOA)

SOA — это стандарт архитектуры предприятия, прежде всего, архитектуры приложений. Он направлен на решение проблем интеграции разнообразных программных систем в единую программную среду предприятия. Стандарт основан на понятии сервиса, т.е. удовлетворения потребностей одного объекта другим. Согласно Gartner Group, под сервисно-ориентированной архитектурой понимается подход к проектированию прикладных информационных систем, который руководствуется следующими принципами:

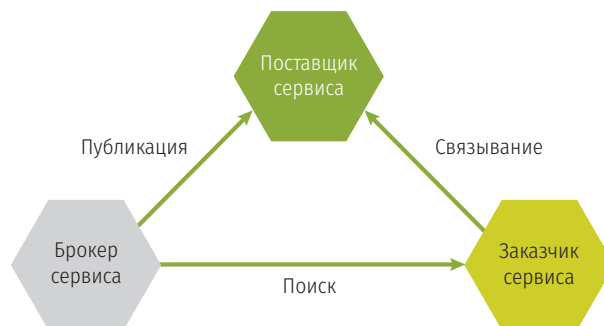
- явное отделение бизнес-логики прикладной системы от логики презентации информации;
- реализация бизнес-логики прикладной системы в виде некоторого количества программных модулей (сервисов), которые доступны потребителям в режиме «запрос-ответ» через чётко определённые формальные интерфейсы доступа;
- потребитель сервиса может быть прикладной системой или другим сервисом и вызывает сервис, используя соответствующие коммуникационные механизмы.

То есть в сервисно-ориентированной архитектуре любой компонент предоставляет свою функциональность другим компонентам на основании простых правил, иллюстрацией которых служит Рис. 1.2.11.

На рисунке представлены три основные роли этого процесса: заказчик сервиса, который хочет получить определённый функционал, не зная, кто может его предоставить, поставщик, который может предоставить запрошенный функционал и брокер (посредник), который связывает заказчика и поставщика, на основании запроса от заказчика и информации о доступных сервисах, которые ему предоставляет поставщик.

Заметим, что интерфейсы, по которым происходит взаимодействие этих трёх объектов, не должны зависеть от используемых аппаратных платформ, операционных систем или языков программирования, используемых для разработки самих сервисов. Это позволяет сервисам взаимодействовать единым стандартным и универсальным способом. Такое взаимодействие получило название «слабосвязанное» взаимодействие. Его очевидным преимуществом является гибкость и адаптируемость, поскольку замена или модернизация одной из компонент системы не сказывается на остальных. В отличие от так называемого «сильно связанного» взаимодействия, при котором компоненты приложений взаимодействуют между собой по уникальным, написанным специально для конкретного случая интер-

Рис. 1.2.11. SOA.



фейсам. В этом случае модернизация одной компоненты потребует изменения всех компонент системы, которые с ней взаимодействуют. То есть сервисно-ориентированный подход к архитектуре приложений позволяет перейти от жёсткой архитектуры к, обеспечивающей возможности создания новых систем из набора доступных сервисов, т. е. более гибкой и динамичной.

SOA подробно описан в TOGAF и является одной из рекомендованных стандартных архитектурных моделей этого фреймвока.

Один из вариантов SOA — стандарт web-сервисов, предложенный в 2004 г. компаниями Microsoft и IBM. Web-сервисы опираются на три стандарта:

- **SOAP** — Simple Object Access Protocol, простой протокол доступа к объектам; протокол обмена сообщениями, основанный на XML, используемый для взаимодействия между участниками предоставления / получения сервиса.
- **WSDL** — Web Services Description Language, язык описания веб-сервиса и доступа к ним, также основанный на XML.
- **UDDI** — Universal Discovery, Description and Integration, универсальный интерфейс распознавания, описания и интеграции; каталог веб-сервисов и сведений о компонентах, предоставляющих веб-сервисы, описанные на WSDL.

Рис. 1.2.12 поясняет роль этих понятий в архитектуре web-сервисов.

SOA лежит также в основе эталонной архитектурной модели облачных вычислений.

Архитектура, управляемая моделями (Model Driven Architecture, MDA)

Архитектура, управляемая моделями, так же, как и сервисно-ориентированная архитектура, является стандартом архитектуры приложений. Она не противоречит SOA, эти архитектуры могут использоваться одновременно

Рис. 1.2.12. Архитектура Web-сервисов.

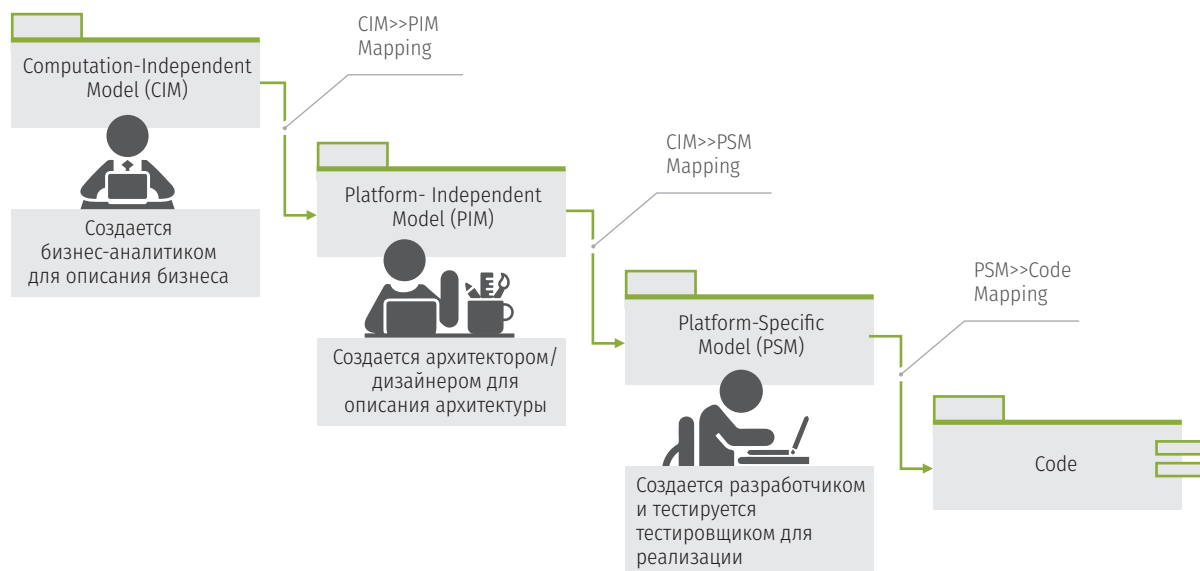


на одном предприятии для построения его единой информационной среды. Архитектуру, управляемую моделями с 2001 г., развивает международный консорциум OMG (Object Management Group). Прежде всего, архитектура MDA предназначена для проектирования программных систем и состоит из спецификаций, написанных в основном на UML, и руководств по их использованию.

Идея MDA состоит в том, что система может строиться как последовательность взаимосвязанных моделей, которые должны последовательно детализировать систему, начиная с независимой от технологий модели CIM, через платформенно-независимую модель PIM, затем – платформенно-зависимую модель PSM и к конкретной программной реализации. Структура MDA показана на Рис. 1.2.13.

Каждый тип моделей соответствует определённому уровню развития программной системы:

- исходной является так называемая «независимая модель вычислений» (Computational Independent Model, CIM), где аналитиком описывается бизнес-логика системы;
- далее она трансформируется в платформенно-независимую модель (Platform-Independent Model, PIM), в которой системе описывают уже ИТ-архитекторы;
- далее она трансформируется в платформенно-специфичную модель (Platform — Specific Model, PSM), создаваемую разра-

Рис. 1.2.13. Model Driven Architecture.

ботчиком системы и уже использующей преимущества выбранной платформы и обходящей её недостатки к программному коду;

- платформенно-специфичная модель автоматически или с минимальным участием программиста приводится к исполняемому коду и соответствующим структурам данных.

Таким образом, мы получаем три подуровня прикладной архитектуры. Уровень CIM лежит ближе к архитектуре бизнеса, код — к инфраструктуре. Стандарт MDA содержит правила перехода от уровня CIM к PIM и от уровня PIM к PSM для некоторых платформ, в число которых входят CORBA, J2EE, .NET.

В заключение хочется отметить, что в послед-

ние годы в нашей стране активизировалось развитие архитектурного подхода. Крупнейшие компании имеют в своём составе архитектурные подразделения, которые приобретают всё больший вес и все чаще участвуют в принятии всех важнейших решений, касающихся не только ИТ, но и бизнеса.

В 2016 г. была создана Национальная Ассоциация Архитекторов Предприятий, которая в настоящее время занимается вопросами сертификации отечественных архитекторов. Хотим также отметить, что в 2014 г. был утверждён приказом Минтруда профессиональный стандарт Архитектор программного обеспечения, который охватывает один из архитектурных слоёв (или одну из перспектив), описанных выше.

Часть 1. Введение

Глава 1.3

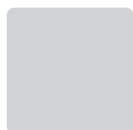
Бизнес-ценность ИТ



Сергей
Македонский



Владимир
Ананьин



Константин
Зимин

Сокращения и термины

BVIT — Business Value of IT, бизнес-ценность ИТ (Tony Murphy)

TCO — Total Cost of Ownership, общая стоимость владения

ROI — Return on Investment, возврат на инвестиции

DCF — Discounted Cash Flow, дисконтированный денежный поток

NPV — Net Present Value, чистая текущая стоимость

IRR — Internal Return Rate, внутренняя норма доходности

BSC — Balanced Score Card, сбалансированная система показателей

BITS — Balanced IT Scorecard, сбалансированная система показателей ИТ (модификация BSC)

TEI — Total Economic Impact, общее экономическое

влияние (Forrester)

TVO — Total Value of Opportunities, совокупная ценность возможностей (Gartner)

SITE — Strategic IT Effectiveness, стратегическая эффективность ИТ (Accenture)

EVS — Economic Value Sourced, извлечённая экономическая стоимость

EVA — Economic Value Added, экономическая добавленная стоимость

KPI — Key Performance Index, ключевой индекс эффективности

CSF — Critical Success Factor, критический фактор успеха

COBIT — Control Objectives for Information and Related Technologies, Задачи управления для информационных и смежных технологий (открытый стандарт)

CSS — Customer Satisfaction Survey, анализ удовлетворённости заказчика

ITUSS — IT User Satisfaction Survey, анализ удовлетворённости пользователя ИТ (модификация CSS)

SLA — Service Level Agreement, соглашение об уровне сервиса (приложение к контракту на

ИТ-обслуживание)

VHCS — Value Health Check Survey, мониторинг («здоровья») ценности взаимоотношений

АоС — Age of The Customer, Эра потребителя (глобальная экономика после 2010 года)

ИС — информационная система

О чём эта глава

Одним из первых системно сформулировал и предложил рынку терминологию и стройную концепцию бизнес-ценности ИТ (BVIT) и её оценки ещё в 2002 году г-н Тони Мёрфи (Tony Murphy), до 2005 года вице-президент компании Gartner Group. Тони Мёрфи — ирландец, сын одного из руководителей компартии Ирландии, бывал в СССР ещё ребёнком, в 2010 году посетил Россию с лекциями по приглашению ассоциации стратегического аутсорсинга «АСТРА». Основные положения концепции BVIT были изложены в его бестселлере того времени «Получение бизнес-ценности от технологий: практическое руководство для современного руководителя» (Achieving Business Value from Technology: A Practical Guide for Today's Executive), впервые опубликованной издательством Gartner Press. В то время главным драйвером изменений во взглядах на ИТ, в целом, а также появления настоящей потребности руководителей предприятий в большем понимании ценности ИТ для бизнеса, по мнению Тони Мёрфи, был переход от традиционной Индустриальной экономики к новой Экономике знаний. Кстати, следующая его книга так и называлась — «Достижение успеха в Экономике

знаний» (Succeeding in the Knowledge Economy). Первопроходцами в разработке методологий оценки инвестиций и ценности ИТ подхода в конце прошлого — начале этого века (1999–2000 гг.) были компании Forrester Research и Giga Information Group, позже вошедшая в состав Forrester Research (методика TEI). Свою методику оценки в 2002 году предложила компания META Group, которая в 2004 году была приобретена Gartner Group (методики TCO и TVO). Компания Accenture также в 2004 году предложила рынку свою методику (услугу) по оценке эффективности ИТ-проектов SITE.

В настоящей главе мы дадим обзор экономических и рыночных тенденций, ведущих к изменению роли ИТ и CIO на предприятии, представим рекомендации по повышению влияния ИТ на развитие основной деятельности (бизнеса) предприятия через синхронизацию («выравнивание») бизнеса и ИТ (IT & Business Alignment), обозначим основные финансовые и нефинансовые подходы, а также представим методологический инструментарий, позволяющий измерить это влияние и сравнить ценность ИТ-проектов для бизнеса.

Эволюция роли и функций корпоративных ИТ

В конце (в 90-е годы) прошлого столетия решения в области ИТ, в основном, относились к компетенции специально выделенной на предприятии службы ИТ. Такое выделение

было связано с тем, что владельцы и руководители предприятий не очень понимали, что такое ИТ, и не ощущали существенного влияния ИТ на развитие бизнеса. Роль службы ИТ

заклучалась в организации закупок, в обеспечении функционирования оборудования и ПО от традиционных поставщиков ИТ (MS, Oracle, IBM, HP и др.). Диалог бизнеса и ИТ строился вокруг расходов на ИТ и повышение эффективности ИТ было связано в основном с заботой о снижении этих расходов. Основным инструментом оценки эффективности ИТ на этом этапе – вычисление общей стоимости владения ИТ TCO. Для оценки эффективности ИТ данный показатель можно сравнивать с показателями TCO подобных предприятий (benchmarking), а также снижать показатель TCO для предприятия с течением времени.

С развитием рынка информационных технологий и решений, построенных на основе ИТ, бизнес все больше начинает использовать ИТ для повышения своей внутренней эффективности путём внедрения решений автоматизации документооборота и бухгалтерии, ERP, систем управления базами данных и т.п. На этом этапе к оценке расходов на ИТ добавляется определение и количественная оценка выгод для бизнес-подразделений от внедрения/использования ИТ. Все чаще решения в сфере ИТ начинают приниматься совместно бизнесом (бизнес-подразделениями) и ИТ. Как следствие, растёт популярность более сложного класса инструментов финансовой оценки эффективности ИТ, в частности методологии оценки возврата на инвестиции ROI. После Интернет-бума начала 2000-х, связанного с появлением новых вендоров и новых технологий, непосредственно влияющих на развитие бизнеса и конкурентоспособность предприятий (электронная торговля, CRM, SCM, мобильность и др.), решения в области ИТ становятся ещё в большей

степени результатом тесного сотрудничества бизнеса и ИТ предприятий. Начинает исчезать само понятие «ИТ-проекта», все больше речь идёт о бизнес-проектах с большей или меньшей составляющей ИТ.

Динамичное развитие и взаимопроникновение бизнеса и технологий влекло за собой значительные качественные изменения роли и функций ИТ и, соответственно, служб ИТ, на предприятии. По мере роста уровня зрелости организации и управления ИТ от ИТ-службы руководство предприятия стало ожидать не только и не столько внедрения самых передовых (и дорогостоящих решений), сколько привнесения непосредственной пользы или ценности для основной деятельности. Таким образом, на первый план повестки дня ИТ-директора начинают выходить следующие задачи:

- Ведение диалога с бизнесом о создании реальной ценности для бизнеса, понимание сути и разрешение стоящих перед бизнесом задач, расстановка бизнес-приоритетов в своей деятельности,
- Аргументированный выбор проектов для инвестиций в ИТ, в зависимости от их текущей и стратегической бизнес-ценности,

Рис. 1.3.1. . Пирамида процессов управления ИТ.



- Количественное определение и реализация бизнес-ценности ИТ-проектов на практике

Упрощённо можно выделить следующие четыре уровня организационной зрелости службы ИТ на предприятии (Рис. 1.3.1), включая управление элементами, операциями, сервисами и собственно ИТ.

1. Управление элементами. Начальный уровень зрелости, который характеризуется восприятием ИТ как вспомогательной службы по «настройке компьютеров». Основным функционалом ИТ на данном уровне является установка, конфигурирование и поддержка отдельных компонентов корпоративной сети, «реактивное» устранение возникающих неисправностей оборудования и ПО по факту их возникновения и др.

2. Управление операциями. На данном уровне восприятие ИТ на уровне предприятия пока остаётся на уровне второстепенной технической службы, однако происходят качественные изменения в организации и функционировании самой службы ИТ. К функционалу начального уровня добавляются задачи комплексного администрирования корпоративной сети, включая управление ошибками, «проактивное» преодоление неисправностей и т.д.

3. Управление сервисами. На уровне управления сервисами служба ИТ как правило воспринимается как самостоятельное и равноправное подразделение (наряду с производством, финансами и др.), оказывающее другим подразделениям предприятия необходимые ИТ-услуги. К функционалу добавляются анализ «истории» ошибок, определение базовых характеристик работы ИТ, договора и отчёты об уровне обслуживания пользователей, мониторинг опыта пользователей и др.

4. Управление ИТ как бизнесом. На данном этапе служба ИТ становится одним из стратегических бизнес-подразделений предприя-

тия и воспринимается как средство развития основной его деятельности (бизнеса). К «внутреннему» функционалу добавляется взгляд на ИТ как на бизнес-услугу (через различные модели инсорсинга и аутсорсинга) для внутренних пользователей и клиентов предприятия, проводится анализ экономического эффекта ИТ, моделирование, финансовый анализ и планирование и т.п.

Таким образом, рынку потребовались более сложные комплексные методики расчёта экономической эффективности ИТ, на что откликнулись основные глобальные поставщики услуг исследований и аналитики — Forrester Research и Gartner Group — появлением специализированных методик Forrester TEI и Gartner TVO. В это же время получила широкое распространение концепция выравнивания (синхронизации) бизнеса и ИТ (IT&Business Alignment).

При этом уже в начале 2000-х годов с активным развитием в мире рынка услуг и сорсинга ИТ, включая различные модели инсорсинга и аутсорсинга, к руководителям бизнеса и ИТ начало приходить понимание того, что кроме объективных финансовых результатов ИТ и/или выполнения условий контрактов и SLA на ИТ-обслуживание (внутренних или внешних) при оценке эффективности ИТ необходимо учитывать «мягкие» субъективные факторы, например, удовлетворённости пользователя (клиента), и использовать опросные социологические методы для сбора соответствующей информации. Одна из наиболее популярных в то время моделей оценки удовлетворённости клиентов называлась CSS — Customer Satisfaction Survey. Модификация этой методики для нужд ИТ-рынка получила название ITUSS — IT User Satisfaction Survey. Смысл методики заключается в том, что поставщику и заказчику ИТ-услуг задаются похожие («параллельные») вопросы на предмет важности и удовлетворённости теми или иными параметрами ИТ-обслуживания. После сбора данных

делается многомерный GAP-анализ (5 измерений), на основании которого делались выводы об уровне удовлетворённости и рекомендации по её улучшению. Эти исследования, как правило, делались на основе персональных интервью с представителями поставщика и заказчика услуг и стоили достаточно дорого. С ростом зрелости рынка ИТ-аутсорсинга появилась необходимость расширить арсенал и глубину качественных методов исследования и, кроме того, существенно удешевить и автоматизировать сбор и анализ данных, а также подготовку отчёта по исследованию. Так появилась новая методика и соответствующий «облачный» инструмент VHCS — Value Health

Check Survey. Методику VHCS разработал международный консультант в сфере Sourcing Governance Мэтт Шокли (Matthew Shocklee), руководитель компании GSOS (Global Sourcing Optimization Services), «облачное» решение построено на платформе компании Enlighta. В данной методике традиционный подход CSS расширен ещё на 4 важные области оценки взаимоотношений, включая оценку финансов, инновационности, технической квалификации и соответствия законодательству/ограничения (Compliance). Опросы представителей поставщика и заказчика осуществляются через Интернет, что существенно удешевляет и ускоряет процесс исследования.

Что такое бизнес-ценность ИТ?

Попробуем проиллюстрировать идею бизнес-ценности ИТ на предприятии рисунком, на котором выделены три основные категории субъектов ИТ-рынка сверху — вниз (Рис. 1.3.2):

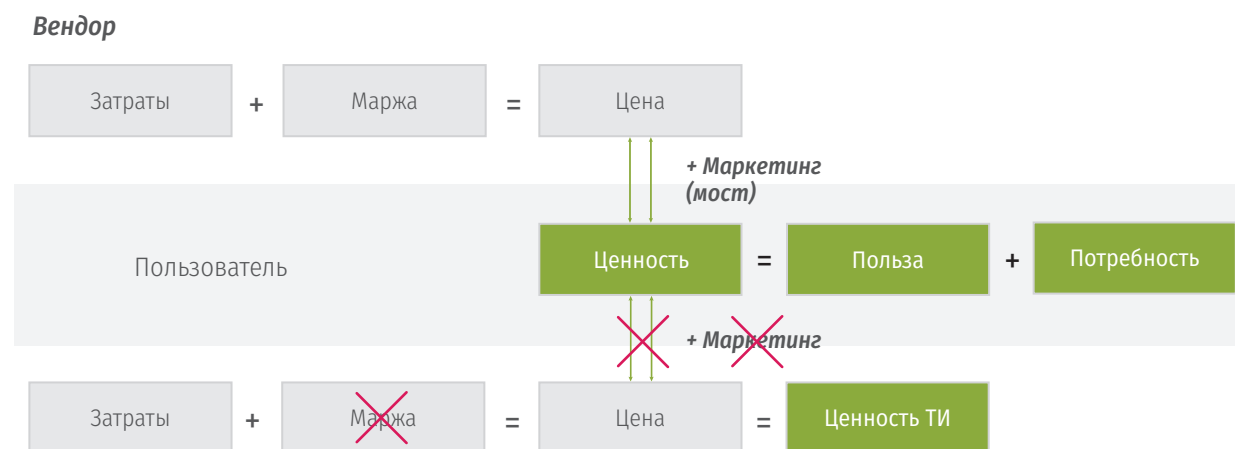
- Производитель оборудования и ПО (вендор);
- Бизнес-пользователи;
- ИТ-служба предприятия.

Как утверждается в бизнес-литературе, ценностью для предприятия является все то, что (а) отвечает потребностям его сотрудников

и клиентов, а также (б) приносит им определённую пользу (полезность).

Производителя в первую очередь интересует цена на продукцию, которая складывается из затрат и прибыли. Чем выше ценность, которую получает предприятие от использования продукции, тем большую цену оно готово заплатить за эту продукцию. Кроме того, производитель имеет возможность увеличивать цену на свою продукцию, убеждая предприятие в её ценности посредством **активного**

Рис. 1.3.2. Категории субъектов ИТ-рынка.



маркетинга (сегментация рынка, фокусная адресация потребностей и т.п.).

ИТ-служба также несёт затраты, связанные с предоставлением услуг. Кроме того, ему также необходимо получать определённую прибыль (или строку в бюджете на ИТ) для дальнейшего развития. Если поставщик услуг ИТ не занимается внутренним маркетингом, т.е. не работает над повышением ценности своих

услуг, ему сложно убедить руководство бизнеса в том, что он является бизнес-партнёром и, что кроме покрытия текущих расходов, ему также нужно постоянно развиваться, чтобы успевать за динамичным развитием технологий и самого бизнеса. Ценность ИТ в таком контексте прямо зависит от понимания и реализации ИТ-службой бизнес-ценности для бизнес-пользователей предприятия.

Важность построения бизнес-кейса

Любой серьёзный ИТ-проект или инициатива должны сопровождаться созданием бизнес-кейса — документа, который описывает причины и необходимость инвестирования, ожидаемые выгоды от инициативы, расходы на реализацию проекта, анализ рисков и будущие опции, порождаемые проектом. Бизнес-кейс документирует релевантные факты и анализ ситуации, основные метрики, финансовый анализ, план-график выполнения проекта, также демонстрирует бизнес-императивы по инициации и финансированию проекта.

Бизнес-кейс является эффективным инструментом коммуникации бизнеса и ИТ, который позволяет уводить содержательное обсуждение от дебатов по узким техническим проблемам или эмоциональной дискуссии о преимуществах и недостатках того или иного конкретного вендора или платформы. Разработка системной и чёткой структуры бизнес-кейса позволит сравнивать и сопоставлять различные проекты с различными целями, а также транслировать всем участникам процесса единые цели и бизнес-ценность проектов, выбранных для финансирования.

Другими словами, как говорили восточные мудрецы: «Если вы не знаете, куда идёте, как вы узнаете, что пришли?». Хорошо подготовленный и понятный бизнес-кейс приведёт к следующим позитивным следствиям:

- Лучшая привязка проектов к целям организации,
- Возможность оценки со стороны всех заинтересованных сторон,
- Появление инструментов мониторинга и сопровождения проектов,
- Появление инструментария измерений и извлечения уроков из проектов.

Однако, необходимо отметить, что бизнес-кейс не является статическим документом, он должен дорабатываться по мере прохождения проекта и реализации его различных сценариев с учётом рисков проекта. Бизнес-кейс определяет основные вы-

«Мы делаем _____,
для того, чтобы сделать
_____ лучше, что
измеряется как _____,
и будет соответствовать €_____»

годы проекта, предоставляя руководству и всем участникам процесса принятия ИТ-решений единую, сжатую и одновременно содержательную информацию о том, что:

Таким образом, при измерении бизнес-ценности ИТ на предприятии на передний план выходят два аспекта проблемы, а именно:

- Бизнес-перспектива — задача синхронизации бизнеса и ИТ.
- ИТ-перспектива — задача измерения и обоснования ценности ИТ.

Очевидно, что любая ценность является относительной и бизнес-ценность ИТ не является исключением. В зависимости от уровня зрелости ИТ на предприятии, а именно в зависимости от степени влияния ИТ на внутренние операции и на стратегию предприятия, в целом, меняется и ценность ИТ. Меняется также и набор метрик, который позволяет замерять и оценивать эту эффективность. Эти метрики условно можно разделить на три категории:

1. Метрики внутренней эффективности ИТ в терминах расходов на ИТ, производительности ИТ,

отказоустойчивости оборудования и ПО и др.

2. Метрики влияния ИТ на эффективность работы внутреннего персонала через удовлетворённость и производительность пользователей ИТ, качество и готовность обслуживания, планирование ИТ и др.

3. Метрики внешней эффективности ИТ через удовлетворённость и обслуживание клиентов предприятия, гибкость бизнеса, время выхода на рынок, доходы и прибыль.

Таким образом, от узко технических ИТ-терминологии и ИТ-метрик, таких как масштабируемость, доступность, широта каналов, производительность оборудования и т.п., происходит качественный переход к финансовым метрикам и оценкам, в том числе ROI, IRR и др.

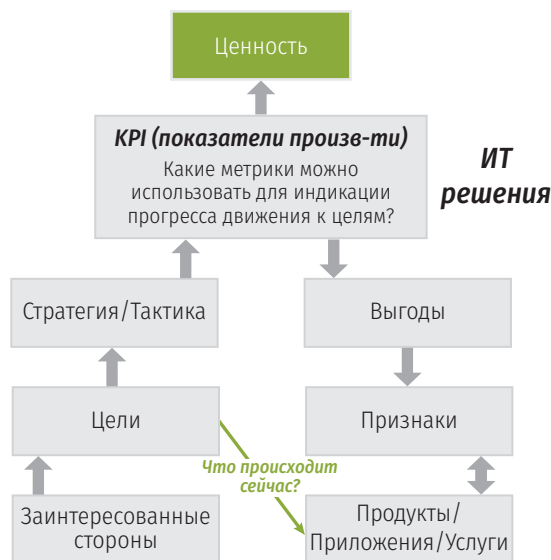
Синхронизация целей бизнеса и ИТ

На Рис. 1.3.3 представлена модифицированная схема «обычного» стратегического бизнес-планирования, используя которую возможно провести синхронизацию (или выравнивание — alignment) целей бизнеса и ИТ, а также определить набор метрик, который позволит увязать инвестиции в ИТ с ключевыми показателями эффективности бизнеса (KPI).

Выработка общих метрик, которые прямо относятся к бизнес-целям, а также нахождение согласия руководителя бизнеса и руководителя ИТ являются наилучшими индикаторами ориентированного на бизнес мышления и синхронизации ИТ. Такое согласие обеспечивает совместную работу всех заинтересованных сторон процесса на одинаковые цели, которые непосредственно относятся к стратегии предприятия. В том случае, если участникам проекта разрешено разрабатывать независимые метрики («план-факт» для инженеров, «количество полученных наград» для маркетинга, «минимальные рас-

ходы» для бухгалтерии и т.п.), базовые бизнес-причины проекта будут достаточно быстро забыты.

Рис. 1.3.3. Схема стратегического бизнес-планирования.



Только через процесс совместной работы всем участникам становится ясным, что соответствующее ИТ решение поддерживает одну или несколько бизнес-целей, стратегий или тактик предприятия. Следовательно, важно, чтобы все участники процесса планиро-

вания, включая ИТ-службы и бизнес-подразделения предприятия, имели одинаковые метрики успеха. Без чёткого направления, которое и обеспечивают общие метрики или набор метрик, проекты могут легко выйти за рамки бюджета или потерпеть неудачу.

Инструменты измерения эффективности ИТ

Одним из вопросов определения эффективности ценности и инвестиций в ИТ является выбор методики оценки. В классической литературе, посвящённой вопросам оценки эффективности, она рассчитывается в целом по весьма простой формуле:

$$\text{Эффективность} = \frac{\text{Эффект}}{\text{Затраты}}$$

Затраты — это совокупные затраты на приобретение, установку и конфигурирование, сопровождение и поддержку, а также затраты связанные с простоем оборудования во время технического обслуживания или устранения неисправностей.

Эффект — это совокупные эффекты, достигаемые при внедрении/использовании ИТ. Однако, как правило, определить прямой эффект от их внедрения (во временных или финансовых показателях) требует определённых навыков и даже искусства.

На сегодняшний день множество известных методов оценки можно разделить на следующие четыре основные категории:

1. Затратные методы — оценка производится на основе затраченных ресурсов, а не на основе измерения конечного продукта или результата.

2. Методы оценки прямого результата — оценка прямого измеримого результата, например, снижение стоимости владения,

повышение функциональности системы, снижение трудозатрат или появление побочного продукта/эффекта.

3. Методы, основанные на оценке идеальности процесса — базируются на статических или динамических сравнительных алгоритмах. Базовым показателем выбирается объект рассматриваемой системы, идеальной считается ИС с альтернативными или лучшими, например, для отрасли показателями затрат на единицу выхода.

4. Квалиметрические подходы — комплексно рассматривают объект ИТ (ИС и др.), организуют её измерение и обрабатывают полученные результаты статистическими, социологическими и/или экспертными методами.

Ниже приведём некоторые наиболее известные/популярные методики оценки из каждой категории с краткой характеристикой:

Затратные методы оценки

Котловой метод

Метод основан на определении соотношения объёмов вложений в ИТ, включая внедрение и сопровождение, с размерами предприятия и направлениями его бизнеса. Часто данное соотношение задаётся в виде максимально-допустимого объёма вложений в ИТ по отношению к годовому обороту компании, например, не более 1% для небольших компаний и не более 3% для крупных.

Метод функциональной точки

Данный метод используется для приблизительной оценки стоимости создания и вне-

дрения ИС в зависимости от требований пользователя. Каждое требование оценивается по шкале трудности (лёгкие, средние и трудные) и по шкале важности для пользователя. Требования представляются в виде вектора (функциональной точки) в многомерном пространстве. Далее в соответствии с гипотезой «компактности» предполагается, что чем ближе функциональные точки проектов друг к другу в пространстве требований, тем их параметры, включая и эффективность, более схожи. Соответственно, в базе ранее внедрённых проектов находится такой, чья функциональная точка ближе всего находится к проектируемой ИС, и предполагается, что их эффективности максимально близки.

Метод ТСО

Данный метод предполагает количественную оценку на внедрение и сопровождение ИТ, часто рассчитываемую по формуле:

$$Z_{r\text{ инт}} = Z_{r\text{ факт}} + \sum_{t=r}^T (1 + E)^{-t} \times Z_{t\text{ оц}}$$

где $Z_{r\text{ инт}}$ — оценка интегрированных затрат по проекту в момент r ; E — норма дисконтирования, отражающая временной характер финансовых ресурсов; — дисконтированная сумма фактически произведённых интегральных затрат на момент t ; T — период жизненного цикла системы; $Z_{t\text{ оц}}$ — оценка интегральных затрат на проект в периоде t .

Модель ТСО позволяет разобраться в структуре расходов, связанных с ИС, и открывает широкие перспективы для их сокращения, также способствует выявлению текущих проблем, обеспечивает постоянную обратную связь в управлении затратами.

Методы оценки прямого результата

Потребительский индекс (Customer index)

Этот метод предполагает оценку результатов внедрения ИТ в виде совокупности индексов, отражающих положительные изменения в работе компании (увеличение доходов, снижение затрат, увеличение оборотов, увеличение клиентской базы и т.п.).

Метод EVS

Представляет собой оценку того, какую пользу ИТ приносит компании при его использовании, оценивается по четырём показателям: увеличение доходов, повышение производительности труда, сокращение времени выпуска продуктов, снижение рисков.

Метод EVA

Данная методика предполагает определение эффекта как фактическую прибыль от использования ИТ, которая равна чистой операционной прибыли за минусом стоимости капитала. Применительно к ИТ проектам EVA означает, что:

- при использовании капитала в ИТ проектах, необходимо учитывать его стоимость, за него необходимо платить также, как и за труд работников;
- предполагается, что ИТ-специалисты продают свои услуги другим подразделениям по рыночным расценкам.

Это позволяет рассматривать ИТ как центр прибыли, а не затрат, при этом чётко отбрасывая, как увеличиваются доходы.

Методики, основанные на идеальности процесса

Сравнительный анализ результатов (Benchmarking)

Оценка результативности внедрения ИТ проводится по сравнению со средними отраслевыми и иными известными результатами.

Результаты такого сравнительного анализа обычно приводятся в открытых публикациях и маркетинговых материалах.

Метод ROI

Суть методики заключается в выборе для компании типового проекта, оптимального по показателю сроков возврата инвестиций в ИТ.

Квалиметрические методы

Gartner TVO

Универсальная и поэтому весьма громоздкая модель — 54 метрики, объединённые в 9 групп. Модель предполагает выполнение 7 этапов (шагов), включая:

1. Определение релятивных метрик и их величин
2. Выяснение возможностей инициативы и их влияния на метрики
3. Расчёт суммарной выгоды по годам
4. Определение TCO
5. Определение класса инициативы
6. Определение опций для будущего развития
7. Выяснение способности предприятия реализовать выгоды (риски проектов)

BSC

Система стратегического управления организацией на основе измерения и оценки её эффективности через использование комплексной функции, включающей набор показателей, учитывающих все аспекты деятельности компании (финансовые, маркетинговые и т.д.). К таким показателям обычно относят:

- критические факторы успеха (CFS) — стратегические показатели: финансы, клиенты, внутренние бизнес-процессы, обучение и рост;
- ключевые показатели эффективности

(KPI), включая достигнутые результаты деятельности компании.

Состав и количество сбалансированных показателей определяются исходя из специфики каждой компании.

Forrester TEI

В качестве затратной компоненты данного метода используется модель TCO, а эффект рассчитывается на основе следующих факторов: Преимущества, Гибкость, Риски. Рассмотрим метод немного подробнее.

Основными предпосылками при создании методологии TEI были следующие:

- необходимость для CIO обоснования перед руководством компании и акционерами запросов на финансирование дорогостоящих ИТ проектов;
- необходимость ведения диалога с представителями бизнеса не в технологических, а в бизнес терминах. Демонстрация ценности проекта для основного бизнеса компании, его сопряжения (синхронизации) со стратегией развития бизнеса компании;
- осознание трудностей в использовании существовавших стандартных методологий (например, ROI или TCO) для определения экономической эффективности предстоящих ИТ-проектов;
- осознание рисков принятия неправильных решений в отношении ИТ проектов при использовании только затратных методов оценки проектов (например, TCO);
- необходимость учёта как материальных, так и нематериальных выгод, получаемых в результате ИТ-проекта;
- связь ИТ-проекта с другими проектами, в том числе и возможными будущими проектами;
- введение поправок на риски реализации проекта.

Традиционные методики, например, TCO, вошли в указанную модель как составные части для оценки затрат и для сравнения с лучшими практиками в целях минимизации затрат. Кроме того, модель TEI включает как подробный анализ и количественную оценку как выгод, получаемых бизнесом компании вследствие данного ИТ-проекта, так и оценку гибкости ИТ-решения (связь его с другими проектами и «отложенные» выгоды) и поправку на риски реализации проекта.

Используя TEI организации могут оценить свои решения по отдельным ИТ-проектам и/или ИТ-продуктам в свете индивидуальных бизнес-целей организации и тем самым представить ИТ-службу не как «центр расходов», а как «центр прибыли» предприятия. Тем самым технологические ИТ решения будут лучше сопряжены с бизнес-целями компании, рейтинг успешности проекта возрастет, оценка рисков становится более конкретной и понятной, а сами риски могут быть уменьшены. Все это предполагает в конечном итоге достижение одной цели — повышение эффективности ИТ и бизнеса компании.

Модель TEI в основе содержит четыре базовых элемента, имеющими каждый свои методологию и инструментарий. (Рис. 1.3.4). Эти

четыре элемента предполагают рассмотрение и анализ следующего:

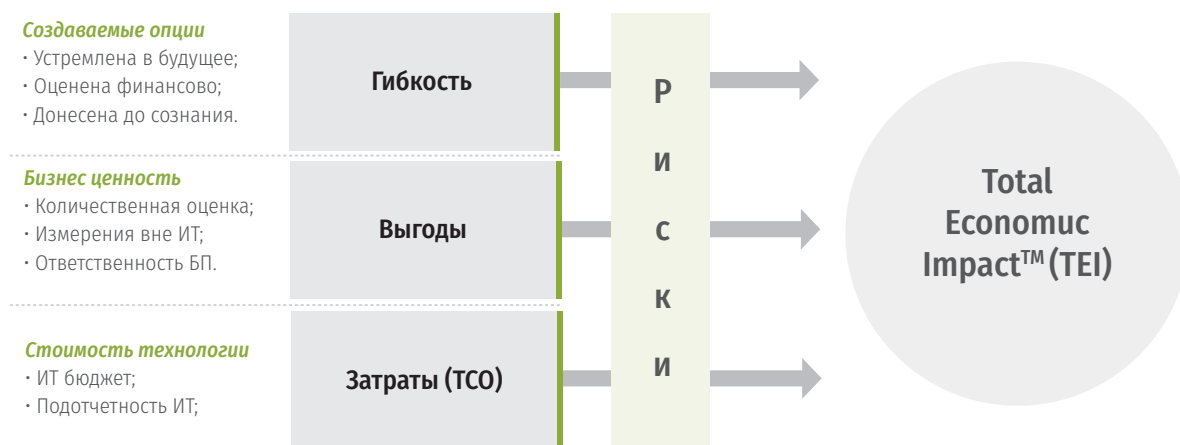
- Затрат (на основе модели TCO)
- Гибкости ИТ решения
- Получаемых выгод
- Оценку показателей первых трёх позиций с учётом рисков проекта.

Затраты. Любая модель оценки ИТ-проекта должна содержать достаточно подробный и всесторонний анализ расходов. Этот анализ предполагает оценку как прямых, так и косвенных ИТ-расходов. Примерами таких расходов могут быть:

- Капитальные затраты на приобретение оборудования и лицензий ПО, а также их апгрейдов.
- Текущие расходы на поддержку и операционные расходы, включая расходы на внутренний персонал и на внешние ИТ-услуги.
- Административные расходы и т.д.

Прямые выгоды. Категория выгод в TEI, в основном, содержит количественные показатели, связанные с изменениями вне ИТ-департамента, т.е. в бизнес подразделениях, внешних связях, позициях на рынке и т.д. Этот подход представляет собой краеугольное отличие подхода Forrester от ранее существовавших моделей оценки затратной эф-

Рис. 1.3.4. Схема модели TEI.



Источник: Forrester

эффективности ИТ расходов (в частности ТСО). Другой момент, на который следует обратить внимание при разработке метрик для оценки выгод для бизнеса, — оценка нематериальных выгод. Здесь имеется большой простор для креативных способностей консультантов и аналитиков.

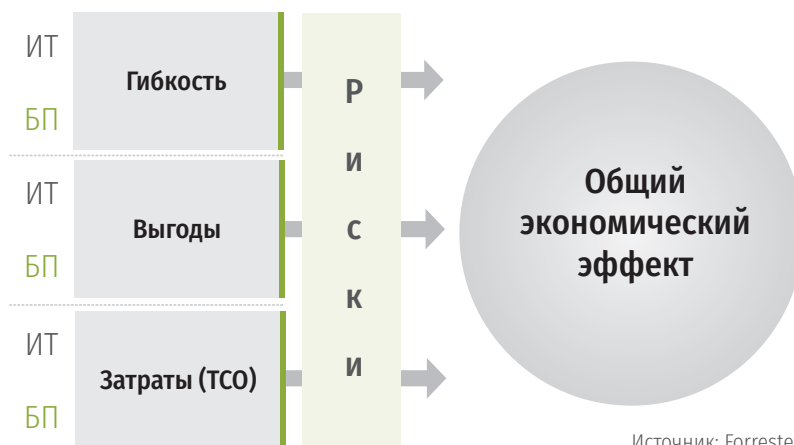
Кроме того, было учтено следующее: при внедрении многих ИТ-систем первоначальное внедрение требует изменений в поведение персонала-пользователей ИТ. Может случиться, что специалисты по маркетингу не будут доступны или их работа будет менее эффективной при внедрении систем управления маркетингом. Персонал отделов продаж будет проходить обучение, вместо того чтобы продавать. Поэтому внедрение многих систем может иметь отрицательный начальный эффект, который, однако, будет компенсирован в дальнейшем. В TEI для учёта этого обстоятельства использовался следующий подход: рассмотрение двух сценариев развития организации — без ИТ-инновации и с учётом ИТ-инновации.

Гибкость (будущие выгоды). Другой ключевой момент TEI: Часто любой ИТ-проект имеет или предполагает продолжение и/или связан с другими предполагаемыми ИТ-проектами. Forrester перенёс подход для оценки покупки ценных бумаг с опционом, разработанный Fisher Black, Robert Merton и Myron Scholes в 1965 году, на ИТ-проекты. Этот подход позволял делать оценку будущих выгод в предполагаемых условиях рынка. Аналогично, дополнительные инвестиции в инфраструктуру сейчас (сверх текущих потребностей) позволяют в дальнейшем внедрить те или иные приложения, и тем самым получить дополнительные выгоды от сегодняшних инвестиций.

Риски. Никакие изменения не происходят без рисков. Учёт рисков приводит к необходимости рассмотрения затрат, выгод и гибкости в диапазоне потенциально возможных исходов. Среди факторов, которые влияют (увеличивают) на неопределённость конечного результата, можно назвать следующие:

- Поставщики — риск того, что поставщик или технология могут исчезнуть с рынка и должны быть заменены другими продуктами или поставщиками.
- Продукты — риск того, что покупаемый продукт в действительности не будет соответствовать предполагаемой функциональности или свойствам.
- Архитектура — риск того, что получаемая архитектура не позволит в дальнейшем произвести инфраструктурные изменения или внедрить какие-либо решения.
- Корпоративная культура — риск непринятия организацией технологической инициативы или невозможность внедрения инициативы в данных условиях.
- Просрочки — просрочки в поставках, сроках проекта и т.д.
- Размер проекта — чем больше и сложнее проект, тем выше риски, связанные с его неисполнением или не исполнением в рамках заданных бюджета и сроков.

Рис. 1.3.4. Базовая модель TEI (гибкость).



Источник: Forrester

Стандартная ROI

$$\frac{\text{Выгоды} - \text{Затраты}}{\text{Затраты}} = \text{ROI} \%$$

В модели TEI риски учитываются через их
Методология TEI для определения ROI:

$$\frac{(\text{Выгоды} + \text{Гибкость})_r - \text{Затраты}_r}{\text{Затраты}_r} = \begin{matrix} \text{ИТ ROI} \% \\ \text{БП ROI} \% \\ \text{Net ROI} \% \end{matrix}$$

влияние на выгоды, затраты и гибкость. Заметим, что при процедуре количественной оценки рисков становятся ясными и пути их уменьшения. Например, путём разделения больших проектов на ряд малых или закладывание в бюджет больших, но не изменяющихся расходов.

Может возникнуть сомнение, что многие метрики не могут быть реально измерены и сравнены. В частности, нематериальные «мягкие» выгоды. Douglas Hubbard, эксперт в области применения экономических подходов в ИТ, писал в журнале CIO Magazine: «Неизмеряемость» есть иллюзия, вызванная тремя основными моментами непонимания сущности измерений:

1. Не ясен объект измерения.
2. Не ясно назначение измерений.
3. Не ясна методика измерений».

Когда такое непонимание устранено, любое измерение возможно. Для выяснения возможности измерений следуйте следующему простому правилу:

- Если что-то лучше, то существует различие
- Если есть различие, то его можно наблюдать
- Если различие наблюдаемо, то его можно оценить количественно
- Если его можно оценить количественно, то, следовательно, оно измеримо

- Если оно измеримо, значить может быть оценено

Итак, основные этапы методологии TEI:

1. Понять бизнес-цели, стратегию и тактику предприятия.
2. Определить сценарии соответствующего решения.
3. Определить ценность и требуемые инвестиции для сценариев «с точки зрения ИТ».
4. Определить ценность и требуемые инвестиции для сценариев «с точки зрения бизнеса», совместно с бизнесом и ИТ.
5. Выявить/определить количественно потенциальные выгоды от возможных последующих инвестиций (гибкость).
6. Учесть фактор неопределённости для всех финансовых предположений (риски).
7. Определить и применить оптимизирующие метрики.
8. Анализ и оптимизация метрик на текущем базисе (когда необходимо).

Сбалансированная система показателей ИТ-службы (BITS)

Эта комплексная методика оценки ценности ИТ и эффективности работы ИТ-службы является развитием известной сбалансированной системы показателей (BSC) и исходит из целей и потребностей топ-менеджмента.

Основная цель сбалансированной системы показателей ИТ-службы — выстраивание соответствия между целями ИТ и целями бизнеса, фокусирование операционных задач ИТ на достижение соответствующих стратегических целей ИТ.

Не будем вдаваться в описание классической сбалансированной системы показателей, отметим только три наиболее важных момента:

1. Развитие компании согласно сбалансированной системе показателей рассматривается как развитие четырёх направлений:
 - Финансы — Как представляют компанию акционеры и инвесторы?

- Клиенты — Какой видят компанию покупатели её продуктов/услуг?
- Бизнес-процессы — Какие бизнес-процессы требуют оптимизации, на каких организации стоит сосредоточиться, от каких отказаться?
- Обучение и рост — Какие возможности существуют для роста и развития компании?

2. По каждому из этих направлений определяются цели и ключевые показатели эффективности, которые определяют степень достижения этих целей.

3. Наличие причинно-следственной связи. Каждый показатель должен отражать причинно-следственную связь между стратегией компании и её конкретной деятельностью. От общего подхода сбалансированную систему показателей ИТ-службы отличает, прежде

всего, набор перспектив (групп показателей). Различные эксперты рекомендуют заметно различающиеся перспективы (группы показателей деятельности ИТ-департамента). Традиционный вариант набора групп показателей показан на Рис. 1.3.6. Однако, важно понимать, что не существует типового набора показателей деятельности, одинаково эффективного для ИТ-департаментов всех компаний. Для каждой конкретной компании набор перспектив может быть своим, что определяется важностью тех или иных показателей для данной компании и для её стратегических целей.

Например, можно и несколько по-другому определить перспективы, в которые могут быть сгруппированы показатели деятельности ИТ-департамента:

Рис. 1.2.6. Традиционный вариант набора перспектив сбалансированной системы показателей ИТ-службы.

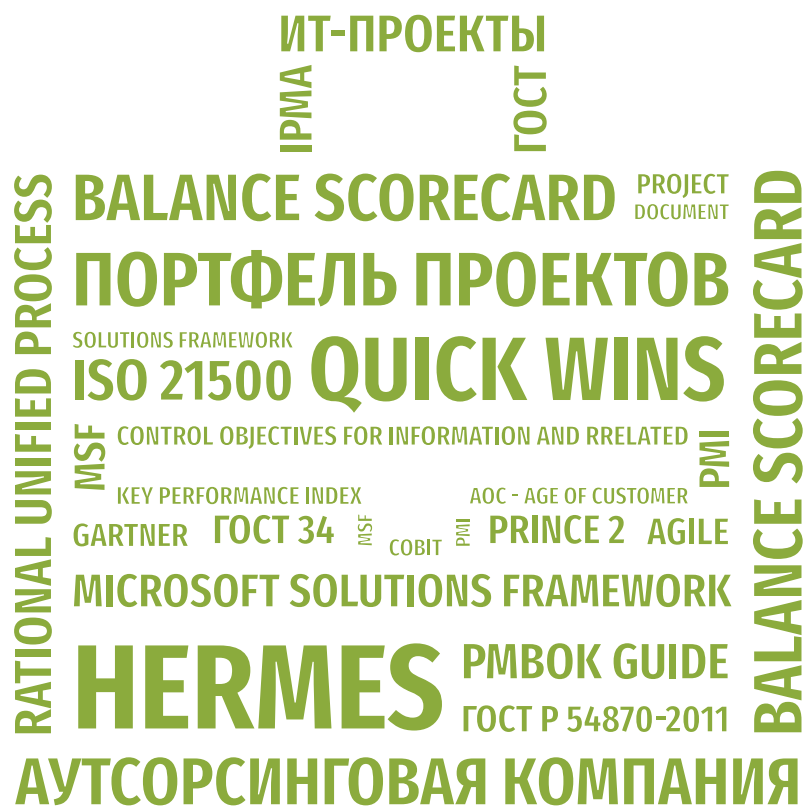


- помощь в развитии бизнеса
- повышение производительности труда
- повышение качества продукции или уровня сервиса потребителей (внутренних и/или внешних)
- повышение качества принятия решений

После выбора перспектив, по каждой перспективе разрабатываются стратегические цели и ключевые показатели эффективности. Пример причинно-следственной связи: обучение ИТ-потребителей (перспектива «Взаимоотношения с конечными пользователями»). Надо заметить, что построение такой связи — далеко не простое дело, в области ИТ многие причинно-следственные связи не очевидны. Согласно логике BSC, оценка показателей

должна проводиться периодически, и её результаты должны сравниваться с предыдущими, оценивая прогресс в выполнении целей ИТ. Однако, при этом методика позволяет определить: что нужно измерять, но не говорит, как именно измерять показатели — эти вопросы остаются на усмотрение руководителей ИТ-департамента и бизнеса.

Применение сбалансированной системы показателей ИТ-службы будет эффективнее, если для оценки эффективности соответствующих областей использовать метрики и показатели, изложенные в стандарте COBIT. Сбалансированная система показателей ИТ-службы в сочетании с моделью COBIT может быть всеобъемлющим механизмом контроля деятельности ИТ.

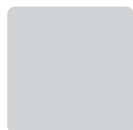


Часть 2

Управление ИТ-деятельностью

Глава 2.1

Управление ИТ-проектами



Павел
Алферов

Определения

Существует огромное количество определений, как самого понятия «проект», так и связанного с ним термина «проектное управление». Фактически, большинство развитых стран имеют свои стандарты по управлению проектами, которые включают в се необходимые определения. В России это ГОСТ Р 54869-2011 «Проектный менеджмент. Требования к управлению проектом». Дальнейшее изложение этой главы в основном опирается на данный стандарт.

ГОСТ Р 54869-2011 даёт следующие определения:

Проект — комплекс взаимосвязанных мероприятий, направленный на создание уникального продукта или услуги в условиях временных и ресурсных ограничений.

Управление проектом (УП) — планирование, организация и контроль трудовых, финансовых и материально-технических ресурсов.

Приведём другие наиболее распространённые определения проекта.

Проект — это временное предприятие (усилие), осуществляемое (предпринятое) для создания уникального продукта или услуги. (PM BOK 2004).

Проект — это уникальный процесс, состоящий из набора взаимоувязанных и контролируемых работ с датами начала и окончания, предпринятый для достижения цели соответствия конкретным требованиям, включая ограничения по времени, затратам и ресурсам. (ISO/TR 10006:1997 (E). Quality Management — Guidelines to quality in project management — p. 1).

Проект — это уникальная совокупность скоординированных действий (работ) с определёнными точками начала и окончания, предпринятая индивидуумом или организацией для достижения определённых целей с установленными сроками, затратами и параметрами выполнения. (British Standard BS 60791:2000. Project management — Part 1: Guide to Project management — p. 2).

Роли проекта. Проектный подход подразумевает в обязательном порядке выделение отдельной организационной структуры для управления проектом. Она может в значительной степени различаться в зависимости от специфики, но в каждом проекте должны быть определены следующие роли (согласно ГОСТ Р 54869-2011):

- **заказчик проекта** — физическое или юридическое лицо, которое является владельцем результата проекта;
- **руководитель проекта** — лицо, осуществляющее управление проектом и ответственное за результаты проекта;
- **куратор проекта** — лицо, ответственное за обеспечение проекта ресурсами и осуществляющее административную, финансовую и иную поддержку проекта;
- **команда проекта** — совокупность лиц, групп и организаций, объединённых во временную организационную структуру для выполнения работ проекта.

Кстати, когда мы говорим о заказчике, необходимо понимать, что это не единая роль. В ней всегда выделяются несколько лиц. Обычно

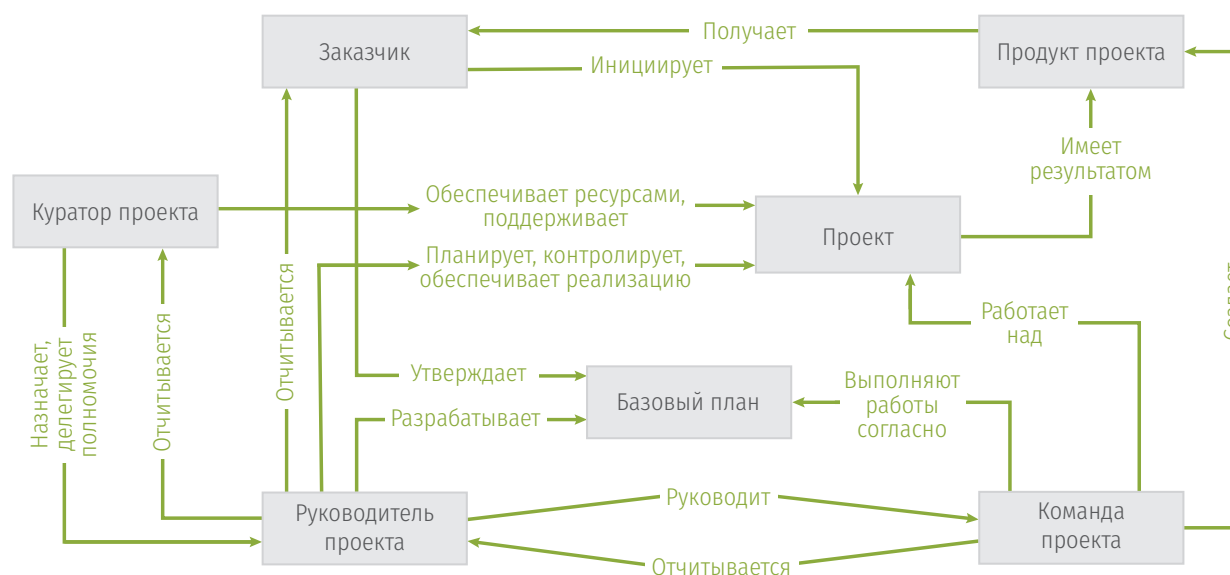
есть топ-менеджер, отвечающий за проект со стратегической точки зрения (есть разные названия: владелец, спонсор и т.д.), и сотрудник бизнес-подразделения, глубоко вовлечённый в проект (ключевой пользователь, лидер проекта, ответственный за проект и т.д.), а также конечные пользователи системы.

Заинтересованные стороны в проекте (stakeholders) — это лица или организации, чьи интересы могут быть затронуты в ходе реализации проекта. Они часто непосредственно не являются участниками проекта, но при планировании проекта необходимо оценивать их реакцию на выполняемые работы и управлять ею.

Жизненный цикл проекта — это совокупность этапов и фаз проекта. Для обеспечения более качественного управления проект разделяют этапы и фазы. Во многих организациях существует чёткая стандартизация этапов, на которые должен делиться проект.

Базовый план проекта — это принятый к исполнению план проекта, содержащий сведения об основных временных и стоимостных параметрах проекта. Базовый план является основой для сравнения фактических по-

Рис. 2.1.1. Взаимосвязи основных сущностей и субъектов проектного управления.



казателей проекта с запланированными, для оценки хода выполнения проекта. Употребляется с уточнениями (базовый календарный план проекта, базовый бюджет проекта).

Изменение в проекте — это модификация утверждённого ранее содержания, сроков, ресурсов, а также установленных процедур.

Что даёт управление проектами?

Сначала несколько слов о том, зачем нужно управление проектами. Знаменитый опрос CHAOS Chronicles, проведённый The Standish Group (Рис. 2.1.2), показал, что в 2009 году в мире только 31% ИТ-проектов завершились успешно, а 23% полностью провалились. В 2013 году картина была уже несколько лучше, чем в 1994, но динамика явно недостаточна. Общей статистики по российским проектам, к сожалению, нет. Существует единственное исследование Hewlett-Packard и Economist Intelligence Unit, согласно которому, только 5% российских ИТ-проектов завершаются в срок. Те, кто утверждает, что провалы — это специфика именно ИТ проектов, не правы: та же

Так как проект, согласно определению, есть создание чего-то уникального, то изменения являются его неотъемлемым спутником. Грамотная организация управления изменениями — одна из наиболее критичных частей управления проектом.

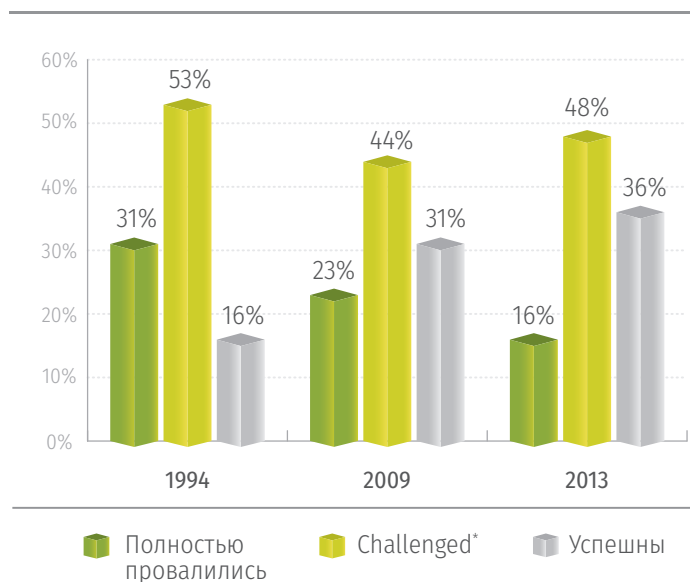
проблема существует, например, и для очень крупных инфраструктурных проектов. Согласно проведённому исследованию, 258 инфраструктурных проектов с общим бюджетом более 90 млрд долл., 9 из 10 проектов сталкиваются с превышением бюджета. И в том, и в другом случае главной проблемой является сложность проектов. Для инфраструктурных проектов это статическая сложность: большие объёмы работ, большое количество техники, материалов, подрядчиков и т.д. Для ИТ-проектов проблема в динамической сложности — элементы системы (серверы, модули, интерфейсы) связаны нелинейными связями — изменение одного вызывает слабо предсказуемую цепочку изменений в остальных.

Помимо вышеприведённых проблем есть три серьёзные причины, которые подталкивают компании к внедрению проектного управления и управления программой проектов:

1. Изменения в организации становятся все более сложными и комплексными.
2. Достижение целей проекта требует тесного взаимодействия и вовлечения множества внешних сторон.
3. Существующая организация, процессы и системы не поддерживают такой вид деятельности, как проект.

Управление проектом с использованием наработанных стандартных инструментов позволяет ощутимо повы-

Рис. 2.1.2. Астатистика реализации ИТ-проектов в мире (The Standish Group).



ситель вероятность его успешного завершения, правда, в обмен на дополнительные затраты (зарплата проектного менеджера, стоимость создания планов, документации, отчётности и т.д.). Дополнительным бонусом идёт сокращение сроков и затрат проекта за счёт избегания непроизводительной, ненужной работы.

И эффект от использования методологий и инструментов управления проектами весьма серьёзный. Исследование The Value of Project Management in IT Organizations, проведённое Center for Business Practices показало, что внедрение методов управления проектами улучшило 20 исследуемых показателей эффективности управления проектами в компании в среднем на 21%. Самые значительные положительные сдвиги были достигнуты в оценках сроков реализации проектов, соответствии проектов стратегическим планам компании, минимизации расходов, повышении продуктивности и качества реализации проектов. 97% менеджеров среднего звена ИТ-компаний, участвующих в управлении или реализации проектов, заявили, что введение методов управления проектами значительно повышает эффективность работы компании. Средний показатель возврата инвестиций на обучение и внедрение системы управления проектами на предприятии оценивается около 28%.

Ещё в 2000 году исследователи Kent Crawford и James Pennypacker провели опрос более

100 руководителей высшего звена, курирующих управление проектами. Исследование показало, что внедрившие управление проектами организации могут ожидать:

- увеличения успешно исполняемых проектов (достижение целей проекта) — в среднем на 50%;
- повышения оборачиваемости капитала — в среднем на 54%;
- повышения удовлетворённости клиентов — в среднем на 36%;
- повышения удовлетворённости персонала — в среднем на 30%.

Наконец, исследование российской ассоциации управления проектами COBHET показало, что профессиональное управление проектами позволяет сэкономить до 30% времени и до 20% средств.

При этом необходимо понимать, что управление проектом — затратная деятельность. Согласно мировой статистике на управление проектом уходит от 2 до 15% бюджета самого проекта. Управление проектом имеет смысл и окупается только в том случае, если перед проектом стоят действительно серьёзные ограничения: по срокам, бюджету, качеству и т.д. Если же перед организацией и проектом серьёзных вызовов — конкурентных, нормативных, экономических и т.д. — нет, то управление проектами внедрять не имеет смысла, оно не будет работать.

Проекты и процессы

Деятельность любой организации — благотворительной, коммерческой, государственной — делится на две большие группы: плановая и внеплановая (Рис. 2.1.3).

Задачей менеджмента организации является максимизация плановой деятельности и минимизация внеплановой. Это очень важный момент — управление по поручениям неплохо работает для небольших компаний в усло-

виях быстро меняющегося окружения, но попытка управлять подобным образом крупной структурой приводит к серьёзным отрицательным последствиям. Эффективно управлять можно только плановой деятельностью. Плановая деятельность, в свою очередь, подразделяется на процессы и проекты. У проектов и процессов есть общие признаки: они выполняются людьми, ограничены доступ-

ностью ресурсов, планируются, исполняются и управляются. Но есть и существенные отличия.

Повторяющаяся деятельность (процессы, операционная деятельность):

- периодически повторяется;
- понятна, имеет высокую степень определённости.

Создание нового (проектная деятельность):

- уникальна, имеет ряд инновационных аспектов, нет опыта выполнения таких работ;
- имеет чётко заданные ограничения: дату завершения, ограниченный бюджет и т.д.

Из различий в этих признаках возникают совершенно разные подходы к управлению.

Управление проектами — отдельная область менеджмента, предназначенная специально для управления временной деятельностью с уникальными результатами, в которой применяются специальные организационные инструменты. Традиционное операционное управление бизнесом, ориентированное на управление устоявшимися бизнес-процессами, не справляется с быстрыми и существенными изменениями.

Граница между видами деятельности в реальной жизни часто условна — в зависимости от различных обстоятельств одну и ту же деятельность можно рассматривать и как проект, и как процесс. Например, доработка существующей и давно внедрённой на предприятии информационной системы, в зависимости от «масштаба бедствия», может рассматриваться и как процесс поддержки системы (создание нового отчёта), и как полноценный проект (изменение настроек системы в связи с изменени-

ями правил регулирования рынка). Ещё один пример — проекты тиражирования существующих систем в региональные офисы. В зависимости от условий — трудоёмкости, необходимости дополнительных настроек — это может рассматриваться и как проект, и как процесс. Учитывая все эти особенности, организация должна ввести свои правила разделения проектной и процессной деятельности.

Основные вопросы, на которые при этом необходимо ответить: адекватно ли для того или иного вида деятельности использование проектных инструментов, позволит ли оно повысить эффективность работы при реализации задачи. При этом необходимо понимать, что если какая-то деятельность названа проектом, в ней обязательно должны быть реализованы некоторые требования (см. ГОСТ Р 54869-2011). В качестве примера классификации деятельности организации во врезке дан алгоритм выделения проектов, процессов и других активностей в АНО «Оргкомитет «Сочи 2014».

Заметим, что чёткое разделение деятельности на проекты и процессы не обязательно может применяться сразу ко всей организации. Этот же подход может быть использован и на уровне отдельного подразделения, например, ИТ-службы. В рамках ИТ-службы можно достаточно легко выделить стандартные про-

Рис. 2.1.3. ССхема структурирования деятельности организации.



цессы и проекты по развитию ИТ. Подробно инструменты управления процессами будут рассмотрены в главе 2.3 «Управление ИТ-процессами и услугами».

В современной теории менеджмента также идёт проработка отдельной плановой сущности, не относящейся ни к проектам, ни к

процессам – кейсов. Кейс выделяется в том случае, если существует высокая неопределённость, изменчивость окружающей среды. Эта область пока ещё недостаточно хорошо изучена и описана, хотя уже существует целый ряд систем управления кейсами – Adaptive Case Management.

Алгоритм разделения программ, проектов, мероприятий и процессов в АНО «Оргкомитет «Сочи 2014»

Оргкомитет «Сочи 2014» — это автономная некоммерческая организация, созданная для подготовки и проведения XXII Олимпийских зимних игр и XI Параолимпийских зимних игр 2014 года в городе Сочи. Вся плановая деятельность в Оргкомитете «Сочи 2014» делилась на 4 типа:

программа — набор взаимосвязанных и взаимовлияющих проектов, процессов и мероприятий, отдельных задач, предназначенных для достижения целей Оргкомитета;

проект — организационная форма выполнения взаимосвязанных работ, направленных на достижение уникальных результатов в условиях ограниченного времени и ресурсов. Проект выделяется в целях повышения управляемости данных работ, минимизации рисков и применения их координации рекомендаций и передовых мировых практик проектного управления;

мероприятие — кратковременный неформализованный набор работ, направленных на получение заданных результатов. Мероприятие можно рассматривать как вид проектной деятельности, к которому применяется упрощённый документооборот в связи с его краткосрочностью и низкой трудоёмкостью;

процесс — связанный и документированный набор работ по получению повторяющихся результатов.

Алгоритм разделения программ, проектов, мероприятий и процессов следующий:

1. Активность классифицируется как Программа, если срок получения результатов больше 24 месяцев. В случае, если срок получения результатов от 12 до 24 месяцев активность может квалифицироваться и как программа и как проект. Для программы должна быть проведена декомпозиция на проекты, процессы и мероприятия.
2. Активность классифицируется как Процесс, если в течение года необходимо получить похожий результат ещё не менее двух раз.
3. Активность классифицируется как Проект, если длительность получения результата больше 3х месяцев или трудоёмкость больше 3х чел/мес. или бюджет более 1 миллиона рублей.
4. Активность классифицируется как Мероприятие, если длительность получения результата меньше 3х месяцев или трудоёмкость меньше 3х чел/мес. или бюджет меньше 1 млн руб. включающий ИТ-компоненту с бюджетом более 10 тыс. долл.

Проект: работы и результаты

Несколько упрощая, проект можно рассматривать как сочетание установленных целей (ожидаемых результатов) и работ по их достижению. Результаты должны быть получены с учётом имеющихся ограничений (по ресурсам, качеству, срокам и т.д.). Исходя из ограничений, могут быть сформулированы критерии успешности проекта. Как правило, такими критериями успешности проявляются удовлетворённость заказчика, качество, сроки, стоимость, время.

Соответственно, **проект считается успешным, если он завершён:**

- в полном объёме;
- в рамках бюджета;
- в установленные сроки;
- с заданным уровнем качества;
- при удовлетворении заказчика.

Отметим, что в разных проектах может быть разное представление об успешности: в каких-то проектах превышение бюджета (или срока) — это провал проекта, а других — совершенно не критично.

Сами работы в любом проекте делятся на две части: работы предметной области (например, для ИТ-проектов это создание кода, протяжка кабелей, установка серверов и т.д.) и работы по управлению проектом (создание планов, написание документов, проведение встреч, совещаний и т.д.).

Основой управления проектом является составление плана работ и отслеживание хода работ по нему. При этом, постоянно идёт поиск компромисса между объёмом, сроками, бюджетом, качеством и удовлетворённостью заказчика с учётом имеющихся рисков.

Треугольник управления проектом

Для любого проекта выполняется следующее равенство, которое называют треугольником управления проектом или «первым законом управления проектом»:

$$f_{\text{(Объем, сроки, стоимость, качество, удовлетворенность)}} = \text{const}$$

Изменение одного из параметров проекта автоматически влечёт за собой изменение одного или нескольких других. Треугольником его часто называют потому, что основных параметров чаще всего считается три: **сроки, бюджет, качество**. В последнее время «по умолчанию» включают объем и удовлет-

ворённость заказчика. «Геометрически» этот закон управления проектом может быть представлен таким образом: если «потянуть» один из отрезков (например, сроки), то, согласно правилам геометрии, изменятся и остальные. В русском языке этот закон отражён в известной поговорке «Мы сделаем вам быстро, качественно и недорого — выберите два из трёх». Важной особенностью связей между параметрами является их нелинейность и слабая предсказуемость. Особенно часто это проявляется в ИТ-проектах: одно небольшое изменение, например, объёма путём добавления нескольких новых требований может вызвать непропорциональное увеличение бюджета и/или сроков выполнения проекта.

Общий подход к выполнению проекта

В основе проектного управления лежит следующая идея: для получения заданного результата в проекте необходимо пройти следующую последовательность шагов.

1. Инициировать проект:

- чётко сформулировать цели проекта;
- определить кому он нужен — кто заказчик проекта, куратор проекта, определить всех основных заинтересованных лиц;
- назначить руководителя проекта.

2. Спланировать проект (сначала укрупнено, потом детально):

- декомпозировать цели на результаты, которые необходимо получить и работы, которые нужно для этого выполнить;
- определить требования к результатам проекта;
- определить необходимые ресурсы для выполнения работ (люди, оборудование, материалы), их стоимость и источник приобретения;
- установить связи между работами и их длительность, разбить работы на логические этапы, создать базовый план выполнения проекта;

- определить риски проекта;
- основываясь на имеющейся информации, определить общий базовый бюджет;
- сформировать проектную группу, распределить ответственность среди её членов;
- определить, как будет происходить обмен информацией в проекте;
- договориться, что делать в случае изменений.

3. Выполнять и контролировать проект:

- выполнять, что запланировано и проверять результат на соответствие требованиям;
- при необходимости проводить перепланирование;
- принять результаты (продукт проекта).

4. Формально завершить проект:

- подписать все необходимые документы;
- премировать и распустить команду;
- подвести итоги проекта и сформировать архив.

Эта последовательность шагов достаточно универсальна и применима к любой предметной области.

Процессы управления проектом

Выше мы привели лишь некоторый минимальный перечень шагов. Более полно (и формально) управление проектами отражено в ГОСТ 54869-2011, согласно которому управление проектом включает совокупность нескольких ключевых процессов, которые объединены в пять групп: инициация, планирование, организация исполнения, контроль и завершение проекта (Рис. 2.1.4).

В полном объёме описание ключевых процессов управления проектом приведено в Табл. 2.1.1. Большинство существующих стандартов управления проектами с различной

степенью детализации описывают примерно те же процессы.

Рис. 2.1.4. Группы процессов управления проектом.

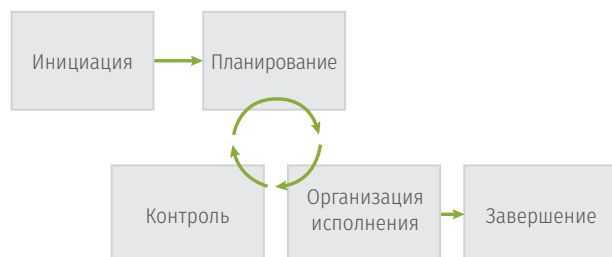
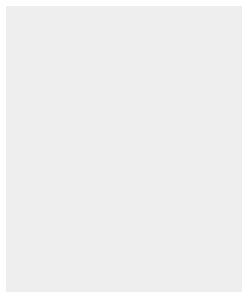
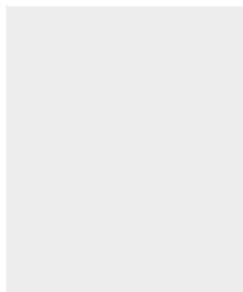
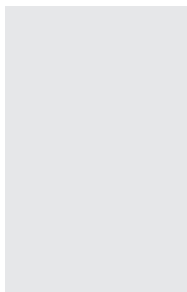


Табл. 2.2.1. Описание ключевых процессов проектного управления.

Группа	Название процесса	Цель	Выходы
Инициация	Процесс инициации проекта	Формальное открытие проекта	Определены и документированы следующие параметры проекта: • наименование проекта; • причины инициации проекта; • цели и продукты проекта; • дата инициации проекта; • заказчик проекта; • руководитель проекта; • куратор проекта.
Планирование	Процесс планирования содержания проекта	Определение требований проекта и состава работ проекта	1. Определены требования к проекту со стороны заказчика, других заинтересованных сторон проекта, а также законодательства и нормативных актов. 2. Требования проанализированы на предмет возможности их выполнения, согласованы с заказчиком проекта и документированы. 3. Определены, согласованы с заказчиком и документированы ключевые данные по продукту проекта, а именно: a) назначение, свойства и характеристики продукта; b) критерии и методы приемки продукта проекта и его составных частей; c) допущения и исключения, касающиеся продукта проекта. 4. Определены, согласованы с заказчиком и документированы работы проекта, а также допущения и исключения, касающиеся работ проекта.
Планирование	Процесс разработки расписания проекта	Определение дат начала и окончания работ проекта, ключевых событий, этапов и проекта в целом	1. Определены взаимосвязи между работами проекта. 2. Проведена оценка длительности работ проекта. 3. Определен и утвержден график привлечения ресурсов, необходимых для выполнения проекта в срок. 4. Определено и документировано расписание проекта. 5. Утвержден базовый календарный план проекта.

Планирование	Процесс планирования бюджета проекта	Определение порядка и объема обеспечения проекта финансовыми ресурсами	1. Определена и документирована структура статей бюджета проекта, позволяющая контролировать затраты на проект в ходе его реализации. 2. Определена плановая стоимость всех ресурсов проекта (материальных и людских) с учетом всех известных ограничений на их использование. 3. Определена стоимость выполнения работ проекта. 4. Утвержден базовый бюджет проекта. 5. Определен и документирован порядок поступления в проект денежных средств.
Планирование	Процесс планирования персонала проекта	Определение порядка обеспечения проекта человеческими ресурсами	1. Определены и документированы роли участников проекта, их функции и полномочия. 2. Определен численный и квалификационный состав команды проекта, а также требования к условиям труда. 3. Персонально определены основные члены команды проекта.
Планирование	Процесс планирования реагирования на риски	Определение основных рисков проекта и порядка работы с ними	1. Выявлены и документированы риски проекта. 2. Проведены оценка и ранжирование по вероятности и степени влияния на результат проекта всех идентифицированных рисков. 3. Разработаны мероприятия по изменению вероятности и степени влияния наиболее значимых рисков, а также созданы планы реагирования на случай возникновения таких рисков. 4. Учтены результаты разработки упреждающих мероприятий по реагированию на риски, в связанных с ними планах.
Планирование	Процесс планирования обмена информацией в проекте	Определение порядка обмена информацией между лицами, участвующими в реализации проекта и заинтере-	1. Определены все участники информационного обмена, а также их потребности в информации. 2. Определены методы и средства распространения информации по проекту.

Планирование		рессованными в результатах проекта	3. Определена процедура разработки, согласования, утверждения, распространения проектных документов. 4. Определены правила и место хранения информации по проекту.
	Процесс планирования управления изменениями в проекте	Определение порядка работы с изменениями в проекте	Определен и документирован процесс работы с изменениями в проекте, а именно: а) выявление изменений; б) согласование и утверждение изменений; в) организация учета версий документов и продуктов проекта; г) доведение информации об изменениях до заинтересованных сторон.
Организация исполнения	Процесс организации исполнения проекта	Организация выполнения проекта согласно разработанным планам	1. Выполнены запланированные работы. 2. Получены продукты проекта. 3. Изменения осуществлены согласно принятым в проекте правилам. 4. Выполнены намеченные корректирующие и предупреждающие действия. 5. Актуализированы документы по управлению проектом.
Контроль	Процесс контроля исполнения проекта	Проверка соответствия процессов и продукта проекта установленным требованиям	1. Документированы результаты регулярной проверки состояния проекта, в частности, отклонения от планов, и проанализированы с целью определения причин отклонений. 2. Произведена оценка соответствия продукта проекта требованиям к нему. 3. Сформированы корректирующие и предупреждающие действия по результатам проверки. 4. Отчеты о выполнении работ проекта соответствуют утвержденной системе отчетности по проекту.
Завершение	Процесс завершения проекта	Формальное закрытие проекта	1. Проведена и документально оформлена приемка продукта проекта заказчиком.



2. Проведено закрытие всех договоров по проекту (в случае их наличия).
3. Документировано окончание проекта.
4. Сформирован архив проекта.
5. Команда проекта и основные заинтересованные стороны проинформированы об окончании проекта.

Стандарты управления проектами

Управление проектами является зрелой профессиональной сферой, но далеко не наукой. Фактически, управление проектами на текущий момент – это набор наблюдений, лучших практик, применение которых, как было кем-то и когда-то замечено, даёт положительный эффект. В этих условиях очень важную роль играют статьи, публикации и другие информационные материалы (они дают примеры этих практик), но особенно важную роль играют именно стандарты — при их разработке собираются, анализируются и сводятся в единый документ все достижения сообщества руководителей проектов.

Таким образом, стандарты по управлению проектами решают несколько задач:

1. Концентрация лучшей практики (best practice) — стандарты в области управления проектами содержат лучший мировой опыт в этой области.

2. Взаимодействие — стандарты являются основой взаимодействия и общей терминологии, особенно в больших и международных проектах.

3. Сертификация — стандарты являются основой для сертификации как организаций, так и отдельных специалистов в области управления проектами.

4. Системная картина — стандарты отражают общую картину области менеджмента «управление проектами». Необходимо отметить, что подавляющее большинство существующих стандартов не являются «истинной

в последней инстанции» — это именно сборники идей в помощь проектному менеджеру, «ящик с инструментами», из которого менеджер должен создать набор, подходящий для его конкретного проекта.

Наиболее юридически точно эта мысль выражена в американском стандарте **PMBOK**.

Аналогичная ситуация сложилась и в области управления ИТ-процессами — ITIL это не «истина в последней инстанции», а свод лучших практик и рекомендаций на их основе. Однако есть и исключения из этого правила. Например, все требования ГОСТ Р 54869-2011 являются обязательными для исполнения для всех проектов. Помимо этого, чаще всего как обязательные к исполнению, строятся корпоративные стандарты по управлению проектами.

Стандарты в области управления проектами разрабатываются органами стандартизации на международном и национальном уровне, а также профессиональными организациями в области управления проектами, наиболее авторитетными из которых являются следующие:

- Международная организация по стандартизации (ISO) опубликовала стандарт ISO 10006 «Системы менеджмента качества. Руководящие указания по менеджменту качества проектов» и ISO 21500 «Руководство по менеджменту проектов».
- Международная ассоциация проектного менеджмента (International Project Management Association, IPMA) основана в Ев-

Основной целью Руководства РМВОК является выделение той части Свода знаний по управлению проектами, которая обычно считается хорошей практикой.

Термин «выделение» предполагает подготовку обобщённого обзора, а не исчерпывающего описания. «Обычно считается» означает, что описываемые знания и практики применимы к большинству проектов в большую часть времени, причём относительно их значения и пользы в целом существует консенсус.

«Хорошая практика» означает, что в целом существует согласие относительно того, что правильное применение этих навыков, инструментов и методов способно повысить вероятность успеха для широкого диапазона различных проектов.

Хорошая практика не означает, что описываемые знания должны всегда одинаковым образом применяться во всех проектах; возможность их применения для каждого конкретного проекта определяется командой управления проектом.

ропе в 1967 году и объединяет 45 национальных ассоциаций (Россию в IPMA представляет национальная ассоциация управления проектами COBHET). Основным стандартом, разработанным IPMA – ICB (IPMA Competence Baseline, 4-я версия выпущена в 2015 году) – определяет требования к квалификации специалистов в области управления проектами и является основой для международной сертификации. В соответствии с правилами и требованиями IPMA в России разработаны национальные требования к компетенции менеджера проекта и программа сертификации специалистов по управлению проектами. Специалисты, прошедшие сертификацию по этой системе, получают сертификаты международного образца, которые признаются во всём мире.

- Институт управления проектами США (Project Management Institute, PMI) сегодня «де-факто» также можно назвать международной профессиональной организацией. PMI основана в 1969 году в США и включает более 200 национальных отделений, в том числе несколько российских. PMI ведёт активную разработку стандартов в области управления проектами. Опубликованы 3 основных стандарта, регламентирующих процессы управления

на уровне проекта, программы, портфеля проектов и более 10 дополнительных стандартов (The Standard for Program Management, Second Edition, The Standard for Portfolio Management, Second Edition и др.). Дополнительные стандарты определяют требования и к отдельным методикам управления проектами (разработка иерархической структуры работ, разработка календарного плана, управление рисками и другие), и к применению проектного менеджмента для определённых типов проектов (Practice Standard for Work Breakdown Structure, Practice Standard for Earned Value Management, Practice Standard for Scheduling, Practice Standard for Conjunction Management и др.).

По областям применения существующие стандарты могут быть разделены на следующие группы:

- Применимые к отдельным объектам управления (проект, программа, портфель проектов) и регламентирующие соответствующие процессы управления.
- Применимые к субъектам управления (менеджеры проектов, участники команд управления проектами) и определяющие требования к знаниям и квалификации соответствующих специалистов, а также к процессу оценки квалификации.

- Применимые к системе управления проектами организации в целом и позволяющие оценить уровень зрелости организационной системы проектного менеджмента.

В России приказом Федерального агентства по техническому регулированию и метрологии (Росстандарта) от 27.08.2008 в техническом комитете по стандартизации ТК 100 «Стратегический и инновационный менеджмент» создан подкомитет «Менеджмент проектов», в задачи которого входит разработка серии стандартов по управлению про-

ектами для России. В 2011 году Росстандартом утверждена серия российских национальных стандартов в области проектного управления:

- ГОСТ Р 54869—2011 «Проектный менеджмент. Требования к управлению проектом»
- ГОСТ Р 54870—2011 «Проектный менеджмент. Требования к управлению портфелем проектов»
- ГОСТ Р 54871—2011 «Проектный менеджмент. Требования к управлению программой».

Табл. 2.1.2. Классификация и сопоставление стандартов.

Классификация стандартов	Наиболее известные мировые стандарты	Российские аналоги	Использование в России
Международные стандарты, определяющие общие требования к процессам управления проектом.	ISO 10006 «Системы менеджмента качества. Руководящие указания по менеджменту качества проектов». ISO 21 500 Standard for Project Management	ГОСТ Р ИСО 10006 2005 «Системы менеджмента качества. Руководство по менеджменту качества при проектировании», 2006. ГОСТ Р ИСО 21500-2014 Руководство по проектному менеджменту	На практике ГОСТ Р ИСО 10006-2005 применяется достаточно редко, поскольку носит общий характер. ГОСТ Р ИСО 21500-2014 является одним из наиболее активно используемых стандартов
Национальные стандарты, определяющие общие требования к процессам управления проектом	A Guide to the Project Management Body of Knowledge (PMBOK Guide) A Fifth Edition. (в сентябре 2017 выходит 6-я редакция) PRINCE2 (PRojects IN Controlled Environments). OGC UK, 2009	«Руководство к своду знаний по управлению проектами». Пятое издание. PMI. 2013. Русская версия	Не является стандартом в России. Однако PMBOK широко применяется на международном уровне и является стандартом «де факто». В России также применяется достаточно широко.
Стандарты, определяющие общие требования к процессам управления программой и портфелем проектов	The Standard for Program Management, Third Edition, PMI 2013. The Standard for Portfolio Management, Third Edition, PMI 2013 Managing Successful Programmes, OGC UK, 2007	PMI «Управление портфелями» (Portfolio Management. Second Edition). Русская версия	Мало используются в России

	P2M. Program and Project Management for Innovation of Enterprises, PMCC, 2002		
Стандарты, определяющие требования к корпоративной системе управления проектами	Organizational Project Management Maturity Model (OPM3®) Knowledge Foundation 2nd Edition	Нет русскоязычных аналогов стандартов	России практически не используется
Стандарты, определяющие требования к последовательности и методикам выполнения отдельных процессов	Practice Standard for Work Breakdown Structure, 2nd Edition, PMI, 2006 Practice Standard for Earned Value Management, PMI, 2011 Practice Standard for Scheduling Second Edition, PMI, 2011 Practice Standard for Configuration Management, PMI, 2007	Стандарты, определяющие требования к корпоративной системе управления проектами	
Стандарты, определяющие требования к квалификации специалистов в области управления проектами	ICB IPMA Competence Baseline, Version 340, IPMA 2006 PMCDF Project Management Competence Development Framework, PMI, 2003	«Основы Профессиональных Знаний и Национальные Требования к Компетентности Специалистов по Управлению Проектами» (НТК 3.0), СОВНЕТ, 2010. (до конца 2017 года должна появиться новая версия) ГОСТ Р 52807-2007 «Руководство по оценке компетентности менеджеров проектов»	НТК не является официальным стандартом в России, но зарегистрирован в Росстандарте. Используется для сертификации специалистов в соответствии с требованиями IPMA

Как в России, так и в мире на сегодняшний день существуют значительные пробелы в стандартизации в области проектного управления.

Основные стандарты, применяемые сегодня в мире, разработаны профессиональными организациями в области управления проектами (PMI, IPMA, национальными ассоциациями) и, как правило, не имеют официального международного статуса.

Стандарты управления ИТ-проектом

ИТ-проекты имеют свою существенную специфику. Поэтому, несмотря на то, что к ним применимы все вышеперечисленные общие стандарты по управлению проектами, в мире существует ряд стандартов, специально нацеленных на выполнение именно ИТ-проектов. Одним из наиболее известных и популярных стандартов является **PRINCE2**. Стандарт на настоящий момент позиционируется как универсальный, но разработан он был в 1989 в Великобритании именно для выполнения ИТ-проектов для нужд государственных органов. Большим преимуществом стандарта является его глубокая проработанность и гибкость.

Необходимо также отметить неизвестный у нас в стране стандарт Hermes, который обязателен для выполнения на всех государственных ИТ-проектах Швейцарии. Стандарт включает в себя подробное описание подхода и полный набор проектных шаблонов на всех официальных языках этого государства: английском, французском, немецком и итальянском.

Вопреки распространённому мнению, **серия стандартов ГОСТ 34** не имеет отношения к управлению проектами. Этот стандарт относится к жизненному циклу автоматизированных систем. И, хотя он содержит отдельные элементы управления проектами (например, документы при создании автоматизированных систем), на него нельзя опираться при внедрении процессов управления ИТ-проектами в организации.

Существует также ряд документов, которые нельзя назвать стандартами в полном смысле этого слова, скорее, это разработанные

различными организациями частные своды знаний и методологии по выполнению ИТ-проектов:

- **Software Engineering Body of Knowledge (SWEBOK)** — документ, созданный комитетом Software Engineering Coordinating Committee. Назначение SWEBOK — в объединении знаний по инженерии (разработке) программного обеспечения.
- **Rational Unified Process (RUP)** — методология разработки ПО, созданная компанией Rational Software (сейчас часть IBM), очень жёсткая, глубоко проработанная и «тяжёлая». Ввиду сложности внедрения используется редко.
- **Microsoft Solutions Framework (MSF)** — методология разработки программного обеспечения, предложенная корпорацией Microsoft. Представляет собой согласованный набор концепций, моделей и правил. MSF описывает управление людьми и рабочими процессами в ходе разработки ПО.

Также практически у каждого крупного производителя бизнес-приложений существует своя стандартная методология внедрения: Project Management Method (PJM) / AIM у Oracle, Accelerated SAP (ASAP) / Global ASAP у SAP и т.д. Все эти стандарты опираются на международные стандарты управления проектами, в той или иной мере используют перечисленные выше стандарты и своды знаний по управлению ИТ-проектами.

Следует отметить отдельный класс стандартов — семейство так называемых **«лёгких» (Agile) методологий разработки ПО** (X, Lean, Scrum и др.).

Особенности выполнения ИТ-проектов

Вопрос «Что такое ИТ-проект?» не так очевиден, как кажется на первый взгляд. Ведь ИТ-проект может быть частью большого бизнес-проекта, а может быть отдельным, самостоятельным проектом, при этом включающим существенную бизнес-часть (например, перестроение существующих бизнес-процессов). Один из примеров такого разделения дан во врезке.

го директора компании PM Expert Андрея Конузова):

1. Первая отличительная особенность ИТ-проектов лежит на поверхности. Она заключается в том, что любой просчёт или ошибка, как правило, очень быстро становятся известными широкому кругу людей. Если, например, осуществляется замена сервера или настройка какой-либо информационной си-

Выделение ИТ-проектов в ТНК-ВР

Разделение бизнес-проектов и ИТ-проектов, а также соответствующих составных частей комплексных проектов, в ТНК-ВР было сделано введением понятия ИТ-компоненты проекта и производных от нее определений:

ИТ-компонента бюджета проекта (ИТ-компонента) — расходы на работы внутренних и внешних трудовых ресурсов ИТ-службы, аппаратное обеспечение, телекоммуникационное оборудование, лицензии, затраты на первый год поддержки, консалтинговые работы по разработке или настройке приложений, а также на расширение пропускной способности канала.

ИТ-проект (IT driven project) — проект, в котором ИТ-компонента составляет большую часть бюджета. Бизнес-проект с ИТ-компонентой (business driven project) — проект по развитию бизнеса, включающий ИТ-компоненту с бюджетом более 10 тыс. долл.

Приведённый во врезке вариант выделения ИТ-проектов не идеален, в других организациях подход может отличаться и, например, основываться на трудозатратах. Это не столь важно, главное — чётко определить, что именно мы будем называть ИТ-проектом. Если неправильно рассчитать и поставить планку слишком высоко, то множество работ, которые должны выполняться как проекты, будут вестись «как получится» с соответствующими результатами. Если же занижить планку, то небольшие простые задачи будут обременены необходимостью готовить проектные документы, что затянет их сроки и приведёт к демотивации персонала.

Чёткая идентификация ИТ-проекта важна из-за их особенностей. Можно выделить четыре наиболее важных из них (основные тезисы взяты из интервью бывшего исполнительно-

стимы и происходит сбой, то все пользователи тут же узнают об этом. В отличие от этого, например, в маркетинговом проекте просчёты далеко не так очевидны. Можно в его рамках сделать все правильно, но, допустим, не в полном объёме учесть интересы целевой аудитории. И напрямую обвинить в этом упущении руководителя проекта довольно сложно, т.к. существует большое количество внешних факторов. В ИТ-проекте внешних факторов тоже предостаточно, но ассоциативный ряд «преступление — наказание» у участников выстраивается однозначно: кто реализовал проект, тот и виноват.

2. Вторая особенность заключается в том, что в настоящее время многие ИТ-проекты имеют колоссальные бюджеты. В крупных организациях масштабы проектной деятельности в области ИТ измеряются десятками милли-

онов долларов. Большие бюджеты, в свою очередь, подразумевают больший уровень ответственности и, соответственно, больший уровень компетенции тех людей, которые этими проектами управляют.

3. Третья особенность состоит в том, что реализация новых проектов происходит постоянно. Если, например, промышленное предприятие достаточно один раз построить, и оно будет работать, не требуя существенных регулярных инвестиций, то развитие ИТ-ин-

фраструктуры в растущих компаниях требует больших и регулярных вложений.

4. Четвертая особенность — разделение заказчика и исполнителя на уровне идеологии. Заказчиком, как правило, является бизнес, а исполнителем — ИТ-специалисты. В результате возникают трудности в выявлении требований и ожиданий от проекта, в формировании технического задания, возникают проблемы коммуникаций.

Оценка сложности ИТ-проектов

Существует большое количество разнообразных классификаций ИТ-проектов, но не существует одной общепринятой. Каждая крупная организация вводит свою классификацию для постановки внутренних целей и оценки сложности. Наиболее распространена классификация ИТ-проектов по сложности и по видам работ. Пример классификации ИТ-проектов по уровню сложности по 17 различным параметрам (на основе опыта ТНК-BP), приведён в Табл. 2.1.3.

Долговременность архитектуры Используется существующая архитектура Разработанная архитектура может использоваться ещё в нескольких проектах Разрабатываемая в рамках проекта архитектура будет стратегической для всех дальнейших ИТ-проектов.

Приведённая классификация определяет линейную одноранговую шкалу уровня сложности ИТ-проекта (низкая/средняя/высокая), которая часто применяется на практике, однако, в целом ряде случаев её оказывается недостаточно.

Табл. 2.1.3. Пример классификации ИТ-проектов.

Параметры проекта	Низкая сложность «1»	Средняя сложность «2»	Высокая сложность «3»
Бюджет	До 100 тыс. долл.	100 тыс. – до 1 млн. тыс. долл.	1 тыс. долл. и выше
Длительность	До 6 месяцев	7 – 12 месяцев	13 и более месяцев
Численность проектной команды (включая основных представителей заказчика)	До 5 человек	6 – 20 человек	21 и более человек
Географическая распределенность	Пользователи находятся на 1 площадке или/и в 1 регионе	Пользователи находятся на 2-3 территориально удаленных площадках или/и в 1-2 регионах	Пользователи находятся на более чем 3 площадках или/и в более 2 регионах

Вовлеченность бизнес-направлений (БН), бизнес-функций (БФ) и входящих в компанию юридических лиц	1 БН/БФ или/и 1 юридическое лицо	2 БН/БФ или/и 2-3 юридических лица	3 и более БН/БФ или/и 3 и более юридических лиц
Инновационность используемых технологий	В компании есть системы, построенные на данной технологии. В проекте будут участвовать сотрудники компании, имеющие опыт работы с ней	В компании был проведен пилотный проект, есть незначительный опыт использования данной технологии или/и существует опыт внедрения в мире, есть доступная проектная команда с требуемой компетенцией	Полностью новая технология
Влияние на корпоративную инфраструктуру	1 новая система и отсутствие новых каналов связи	2-3 системы или/и расширение существующих каналов связи	Более трех систем или/и построение новых каналов связи
Количество пользователей	До 30 пользователей	От 31 до 100 пользователей	Более 100 пользователей
Изменения в бизнес-процессах	Незначительное изменение 1-2 бизнес-процессов 3-го уровня	Изменение 3 и более процессов 3-го уровня	Изменение 3 и более процессов 3-го уровня и изменения существующей организационной структуры
Взаимосвязь и зависимость от других проектов	Отсутствует	Зависимость от 1 проекта	Зависимость от 2 и более проектов
Схема контрактования	1 генеральный подрядчик, находящийся в регионе заказчика	1-2 генеральных заказчика или подрядчик находится в другом регионе	3 и более генеральных подрядчиков
Интеграция	Отсутствие интерфейсов интеграции или используются существующие интерфейсы	1-3 разрабатываемых или дорабатываемых интерфейсов интеграции	4 и более разрабатываемых или дорабатываемых интерфейсов интеграции
Критичность для бизнеса	Контролируется руководством не выше директора департамента	Контролируется на уровне вице-президента компании	Включен в KPI какого-либо блока компании и контролируется на уровне не ниже вице-президента

Стабильность окружения	Стабильная ситуация. Определены перспективы/стратегия развития подразделения	Происходит незначительная реорганизация оргструктуры и/или бизнес-процессов. Происходит пересмотр стратегии развития	Происходит существенная реорганизация. Стратегия и планы не определены
Изменения в ИТ-процессах поддержки	«0» - нет изменений «1» - незначительные изменения в SLA	Значительное изменение существующих сервисных линий	Создание новой сервисной линии
Подрядчик	Нет внешнего подрядчика или один подрядчик с большим опытом совместной работы	Больше одного подрядчика. Нет большого опыта взаимодействия конкретно с этими подрядчиками	Больше трех подрядчиков, между которыми необходимо наладить взаимодействие
Долговременность архитектуры	Используется существующая архитектура	Разработанная архитектура может использоваться еще в нескольких проектах	Разрабатываемая в рамках проекта архитектура будет стратегической для всех дальнейших ИТ-проектов

Более эффективным подходом является не прямое суммирование баллов по всем вышечисленным «осям координат» (технологическая сложность, эффективность для бизнеса и т.д.), а классификация каждого проекта по всем этим осям отдельно (можно сказать, в n-мерном пространстве). На основе этих параметров можно построить профиль управления проектом, включающий основные элементы управления, которые будут применяться на данном проекте (документы, этапы, роли и т.д.).

При этом, каждая ось оказывает своё влияние на то, какие элементы управления будут входить в профиль. Так, например, наличие представителей нескольких подразделений в проектной команде требует обязательной подготовки матрицы ответственности, а наличие крупного бюджета, распределённого между несколькими подрядчиками — рабочего финансового плана проекта и т.п.

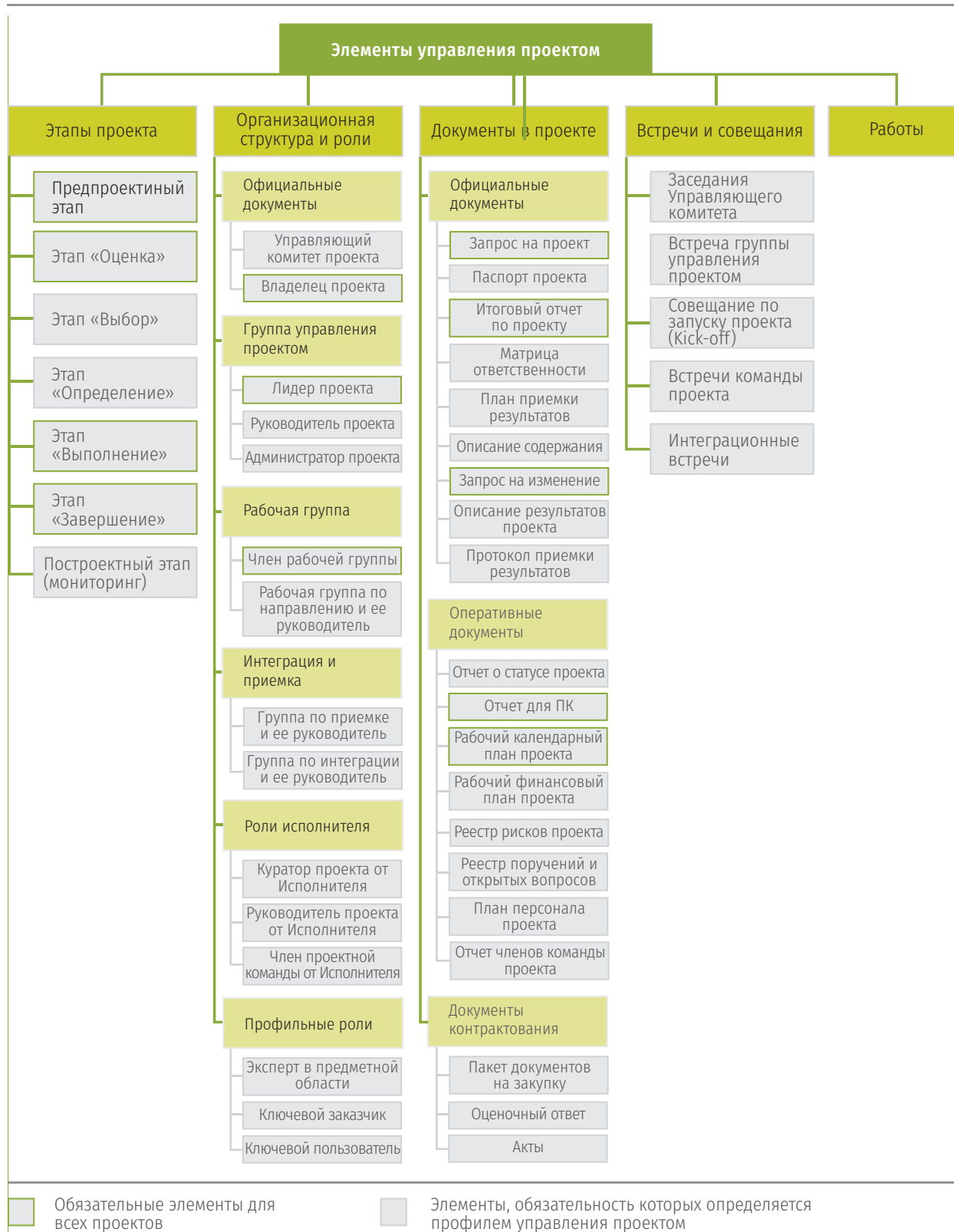
Пример обязательных и зависящих от профиля элементов управления проектами показан на Рис. 2.1.5. Развитие и обсуждение данного подхода можно найти на сайте трёхуровневой Российской инструментальной модели Rim-III.ru.

Если вернуться к оценке сложности ИТ-проектов, то, как правило, наиболее сложными с точки зрения работы по управлению проектом, но и наиболее интересными проектами являются проекты по разработке и внедрению приложений.

Проекты реорганизации и консалтинговые проекты хотя и имеют свою специфику, но для ИТ-подразделений они обычно невелики по объёму.

Инфраструктурные проекты могут быть очень сложны с технологической точки зрения, но, если не брать масштабные проекты вроде создания новых ЦОД или массового перехода пользователей на новую стандарт-

Рис. 2.1.5. Элементы управления проектом: обязательные и зависящие от профиля проекта.



ную систему, они не очень сложны с точки зрения управления проектами. Инфраструктурные проекты чаще всего отличаются своей стоимостью, но не сложностью.

При этом необходимо отметить, что стоимость сама по себе может быть важным критерием выделения работы в отдельный проект. Настолько, что иногда имеет смысл выделять

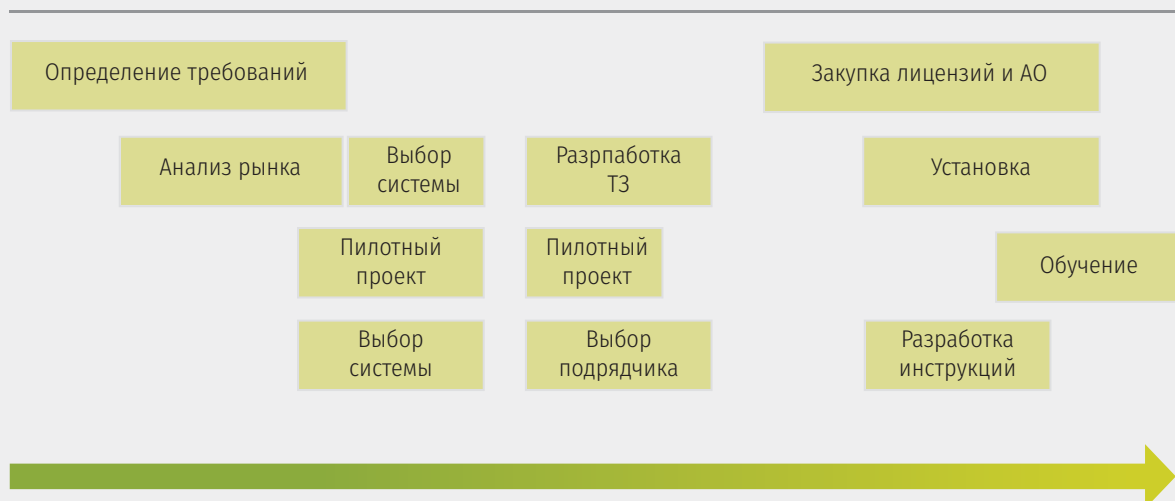
закупку оборудования и лицензий в отдельный проект. Чаще всего это происходит при планировании очень крупной закупки. Поэтому далее мы кратко опишем последовательность шагов и этапы проектов разработки и внедрения приложений без доработки и с существенной доработкой.

ПРОЕКТЫ ВНЕДРЕНИЯ ПО

Внедрение коробочного ПО (out-of-the-box) без доработки

Ниже мы опишем последовательность шагов и работы предметной области проекта, процессы управления проектом остаются теми же, что были описаны выше. Общая последовательность шагов при внедрении существующего на рынке программного обеспечения без его существенной доработки (например, установка системы «Консультант») показана на Рис. 2.1.6.

Рис. 2.1.6. Последовательность шагов при внедрении программного обеспечения без его существенной доработки



1. Определение требований. Данный этап является ключевым для проекта любого типа. Перед тем, как что-то делать, нужно сначала более-менее определиться, ЧТО именно нужно сделать. Требования должны быть зафиксированы в документе «Требования к системе».

2. Анализ рынка. Хотя российский рынок готового ПО значительно слабее развит по сравнению с западным, на нем представлено довольно большое количество систем. Для проведения анализа можно использовать Интернет, специализированные информационные источники (например, отчёты Gartner) или внешних консультантов.

3. Выбор системы. Выбор системы должен проводиться по некоторым формальным критериям, которые выводятся из документа «Требования к системе». Данные критерии можно разделить на три категории:

а) Требования к производителю. Например, сильные позиции на рынке, наличие российского представительства (для западных систем), наличие партеров по внедрению, наличие внутреннего формализованного процесса разработки и т.д.

б) Функциональные и нефункциональные требования к системе. Каждый критерий должен соответствовать одному из требований к системе. Данные критерии должны быть чётко формализованы, чтобы по ним можно было произвести однозначную оценку каждого требования для каждой рассматриваемой системы (например, «да/нет», «отлично/хорошо/средне/плохо/функция отсутствует» и т.д.).

в) Стоимость. Должна быть проведена сравнительная оценка стоимости лицензий и требуемого аппаратного обеспечения (различные системы могут существенно отличаться по требованиям к аппаратному обеспечению). Также в неформализованном виде стоит выписать обобщённые плюсы и минусы систем. Это особенно полезно для презентации руководству. При выборе системы важно помнить, что для программного обеспечения правило «дорого, значит хорошо» не работает. То, что хорошо для одних условий может быть совсем не хорошо для других.

4. Пилотный проект (опционально). Чаще всего по описаниям и документации очень сложно составить полное понимание системы. Коммерческие предложения производителя зачастую оставляют сомнение в их адекватности. В таком случае единственным способом, который позволяет более-менее уверенно заранее утверждать, что данная программа подходит для нужд организации является проведение пилотного проекта, т.е. реальная инсталляция и использование программы в ограниченных масштабах.

5. Выбор подрядчика по внедрению (опционально). Для проектов внедрения стандартных систем выбор подрядчика не обязателен. Данная работа может быть выполнена сотрудниками ИТ-службы.

6. Разработка/согласование технического задания (опционально). Так как данный вид ИТ-проекта часто не предполагает проведения сложных работ по доработке и настройке системы, нет необходимости разрабатывать отдельный документ с подробным описанием работ и требований. Все необходимые работы могут быть формализованы в договоре поставки и внедрения.

7. Закупка лицензий и аппаратного обеспечения. Проводиться согласно правилам организации.

8. Установка, обучение, разработка инструкций по использованию системы. Данные работы могут проводиться параллельно. Для установки лицензий на рабочие места пользователей могут использоваться средства удалённого управления. Особое внимание необходимо уделить обучению, без его грамотного проведения установленное ПО будет использоваться неэффективно.

Внедрение приложений с адаптацией

Внедрение существующего на рынке решения системы с её настройкой под организацию (например, внедрение CRM и ERP-систем), является промежуточным вариантом между «чистым» внедрением и «чистой разработкой». Таким образом, этапы данного проекта

являются некоторой комбинацией этапов других видов ИТ-проектов. Важно отметить, что в этих проектах очень велика роль бизнес-заказчика — без его плотного взаимодействия с проектной группой проект обречён на неудачу.

Особенности ИТ-проектов по внедрению приложений с адаптацией следующие:

- для проектов данного типа чаще всего необходимо привлечение внешнего подрядчика;
- совершенно необходимо разработать «Техническое задание» на настройку и доработки внедряемой системы;
- крайне рекомендуется проведение пилотного проекта или разработка макета системы;
- рекомендуется проводить опытную эксплуатацию после внедрения продукта до закрытия проекта.

Разработка программного обеспечения «с нуля»

Процессу разработки программного обеспечения присущи две основные проблемы:

1. Непредсказуемость. Причиной непредсказуемости процесса является гибкость программного обеспечения — запрограммировать можно практически все что угодно. Обратной стороной является то, что это «практически все что угодно» сильно затрудняет планирование, мониторинг и управление разработкой ПО.

2. Высокая стоимость. В отличие от многих других продуктов человеческой деятельности стоимость программного продукта всегда определялась стоимостью разработки, а не тиражирования. Так как разработка осуществляется группой высококвалифицированных профессионалов в течение довольно длительного времени, то общая стоимость является весьма значительной. К сожалению, производительность данного процесса также оставляет желать лучшего. В данной главе мы не будем детально останавливаться на этом типе проектов. По этой теме существует обширная и очень качественная литература.

Контроль ИТ-проектов

Контроль проекта проектным менеджером — часть всех существующих стандартов управления проектами. Нужен ли ещё один уровень контроля — контроль самого проектного менеджера и выполняемого им проекта? В идеальном мире контроль за проектным менеджером (проектом) не нужен: проектный менеджер сам фиксирует проблемы и открытые вопросы, определит круг заинтересованных лиц, передаст им нужную информацию и организует разрешение проблем наилучшим образом. Но так как наш мир не идеален, то у многих участников ИТ-проекта

Контроль — это одна из основных функций менеджмента, наряду с планированием, организацией и мотивацией.

Основная цель контроля — понимание текущей ситуации, снижение неопределённости, повышение уверенности в благополучном исходе и своевременное принятие управленческого корректирующего воздействия. Причём, чем меньше понимание ситуации и чем больше неуверенность в конечном исходе, тем выше желание контролировать.

возникает насущная необходимость держать его под контролем. Прежде всего, это относится к CIO, который, как правило, сам не управляет ИТ-проектами, но обязан держать под контролем идущие в его организации ИТ-проекты.

Учитывая, что любой ИТ-проект, согласно своему определению, является предприятием с высокой степенью неопределённости (создание уникального результата), то для ИТ-проекта вопрос контроля актуален по определению. Тем не менее, в настоящий момент стандарты по внешнему контролю ИТ-проекта отсутствуют, в основном всё сводится к подготовке отчётности той или иной степени детальности.

Сложно сказать, насколько контроль реально помогает избежать провала проекта — на эту тему идут довольно серьёзные дискуссии, как теоретические, так практического свойства. Например, любимый ответ проектных менеджеров на просьбу как-то формализовать свою деятельность и детальнее отчитываться: «Вам шашечки или ехать?». То есть Вам документы готовить, или чтобы проект выполнялся? Тем не менее, плохая статистика успешности проектов говорит в пользу повышения степени контроля.

Так что же такое контроль? Контроль — это один из терминов, которым все интуитивно

« У нас в России все только людьми можно сделать, и всякое дело надо держать, не отпуская ни на минуту: как только отпустишь его в той мысли, что все идет само собой, то дело разоряется и люди распускаются и расходятся. »

Оберпрокурор Синода К.П. Победоносцев, конец XIX века

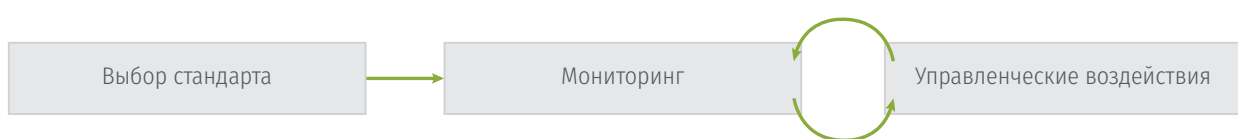
пользуются, но часто затрудняются дать его точное определение. При этом часто контроль путают с его «младшим братом» — мониторингом. В чем же разница? Согласно словарю по экономике и финансам:

Контроль (от фр. controle — проверка) — это процесс, обеспечивающий достижение системой поставленных целей и состоящий из трёх основных элементов:

- установление стандартов деятельности системы, подлежащих проверке;
- измерение достигнутых результатов и их сравнение с ожидаемыми результатами;
- корректировка управленческих процессов, если достигнутые результаты существенно отличаются от установленных стандартов.

Таким образом, **мониторинг** — это только часть контроля, важная, но не единственная. Ключевое отличие контроля — это возможность принятия управляющих воздействий. Если вы можете только смотреть на ситуацию и более ничего — это не контроль, а мониторинг. Графически это показано на Рис. 2.1.7.

Рис. 2.1.7. Три составных элемента контроля.



Объекты и субъекты контроля

Как уже было указано выше, с формальной точки зрения для целей контроля проект можно представить, как состоящий из работ и результатов. С точки зрения контроля важны три области:

- работы по управлению проектом;
- работы предметной области;
- результаты проекта, включая промежуточные.

Каждого участника контроля проекта или заинтересованного в контроле лица интересует свой аспект. Например, если контролируется управление проектом внедрения ERP-системы, то результат, т.е. как настроена система, насколько она соответствует функциональному заданию, может не контролироваться. Предполагается, что при правильном управлении проектом это в обязательном порядке будет сделано — будет запланировано тестирование и приёмка со стороны заказчика и ключевых пользователей.

Субъекты, выполняющие контроль ИТ-проектов — это бизнес-заказчик (заказчик), непосредственные руководители менеджера проекта (в том числе и CIO), проектный офис компании, служба внутреннего аудита, а также руководство и проектный офис компании

исполнителя. Две трети ИТ-проектов осуществляются при участии этих субъектов контроля. Хотя в зависимости от масштаба организации и масштаба проекта этот список может сужаться или расширяться, но практически никогда он не превращается в пустое множество.

Интересы и глубина погружения этих лиц в проект различна, но все они, так или иначе, заинтересованы в его успехе и видят (по крайней мере, должны видеть) контроль непосредственной частью своей роли. Наиболее эффективно работа над проектом протекает, когда эти роли непосредственно вписаны в корпоративную методологию управления проектом, как, например, в процессе CVP (BP и THK-BP), процессе G5 («Альфа-групп»), в методологии «Оргкомитета Сочи-2014» и др.

Построение системы контроля

При невозможности контролировать все аспекты проекта требуется выделить ключевые области и контролировать именно их. Для того чтобы контроль и, соответственно, управляющие воздействия были достаточно эффективны необходимо выстраивать систему контроля, т.е. комплекс продуманных и взаимосвязанных мероприятий, выстроенный с учётом целей контроля. Она должна быть зафиксирована и донесена до подконтрольных лиц.

Как же выстроить такую систему? К сожалению, готовых ответов не существует: слишком специфична практика управления проектом для каждой организации. На эту специфику накладывается стратегия организации, корпоративная культура, личностный аспект менеджмента и получается, что очень сложно говорить о некоей унифицированной стандартной для всех системе контроля. Но можно говорить о стандартной технологии построения системы контроля.

Для построения системы контроля нужно по-

следовательно ответить на четыре вопроса:

- Зачем контролировать?
- Что брать за эталон (с чем сравнивать)?
- Как влиять?
- Какие инструменты использовать?

Детально проработав вышеуказанные вопросы, можно получить адекватную каждому конкретному проекту систему контроля.

Зачем необходимо контролировать проект?

Не ответив на вопрос «зачем?» невозможно понять, насколько глубоко необходимо погружаться в проект. Ответы на эти вопросы зависят от той ситуации, в которой осуществляется контроль проекта, от соотношения субъектов контроля и природы самого проекта. Основных моментов, на которые здесь важно обратить внимание, также четыре. И именно они определяют **индивидуальные особенности** той или иной системы контроля проекта:

- **уровень запроса** — от кого поступил запрос на осуществление контроля: если от топ-менеджмента, то это один приоритет, если запроса не было и это личная инициатива, то другой;
- **стратегическая важность и срочность проекта** — стратегический проект требует большего внимания, низкоприоритетный — меньшего;
- **сложность и масштаб проекта** — в проекте высокой сложности больше подводных камней и больше опасность провала, а значит, необходим более плотный контроль;
- **опыт проектного менеджера** — менее опытный менеджер проекта нуждается в большем контроле и поддержке, опытному проектному менеджеру меньше нужен контроль, более того, избыточный контроль будет его раздражать.

Дополнительные критерии, которые могут повлиять на решение о необходимости и глубине контроля, перечислены в разделе о классификации ИТ-проектов (величина бюджета проекта, длительность проекта, влияние на корпоративную инфраструктуру и проч.). Эти оценки можно дать качественно, а можно формализовать, построив оценочную таблицу: например, по каждому из критериев проставить оценку от 1 до 3 и смотреть на итоговый балл по проекту. Чем выше балл — тем важнее проект с точки зрения контроля, тем больше инструментов нужно применять и тем глубже надо вникать в проект.

Поскольку ответы на эти вопросы очень индивидуальны, в результате использования вышеприведённой технологии у каждого субъекта контроля проекта получится своя индивидуальная система: у руководителя проектного менеджера одна, у CIO другая, у заказчика — третья. Наличие разных систем, разумеется, не является положительным аспектом, но, к сожалению, построить

комплексную систему управления проектом, объединяющую всех участников и при этом их ещё и удовлетворяющую, получается далеко не всегда. Это возможно только при высоком уровне зрелости проектного управления в организации.

Что брать за эталон?

Как уже было показано выше, контроль — это всегда сравнение с каким-то эталоном. Значит, нужно определить, что брать за эталон для сравнения. Каких-либо единых, общих для всех, эталонных показателей по выполнению проектов, к сожалению, не существует. Все носят рекомендательный характер. Это несколько осложняет достижение договорённости с подрядчиком и бизнес-заказчиком о том, что именно мы понимаем под «нормальным управлением проектом».

В любом случае, сравнивать можно и нужно с:

- нормативными документами самого проекта («Устав», «План», «Техническое задание» и т. д.);
- методологией и другими нормативными документами организации (если они есть);
- международными и отраслевыми стандартами (ISO, PMI, IPMA, PRINCE2 и т.д.).

Причём важно соблюдать именно такую последовательность: в первую очередь надо сравнивать с нормативными документами самого проекта, потом с методологией организации и только потом уже с международными стандартами.

Хотя существующие международные стандарты слишком обширны и не выделяют минимальные критические требования к проекту, все же возможно выделить из них общие для всех требования. Не так давно появился стандарт ГОСТ Р 54869 «Требования к управлению проектом». Хотя, как и все остальные ГОСТ, он носит рекомендательный характер, тем не менее в нём чётко указываются, по крайней мере, обязательные управленческие

результаты. Хотя существующие международные стандарты довольно сильно отличаются друг от друга, тем не менее, можно выделить из них общие для всех требования.

На основе анализа этих требований можно уверенно сказать, что для ИТ-проекта любого масштаба и сложности есть ряд общих обязательных требований к документам и информационному обеспечению (см. врезку «Обязательные минимальные требования к ИТ-проектам»). Если перечисленных документов в проекте нет, то вряд ли подобную деятельность вообще можно назвать проектом.

Как влиять?

Необходимо определиться, как можно повлиять на ситуацию, в случае, если что-то идёт не так. Это зависит от имеющихся полномочий, формальных и неформальных рычагов влияния. Если у вас нет рычагов влияния на проект, то не надо себя обманывать — вы занимаетесь мониторингом, а не контролем.

« Ubi nil vales ibi nil veils — там где ты ничего не можешь, ты не должен ничего хотеть. »

Древние римляне

Этот факт надо учесть при выборе и использовании инструментов. И, кстати, нужно ещё уточнить получится ли их применить — возможно, вам не удастся добиться даже просто получения отчётов по проекту.

Какие инструменты контроля использовать?

Теория и практика проектного управления на текущий момент располагает большим количеством инструментов, которые с успехом

можно применять для целей контроля. На эту тему есть обширный список литературы, одной из наиболее полезных книг является «Набор инструментов для управления проектами» Драгана Милошевича. Как ключевые инструменты контроля, можно выделить (в порядке убывания степени формальности и повышения эффективности):

- аудиты;
- точки принятия решений (Ворота);
- экспертные отчёты (peer reviews);
- контрольные точки;
- отчёты проекта;
- собрания;
- встречи один-на-один.

Каждый из этих инструментов имеет свои плюсы и минусы, свою сферу применимости. И по каждому можно написать отдельную статью или даже книгу.

Особенно богатая тема по отчётам. Существуют, без преувеличения, тысячи различных форматов проектных отчётов. Но самым мощным инструментом, является точка принятия решений. В ИТ-проектах и вообще инновационных проектах это очень важно — когда мы не знаем, что мы хотим получить в результате, нужно разбить проект на очень чёткие фазы, по которым осуществлять контроль. Надо сказать, что ещё более мощный инструмент — это, конечно, проектный офис. Когда у вас есть организационная единица, которая специально «заточена» на то, чтобы учить людей развивать процесс управления проектами и контролировать проект, это даёт максимальный эффект.

Определив инструмент, следует также определить и периодичность его использования.

Тёмная сторона силы

Необходимо всегда иметь в виду, что помимо плюсов, контроль проекта несёт и существенные минусы.

Во-первых, необходимо учитывать, что любой контроль требует затрат времени как того, кого контролируют, так и того, кто контролирует. Чем больше глубина и тщательность контроля, тем выше трудозатраты.

Во-вторых, и это гораздо опаснее, мало что так сильно раздражает работающего человека как постоянный и мелочный контроль. Если все плотно контролировать, члены проектной команды перестают чувствовать свою ответственность за результаты работы и теряют мотивацию.

Контролировать так, чтобы никто не вздохнул — это сильнейший демотивирующий фактор. Особенно это губительно для проектных менеджеров. Придавленный, несамостоятельный руководитель проекта — уже совсем не менеджер проекта. Менеджером проекта де-факто становитесь вы.

В-третьих, контроль несёт за собой необходимость принимать решения и, соответственно, нести ответственность за результаты этих решений. Исчезает возможность сказать: «Ну, вот они тут все напортачили. Меня на них не было...».

Учитывая всё это, стоит сильно задуматься о том, насколько нужен контроль.

Экспресс-контроль проекта

Если по каким-либо причинам нет возможности строить полноценную систему контроля, но есть срочная необходимость прямо здесь и сейчас разобраться в том, что происходит на проекте, можно посоветовать использовать инструмент под условным названием «шестиугольник контроля». Этот «шестиугольник» определяет 6 основных направлений (углов) экспресс-контроля проекта:

1. Объем работ. Каковы цели проекта и ожидаемые результаты? Где описаны требования к результатам (Техническое Задание, Технические требования, Спецификация)? Каковы географические рамки и количество пользователей? Наконец, вопросы технологии: архитектура системы и используемые технологии.

2. Бюджет. План (кем утвержден), факт и прогноз. И, соответственно, расхождения плана и факта.

3. Качество. Какие есть критерии качества выполнения работ, критерии качества получаемых результатов? Кто должен принимать результаты? Где это прописано?

4. Выгоды. Какую проблему решаем? Какие выгоды ожидает заказчик от проекта? Как именно результаты проекта помогут решить проблему заказчика и/или принести выгоды заказчику (финансовые/нефинансовые, измеримые/неизмеримые)?

5. Ресурсы. Каков состав проектной команды, подчиненность, процент загрузки и есть ли проблемы с людьми? Насколько им нравится работать на проекте? Подрядчики: кто работает, как и кем были выбраны?

6. Сроки. Каковы этапы и основные вехи проекта? Полный план работ (кем утвержден), факт, прогноз.

Что конкретно нужно сделать для экспресс-контроля проекта? Надо получить проектную документацию, ознакомиться с ней и затем, сев на пару часов с проектным менеджером, пройтись вышеприведенными вопросами по основным направлениям, расширяя глубину обсуждения в

случае необходимости. Если есть время и возможность, то 6 основных направлений экспрес-сконтроля можно дополнить **шестью дополнительными направлениями контроля** (ребра шестиугольника):

- 1. Руководство проектом.** Кто основные лица, вовлеченные в принятие решений по проекту: владелец проекта, ответственный от бизнеса, ключевые пользователи. Как часто они собираются и как принимают решения. Как они оценивают ход проекта.
- 2. Утверждения.** Кто участвует в согласовании и утверждении документов. Где это прописано.
- 3. Риски.** Где описаны. Как отслеживаются. Как часто пересматриваются.
- 4. Открытые вопросы.** Какие есть вопросы/проблемы и где они зафиксированы. Какие варианты решений. Кто и когда должен принять решение.
- 5. Коммуникации.** Есть ли план коммуникаций. Какая информация, кому и когда она передается. А действительно ли она передается. А точно ли она передается. Когда последний раз передавалась.
- 6. Изменения.** Были ли. Как отслеживаются и кем утверждаются. Журнал изменений. Предложенный подход даст не полную, но вполне целостную картинку по проекту и его состоянию. Есть и альтернативный вариант—пройтись по проекту не по предложенным направлениям, а с точки зрения областей знаний PMI PMBOK.

Система управления проектами в организации

Когда в организации начинает одновременно выполняться больше 5-10 проектов, встаёт вопрос о внедрении некоторого общего подхода. Сочетание организационной, методологической составляющей и информационной системы поддержки проектного управления в организации принято называть корпоративной системой управления проектами. При внедрении системы управления проектами всегда рекомендуется придерживаться последовательности:

Люди $\Rightarrow$ Процессы $\Rightarrow$ Технологии

То есть сначала создать специальное подразделение, ответственное за внедрение управления проектами (офиса управления проектами, центра управления проектами), обучить людей проектному управлению, внедрить временную простую методологию. Затем разработать и утвердить детальную корпоратив-

ную методологию управления проектами. И только после этого внедрять информационную систему управления проектами. Отступление от этой последовательности быстро и болезненно отзовется при внедрении системы управления проектами.

Существует два основных подхода к внедрению корпоративной системы управления проектами:

- **на уровне всей организации** — система охватывает все выполняемые организацией проекты;
- **на уровне отдельного подразделения** — чаще всего проектные офисы создаются в ИТ-службах.

В очень крупных организациях (Сбербанк, ТНК-ВР) внедряется двухуровневая система: Центральная корпоративная система управления проектами, задающая «общую рамку» и отвечающая за стратегические проекты и системы управления проектами подразделений.

Люди

Люди — это основной элемент корпоративной системы управления проектами. Без правильной работы с людьми система работать не будет. Дело в том, что внедрение системы управления проектами ощутимо меняет расклад сил в организации. Соответственно, требуется работа в четырёх направлениях:

« Нет ничего труднее, опаснее и неопределеннее, чем руководить введением нового порядка вещей, потому что у каждого нововведения есть ярые враги, которым хорошо жилось по-старому, и вялые сторонники, которые не уверены, смогут ли они жить поновому. »

Никколо Макиавелли

1. Создание организационной структуры, отвечающей за проектное управление в организации — офиса управления проектами. Ни один бизнес-процесс в организации не будет работоспособным без поддерживающей его структуры — это в полной мере касается и проектного управления.

2. Проведение масштабного обучения сотрудников. Требуется, как минимум, трёхуровневая система обучения:

- для топ-менеджмента — краткий базовый курс по основным понятиям;
- для руководителей проектов — детальный углублённый курс;
- для сотрудников (участников проектных рабочих групп) — краткие курсы по основным положениям проектного управления.

3. Создание системы мотивации сотрудников, привязанной к результатам проектов. Это критически важная задача. Если не поменять мотивацию людей, то внедрение с высокой степенью вероятности обречено на провал.

4. Формирование проектной культуры. Необходимо вести постоянную разъяснительную работу по тому, что такое проектное

управление, зачем оно нужно, какую пользу несёт. Распространять информацию об имеющихся достижениях.

Отдельно необходимо сказать о проектном офисе. **Проектный офис** — это подразделение или группа, которая определяет и поддерживает стандартные процессы, связанные с управлением проектами внутри организации.

Существует две базовые модели проектного офиса. Первая — **консультативный проектный офис**, который выполняет консалтинговую роль, обеспечивая руководителей проектов в подразделениях методической поддержкой, обучением и рекомендациями по лучшему опыту

выполнения проектов. Вторая модель — **централизованный проектный офис**, имеющий в своём штате руководителей проектов, которые выделяются подразделениям организации для работы над конкретными проектами. На то, по какой модели будет организован проектный офис и как будет укомплектован его состав, оказывает влияние большое количество организационных факторов, включая поставленные цели, традиционные влияния и культурные установки. Несмотря на то, что проектные офисы отличаются по размеру, структуре и обязанностям, существуют **семь основных функций**, которые может брать на себя проектный офис.

- **помощь проектным менеджерам** — помощь по управлению проектами менеджерам в подразделениях;
- **методология** — разработка и развитие методологии управления проектами;
- **обучение** — проведение тренингов или постановка задач по обучению для внешних провайдеров;
- **дом для руководителей проектов** — поддержка централизованного офиса, сотрудники которого выделяются для работы над

проектами (модель централизованного проектного офиса);

- **внутренний консалтинг и наставничество** — распространение лучших практик среди сотрудников организации;
- **информационная система управления проектами** — внедрение, поддержка и развитие программного обеспечения для управления проектами;
- **управление портфелем проектов.**

В исследовании проектных офисов компанией PM Expert были выделены следующие наиболее «популярные» функции, которые возложены на проектный офис в компаниях участников опроса:

1. Функции по управлению проектами:

- контроль изменений и отслеживание проблем по проектам;
- мониторинг эффективности выполнения проектов (анализ отклонений);
- анализ результатов проектов по завершении;
- контроль соблюдения методологии управления проектами;
- анализ проектов на соответствие стратегии (на этапе инициации проектов);
- обеспечение коммуникаций с функциональными подразделениями заказчиками проектов поддерживающими службами;
- управление отдельными проектами компании.

2. Функции по управлению ресурсами:

- наставничество и консультирование участников проектной деятельности;
- контроль распределения ресурсов в проектах;
- оценка эффективности руководителей проектов;
- организация и/или проведение обучения по управлению проектами.

3. Функции по управлению портфелем проектов:

- отслеживание портфеля;
- планирование портфеля (включая распределение ресурсов и разработку общего расписания);
- управление ресурсами портфеля.

Процессы (корпоративная методология)

Корпоративная методология управления проектами непереносимый и обязательный элемент корпоративной системы управления проектами. Чаще всего внедрение системы управления проектами начинается именно с неё. Как показывают исследования, даже в тех организациях, где нет остальных элементов системы управления проектами, практически всегда присутствует корпоративная методология.

Часто возникает вопрос, зачем организации нужна своя методология, если существуют признанные международные стандарты. На самом деле прямое использование их в организации практически невозможно. Как было указано ранее, они скорее являются набором «лучших практик», чем нормативным документом прямого действия. Необходима «привязка» к местным условиям: уточнение ролей и привязка их к организационной структуре, выделение именно тех процессов, инструментов и документов по управлению проектами, которые наиболее важны для проектов и культуры компании.

Корпоративная методология управления проектами должна включать как минимум следующие основные моменты:

- глоссарий;
- основные термины и определения;
- определение проекта, признаки выделения проекта;
- жизненный цикл проекта;
- описание основных ролей;
- документы проекта;

- встречи и совещания;
- процессы управления проектом;
- систему отчётности;
- шаблоны документов.

Технологии (информационные системы управления процессами)

При внедрении информационной системы управления проектами необходимо помнить ровно то же правило, что и для остальных систем: система подбирается под требования и задачи организации, а не наоборот. Информационную систему управления проектами невозможно просто поставить и начать работать. Как для ERP, CRM или других сложных систем сначала необходимо понять потребности бизнеса, потом настроить под них систему. Настройки по умолчанию не работают. Разумеется, в данном случае речь идёт о корпоративной системе, а не о локально установленным приложениям. В качестве «рисовальщика планов» Microsoft Project прекрасно работает и без настроек.

Согласно исследованию компании PM Expert,

наиболее используемые функции информационной системы управления проектами:

- календарное планирование и контроль сроков;
- учёт трудовых ресурсов;
- ведение проектной документации;
- управление бюджетом;
- таблицы учёта рабочего времени (таймшиты) сотрудников организации;
- управление рисками;
- управление потоками работ (workflow);
- ведение договоров и планирование поставок;
- учёт материальных ресурсов и механизмов.

Согласно последнему исследованию Gartner Magic Quadrant for IT Project and Portfolio Management на международном рынке систем управления проектами установилась стабильность. За последний год количество и состав лидеров не изменился, а все вендоры стали ещё ближе друг к другу. Нельзя не отметить, что в последнее время начали активно набирать силу «лёгкие» SaaS-решения по управлению проектами.

Часть 2. Управление ИТ деятельностью

Глава 2.2

Управление портфелем проектов



Павел
Алферов

Определения

В деятельности организации определяют 3 уровня, на которых происходит управление инновациями (Рис. 2.2.1):

- Уровень Проектов – должен обеспечить решение тактических задач;
- Уровень Программ – предполагает решение комплексных проблем;
- Уровень Портфелей – нацелен на достижение стратегических целей бизнеса.

Портфель состоит из других портфелей, программ и проектов.

Портфель проектов — это набор компонентов, которые группируются вместе с целью эффективного управления и для достижения стратегических целей организации. (ГОСТ Р 54870-2011 «Требования к управлению портфелем проектов»).

Дадим два определения управления портфелем проектов, формальное и неформальное.

Управление портфелем проектов подразумевает деятельность, направленную на до-

стижение стратегических целей организации путём формирования, оптимизации, мониторинга и контроля, управления изменениями портфеля проектов в условиях определённых ограничений. Управление портфелем проектов обеспечивает связь между уровнем стратегического управления в организации и уровнем управления проектами и программами. (ГОСТ Р 54870-2011 «Требования к управлению портфелем проектов»)

Управление портфелем проектов (Project Portfolio Management) переводится на практический язык управления весьма просто — это определённый свод инструментов, который помогает выбрать наиболее важные и выгодные проекты для организации, а также оценивать успешность проектов и в целом поддерживать постоянную связь между текущими наборами проектов и актуальными стратегическими приоритетами развития. Проще говоря, кто-то должен думать о том, почему мы запускаем те или иные проекты, по каким принципам мы распределяем ре-

сурсы между проектами, по каким критериям мы считаем тот или иной проект успешным или неуспешным, какие факторы обуславливают досрочную остановку или заморозку проекта и так далее. (Билл Дункан — основной разработчик свода знаний по управлению проектами PMBOK в версии 1996 года).

Офис управления проектами — PMO (Project Management Office) — структурное подразделение организации, контрольно-координационный орган, который определяет и развивает в организации стандарты бизнес-процессов, связанные с управлением проектами.

Необходимость управления портфелем проектов

Исследование известного гуру менеджмента Роберта Каплана показало, что только 10% компаний успешно реализуют стратегию. Это же исследование показало, что 60% организаций не способны привязать бюджет к стратегии. Таким образом, получается, что 90% компаний, потратив время на разработку, согласование и утверждение своей стратегии, по факту не могут её реализовать. Каплан предлагает для решения этой проблемы использовать разработанную им методику **Balanced Scorecard (BSC)**, но, как показывает практика, BSC может выступать в качестве прекрасного инструмента декомпозиции стратегических целей и средства измерения их достижения, но не как механизм реализации стратегических инициатив. Таким инструментом является портфельное управление. Портфель проектов — связующее звено между стратегией и реализацией, так как чаще всего одним единственным проектом невозможно достичь стратегических целей. В тех случаях, когда пытаются все-таки обойтись одним проектом, чаще всего из такого проекта получается неуправляемый «монстр».

В данной главе мы сконцентрируем внимание

именно на управлении портфелями ИТ-проектов, но большая часть инструментов применима и для других сфер.

Управление проектами и портфелями проектов можно разделить по двум характеристиками.

По целям:

- цель управления проектом — поставка продукта проекта с надлежащим качеством, вовремя и в рамках бюджета;
- цель управления портфелем проектов — получение наибольшего эффекта от реализации всей совокупности проектов.

По субъектам управления:

- при управлении проектом субъект — в первую очередь руководитель проекта, а также исполнители проекта;

Рис. 2.2.1. Три уровня управления инновациями в организации.



Рис. 2.2.2. . Место управления портфелем проектов и управления проектами в управлении организацией.



- при управлении портфелем проектов субъект — высшие руководители организации и заказчики (спонсоры) проектов, те, кто принимает решения об инвестициях в тот или иной проект.

Эта идея разделения проектного и портфельного управления визуально отражена в стандарте PMI по управлению портфелем (Рис. 2.2.2).

Управление портфелем проектов может осуществляться как на уровне всей организации, так и на уровне отдельного подразделения, например, ИТ-службы. Даже в организациях, в которых не внедрено портфельное управление, довольно широко распространённым является выражение «портфель ИТ-проектов», которое означает совокупность всех

ИТ-проектов, выполняемых в организации.

В каких случаях нужно внедрять управление портфелем ИТ-проектов? Вот какой ответ на вопрос дают эксперты **Gartner**:

- Бизнес жалуется на то, что ИТ тратит много денег и приносит мало пользы;
- Процесс принятия решений по запуску, остановке и рассмотрению ИТ-проектов не структурирован;
- Реализуются маловажные ИТ-проекты, но в то же время не выполняются стратегически важные проекты;
- Происходит дублирование проектов между бизнес подразделениями;
- ИТ все время не хватает ресурсов на выполнение проектов;
- Отсутствует приоритизация потребностей бизнеса в ИТ, ИТ-департаменту приходится выполнять все желания бизнеса;
- Проекты выполняются с задержкой;
- Проекты продолжают выполняться даже когда они вышли за рамки бюджета и сроков или польза от их реализации стала сомнительной. Управление портфелем ИТ-проектов может позволить решить или в значительной степени ослабить остроту этих проблем.

Стандарты по управлению портфелем проектов

В мире существует несколько стандартов по управлению портфелями. Наиболее известным из них является стандарт PMI The Standard for Portfolio Management, Third

Edition, 2013. Он продолжает интегрированную линейку взаимосвязанных стандартов Института управления проектами PMI, в которую входят PMBOK Guide, The Standard For

Program Management и Organizational Project Management Maturity Model.

В 2011 году в России был разработан свой стандарт ГОСТ Р 54870-2011 «Требования к управлению портфелем проектов». Стандарт устанавливает требования к управлению портфелем проектов для обеспечения эффективного достижения целей организации

и повышения качества принимаемых решений при формировании, мониторинге и контроле реализации портфеля проектов. Требования стандарта могут быть применены для управления любыми портфелями проектов независимо от характеристик компонентов, входящих в портфель.

Роли в управлении портфелем проектов

ГОСТ Р 54870-2011 устанавливает, что ролевая (организационная) структура управления портфелем проектов может в значительной степени различаться в зависимости от их специфики, но при управлении любым портфелем проектов должны быть определены следующие роли:

- комитет управления портфелем проектов — коллегиальный орган, образованный для принятия наиболее важных решений по управлению портфелем проектов;
- руководитель портфеля — лицо, ответственное за текущее управление портфе-

лем проектов и отчитывающееся перед комитетом управления портфелем проектов;

- офис управления портфелем проектов — организационная структура, предназначенная для административной поддержки руководителя портфеля проектов и комитета управления портфелем проектов. При этом необходимо отметить, что для каждого компонента портфеля (проекта или программы) согласно ГОСТам по управлению проектами и программами должны быть сформированы свои органы управления.

Процессы портфельного управления

Процесс управления портфелем проектов — это циклический процесс выбора и управления оптимальным набором проектно-ориентированных инвестиций, дающих максимальную полезность. В целом он осуществляется по циклу, построенному в духе цикла PDCA (Рис. 2.2.3).

На этапе определения перечня проектов происходит сбор актуальной информации, обеспечивающей «прозрачность» портфеля проектов. На этапе анализа — анализ проектов по ключевым показателям на основании различных аналитических представлений. На этапе оптимизации — выбор оптимального портфеля проектов. И на этапе реализации — выполнение портфеля проектов и отчётность о показателях его выполнения.

Более детально процессы управления портфелем проектов описаны в **ГОСТ Р 54870-2011**. Согласно стандарту, управление портфелем

Рис. 2.2.3. Цикл управления портфелем проектов.



проектов представляет совокупность следующих процессов:

1. Группа процессов обеспечения управления портфелем:
 - процесс сбора информации об условиях, ограничениях и требованиях к портфелю проектов;
 - процесс формализации процедур управления и параметров оценки портфеля проектов;
2. Группа процессов формирования портфеля проектов:
 - процесс идентификации компонентов портфеля;
 - процесс оценки компонентов портфеля;
 - процесс расстановки приоритетов;
 - процесс оптимизации и балансировки портфеля проектов;
 - процесс авторизации портфеля проектов;
3. Группа процессов мониторинга и контроля портфеля проектов:
 - процесс контроля реализации портфеля проектов;
 - процесс управления изменениями.

Процессы управления портфелем проектов выполняются последовательно, с определённой цикличностью. Общая взаимосвязь понятий портфельного управления согласно ГОСТ Р 54870-2011 показана на Рис. 2.2.4.

Как можно видеть из предыдущего изложения портфельное управление, теоретически, не слишком сложный инструмент управления: он достаточно прост (ГОСТ по управлению портфелем составляет всего 13 страниц), хорошо описан (кроме ГОСТ есть ещё десятки и сотни книг) и весьма эффективен. Тем не менее, в России он приживается с трудом. Можно назвать три ключевые причины этого:

1. Неготовность следовать формализованным процессам на уровне топ-менеджмента. Зачастую решения в руководстве российских компаний принимаются по неформальным критериям, непрозрачным и непонятым со стороны (пессимисты называют это «Византийской моделью управления»).

2. Низкий уровень зрелости организации с точки зрения управления проектами. Хотя теоретически внедрение портфельного управления можно вести параллельно с вне-

Рис. 2.2.4. Взаимосвязь понятий управления портфелем проектов. Сложности при внедрении управления портфелем проектов в организации



дрением проектного управления, на деле всё-таки требуется определённый (достаточно высокий) уровень зрелости организации. То есть с момента, когда началось внедрение проектного управления должно пройти хотя бы 1,52 года для того, чтобы менеджмент разобрался в возможностях этого инструмента.

3. Сложность интеграции с существующими бизнес процессами. Чаще всего в организации существует отработанный процесс бюджетирования, а также инвестиционный процесс. Увязка портфельного управления с ними очень сложная задача как с технической, так и с политической точки зрения. И, естественно, существуют стандартные сложности внедрения любого сложного менеджерского решения: неполное понимание топ-менеджмента, саботаж менеджеров среднего уровня, сопротивление исполнителей и т.д.

Если вы решили внедрять портфельное управление, то сначала необходимо заручиться поддержкой высшего руководства, а затем придерживаться той же схемы, что и при внедрении проектного управления:

Люди $\Rightarrow$ Процессы $\Rightarrow$ Технологии

В общем случае, последовательность шагов при построении портфельного управления следующая.

Шаг 1. Формирование портфельного офиса. Сначала должно быть сформировано подразделение, ответственное за управление портфелем проектов (по принципу «Если нет ответственного за процесс, не существует и процесса»). Чаще всего оно создаётся на основе офиса управления проектами (РМО), инвестиционного подразделения или подразделения, отвечающего за стратегию.

Шаг 2. Разработка методологии портфельного управления. Затем необходимо раз-

работать процессы управления портфелем проектов — обязательно жёстко должен быть зафиксирован порядок запуска проектов, их завершения и внесения изменений в утверждённые параметры проектов. Должно быть запрещено и резко пресекаться выделение финансовых ресурсов в обход установленных продуктов. Так же на этом шаге определяются:

- критерии оптимизации портфеля проектов, по которым будет проводиться балансировка портфеля;
- критерии эффективности, по которым будет производиться мониторинг реализации проектов портфеля.

Шаг 3. Формирование проектного/портфельного комитета. Необходимо создать упомянутый в стандартах комитет по управлению портфелем проектов. Названия могут быть любые — Проектный Комитет, Портфельный Комитет, Комитет по координации, планированию и контролю и т.д. Члены Комитета должны понимать общий подход и процесс работы. Должны быть предусмотрены меры мотивации для всех вовлечённых в процесс участников.

Шаг 4. Формирование первичного портфеля ИТ-проектов. Сбор с подразделений информации по проектам. Совмещается с обучением/просветительской деятельностью по создаваемому в организации процессу.

Шаг 5. Выполнение проектов портфеля и регулярная (ежемесячная) отчётность по всем запущенным проектам. Единая прозрачная отчётность по всем проектам портфеля является одним из существенных преимуществ внедрения портфельного управления для высшего руководства. Чаще всего отчёт по портфелю состоит из нескольких основных разделов:

- Аналитика по портфелю (графики и диаграммы);
- Обновление по портфелю (запущенные, остановленные, завершённые проекты);

- Общий статус проектов и ключевых контрольных точек;
- Краткая информация (на одной странице) по каждому проекту (или только по ключевым проектам) После разработки согласования процессов необходимо их протестировать на некотором заранее определённом подмножестве проектов. Примеры отчётов по портфелю проектов представлены на Рис. 2.2.5, Рис. 2.2.6 и Рис. 2.2.7.

Шаг 6. Внедрение информационной системы для управления портфелем. Только после тща-

тельного тестирования на основе утверждённых процессов можно внедрять информационную систему и обучать людей. Необходимость внедрения информационной системы определяется, в общем, двумя основными параметрами: количеством проектов в портфеле и необходимостью просчитывать варианты/сценарии. Если количество проектов невелико (до полусотни), и сложные сценарии просчитывать не нужно, то без системы портфельного управления вполне можно обойтись, обсчитывая все в Microsoft Excel.

Табл. 2.2.5. Отчёт по статусам проектов в портфеле проектов.

	 Завершённые проекты	 Приостановленные проекты	 Новые проекты
Маркетинг	• PU Bitumen office move • Migration of 4 Marketing subs to corporate domain	• Cashless payments for Retail sites • Dispatching, centralization and optimisation (Secondary Logistics Automation)	• Jupiter integration • RAP Phase II • Yaroslavl migration to CORP domain • Cashless payment system in Yaroslavl • Lubes migration to CORP domain
Переработка	• 2-nd Stage Rollout Corporate LIMS • Segregation of Saratov Business and Technological LAN	• Carson availability tool for Refining	• Material Balance and Technological Monitoring in Saratov (PI/SF) • RNPК-Follow me — printing managements system in RNPК • Refining capital projects office setup
STL	• Exposure Tracking • Supply chain study (Supply chain support) • STL screensaver		
ПУЭК	• ERP template creation • ProMis		• Business Continuity planning • Apps Support Mgmt Centre (ASMC) development

Табл. 2.2.6. . Общий статус по портфелю проектов.

№	Название проекта	Статус					Стадия CVP	Команда проекта			Бюджет, 2008, \$, тыс.		Временные рамки		Комментарии
		Объем	Бюджет	График	Ресурсы	Выгоды		Гейдкипер	Директор проекта	Менеджер проекта	План	Факт	Дата начала	Дата окончания	
1	Lubes ERP	●	●	●	●	●	Выполнение	Габалда	Петрик	Чумак	2272	65	01.10.2005	26.09.2008	Завершается фаза интеграционного тестирования
2	Yaroslavl migration to corp. domain	●	●	●	●	○	Оценка	-	Кашин	Лабудев	100	0	04.03.2008	31.12.2008	Драфт ФМ подготовлен и будет вынесен на утверждение МС после выбора нового офиса для Ярославнефтепродукт. Объем, бюджет и график зависят от этого вопроса
3	Cashless payments system implementation (Yaroslavl)	●	●	●	●	●	Пред-оценка	-	-	Матиец	200	0	01.04.2008	30.08.2008	Бизнесом выбран вариант внедрения локальной системы с дальнейшей миграцией в Магистраль-кард
4	Jupiter integration	●	●	●	●	○	Пред-оценка	-	-	Матиец	2175	104	01.04.2008	31.12.2008	Программа проекта включает: (1) - обеспечение лицензионной чистоты ПО; (2) - приведение к стандартам ИТ-безопасности; (3) - миграция в корпоративный домен; (4) - автоматизация АЗК. Менеджер проектов (1, 2, 3) - Сергей Матиец
5	Retail Automation pilot	●	●	●	●	●	Выполнение	Фейзунлин	Устинов	Королев	110	1	31.10.2004	31.09.2008	Тираж в Саратове завершен. Ведется стабилизация ПО, должна быть закончена до конца Q3. Риск увеличения бюджета
6	RAP Rollout + St. -Pete	●	●	●	●	●	Определение	Фейзунлин	Казимири	Иванова	2477	446	01.11.2006	31.12.2008	
7	Sites Telecom	●	●	●	●	●	Выполнение	Джобсон	Лилеев	Журавлев	182	8	15.10.2006	31.05.2008	Задержка завершения проекта на 5 месяцев (с 31.12) в связи с неготовностью объектов. Ведутся работы по улучшению качества и надежности каналов связи
8	DMS Subs	●	●	●	●	●	Выбор	Резников	Савельева	Проклашкин	500	24	31.09.2007	31.06.2008	Ведется разработка шаблона. С конца мая начнется внедрение в Маслах (с интеграцией SAP)

9	RNP Migration to corporate domain	●	●	●	●	●	Выполнение	Куркин	Стигеев	Матиец	0	0	01.09.2006	30.05.2008	Задержка сроков из-за реализации требований ДЗИ (дополнительный объем работ по миграции существующих приложений в корп. домен)
10	Lubes migration to corporate domain	●	●	●	●	●	Выбор	Габалда	Молдаванов	Лабудев	200	0	15.03.2008	31.12.2008	Защищен ФМ. Согласована спецификация и начата закупка оборудования
11	Construction of a Fiber-Optic link between PU Lubricants units in Ryazan	●	●	●	●	●	Выполнение	Мальцева	Молдаванов	Лозовой	22	22	01.12.2005	31.04.2008	Работы были завершены, но в результате строительных работ произошёл обрыв канала. Сейчас канал восстанавливается

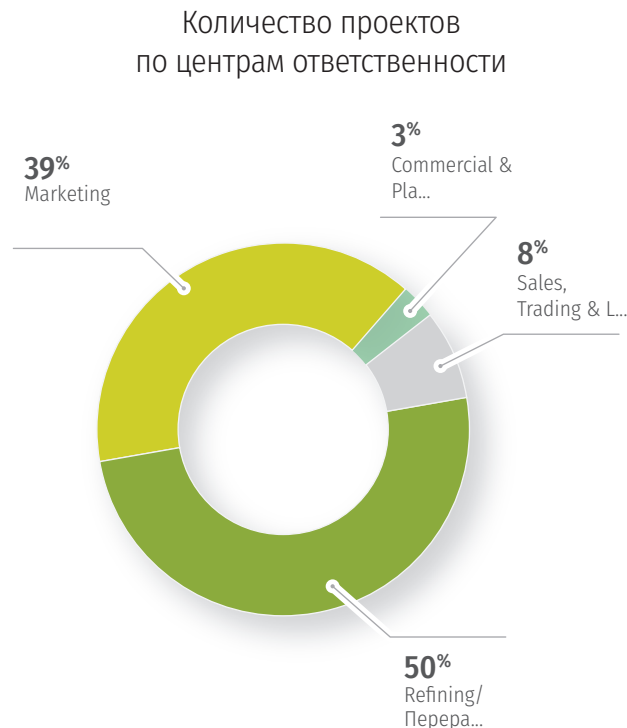
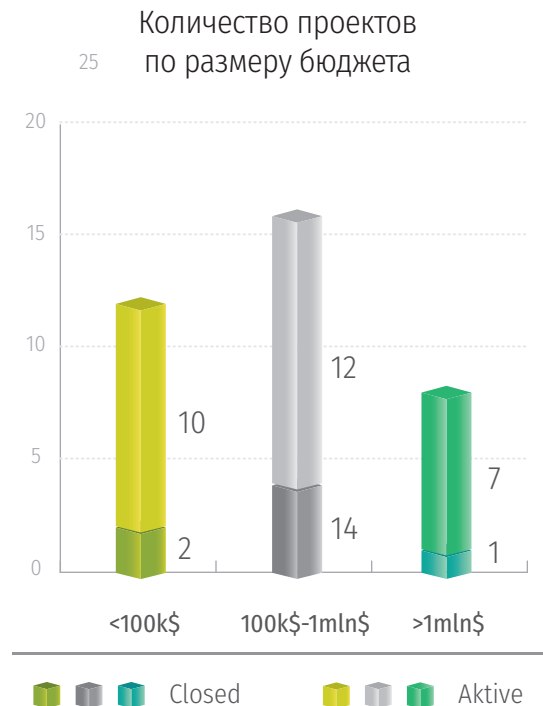
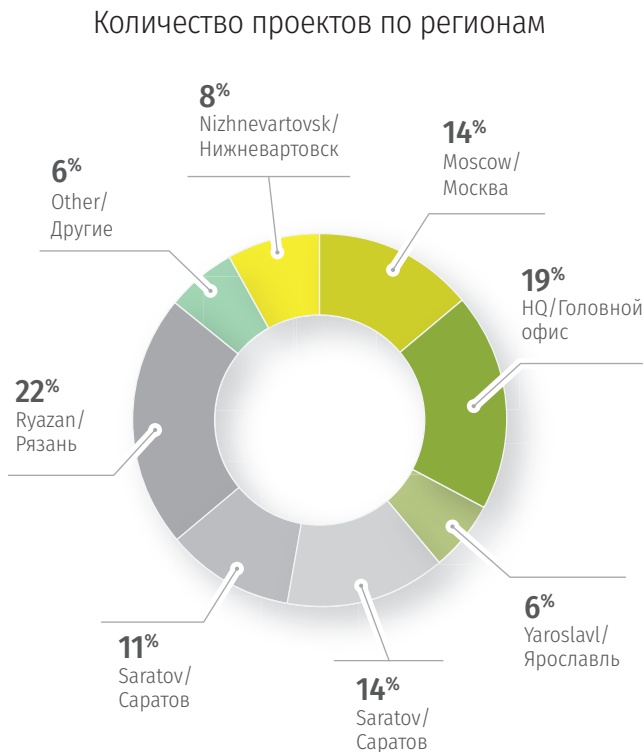
Опыт управления портфелем ИТ-проектов в ТНК-ВР

ИТ-служба компании ТНК-ВР до слияния с компанией «Роснефть» в 2013 году находилась на высоком уровне зрелости проектного управления, ее опыт до сих является весьма интересным и актуальным. В компании ежегодно реализовывалось около 250 ИТ проектов. Принятая в компании методология выполнения проектов CVP (Capital Value Process) включала в себя элементы портфельного управления (запуск проектов на ИТ Комитете, контрольные точки принятия решения по проекту — ворота). Кроме того, в течение длительного времени элементы портфельного управления применялись в рамках ежегодного бюджетирования. Для систематизации процесса в 2011 году было принято решение о внедрении портфельного управления на уровне корпоративного ИТ. «Портфельное управление нужно в том случае, когда руководство компании/функции/подразделения хочет управлять развитием своей организации», — говорит Татьяна Мохова, директор ИТ проектов ТНК-ВР. — Связь простая: развитие всегда происходит через изменения, а изменения — не что иное, как проекты. Управлять — значит с большей вероятностью достигать стратегических целей развития в ожидаемые сроки, за ожидаемые деньги и за счет наиболее эффективных действий, т.е. проектов. Если у компании есть стратегия развития с понятными целями и результатами, и есть ограничения в достижении этих целей, например, по возможным инвестициям, тогда портфельное управление практически обосновывать не нужно, оно интуитивно понятно, и все, что требуется, так это придать осознанную форму. В случае, если стратегии нет, но есть ограничения по ресурсам, то основное обоснование — это возможность эффективно распределять дефицитные ресурсы, включая внимание руководства, на наиболее важные задачи с целью более гарантировано получить результат там, где надо, а не где получится. Если нет ни того, ни другого, то обосновать портфельное управление весьма сложно. На поверхности лежит только необходимость централизованного контроля не только средств, но и результатов их расходования, и необходимость поддержки определенного уровня порядка при реализации проектов».

Приведем также цитату Анны Тиуковой, старшего менеджера проектного офиса ТНК-ВР: «Управление портфелем позволяет:

- лучше понимать реальную, а не декларируемую стратегию компании и приоритеты;

Рис. 2.2.7. Примеры аналитических отчётов по портфелю проектов.



- не планировать лишний бюджет и не сталкиваться с нехваткой бюджета;
- не планировать больше проектов, чем можно сделать при имеющихся ресурсах;
- иметь четкое обоснование приоритетов одного проекта перед другим и избегать конфликта интересов.

Это говоря формальным языком, а по факту управление портфелем — это реальное уменьшение общего хаоса, т.е. это те координаты и рычаги управления, которые в моем представлении должны уменьшить количество внезапных, «мобилизационных» проектов».

Основные определения и элементы процесса управления портфелем ИТ-проектов, а также этапы жизненного цикла портфеля и процесс формирования и регулярного обновления портфеля проектов ИТ ТНК-ВР показаны на рис. 3.2.8-3.2.10

Рис. 2.2.8. Основные определения и элементы процесса управления портфелем ИТ-проектов.

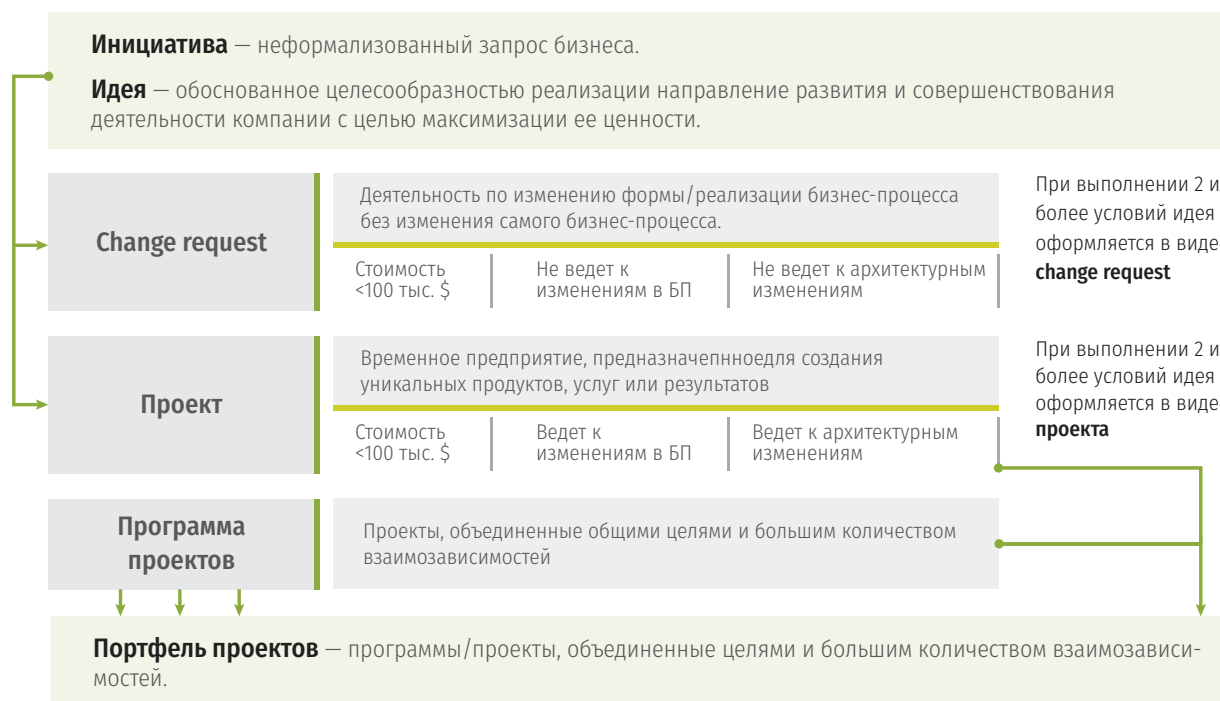


Рис. 2.2.9. Этапы жизненного цикла портфеля и единый механизм управления портфелем ИТ-проектов ИТ в ТНК-ВР



Единая методология, информационная система, отчетность, стандарты и метрики с учетом опыта бизнес-направлений

Рис. 2.2.10. Временная шкала процесса формирования и регулярного обновления портфеля ИТ-проектов в ТНК-ВР.



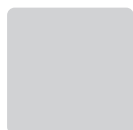
Часть 3. Управление ИТ деятельностью

Глава 2.3

Управление ИТ-процессами и услугами



Олег
Скрынник



Роман
Журавлев



Михаил Потоцкий



Владимир Ананьин

Определения

ITIL® — Набор публикаций, содержащий лучшие практики в области управления ИТ-услугами. ITIL содержит рекомендации по предоставлению качественных ИТ-услуг, процессов, функций, а также других средств, необходимых для их поддержки. Структура ITIL основана на жизненном цикле услуги, который состоит из пяти стадий (стратегия, проектирование, преобразование, эксплуатация и постоянное совершенствование). Каждой из этих стадий соответствует определённая книга ITIL. Также существуют дополнительные публикации, входящие в ITIL, содержащие специфичные рекомендации по индустриям, типам организаций, моделям работы и технологическим архитектурам.

Управление ИТ-услугами — (ITSM) IT Service Management (ITSM) Внедрение и управление

качественными ИТ-услугами, которые соответствуют потребностям бизнеса. Управление ИТ-услугами реализуется поставщиками ИТ-услуг путём использования наиболее оптимального сочетания людей, процессов и информационных технологий.

Служба Поддержки Пользователей — (Служба Service Desk) Service Desk (ITIL Service Operation) Единая точка контакта между поставщиком услуг и пользователями. Типичная служба поддержки пользователей управляет инцидентами, запросами на обслуживание, а также осуществляет коммуникации с пользователями.

Соглашение об Уровне Услуг — (SLA) Service Level Agreement (SLA) (ITIL Service Design) (ITIL Continual Service Improvement) Соглашение между поставщиком ИТ-услуг и заказчи-

ком. Соглашение об уровне услуг описывает ИТ-услугу, документирует целевые показатели уровня услуги, указывает зоны ответственности сторон – поставщика ИТ-услуг и заказчика. Одно соглашение об уровне услуг может распространяться на множество ИТ-услуг или множество Заказчиков.

База Данных Управления Конфигурациями — (CMDB) Configuration Management **Database** —

(CMDB) (ITIL Service Transition) База данных, используемая для хранения конфигурационных записей на всем протяжении их жизненного цикла. Система управления конфигурациями поддерживает одну или несколько баз данных управления конфигурациями, каждая база данных хранит атрибуты конфигурационных единиц и взаимоотношения с другими конфигурационными единицами.

Услуги как форма предоставления ценности

По мере интеграции информационных технологий в бизнес-процессы на протяжении последних двух-трёх десятков лет менялось отношение бизнеса к ИТ-службе, её роли в бизнесе и к тому, каким образом эта служба формирует ценность для заказчиков. В самых общих чертах эти изменения можно описать так:

- первоначально решения на основе информационных технологий рассматривались как ещё один класс инструментальных средств, помогающих выполнять бизнес операции;
- затем акцент был сделан на деятельность ИТ-служб, предоставляющих эти средства, появилось и стало общеупотребительным понятие «ИТ-услуга»;
- наконец, современные подходы к управлению и руководству ИТ предлагают управлять бизнес-ценностью, формируемой на базе информационных технологий.

Почему необходимо изменение акцента управления с продуктов или ресурсов (инструментов автоматизации бизнес-процессов) на услуги? Использование термина «услуга» применительно к информационным технологиям является следствием усложнения взаимоотношений между поставщиками и покупателями. Можно сказать, что в наиболее общем виде услуги — это блага, предоставляемые не в виде реальных, осязаемых вещей. Важнейшим

преимуществом для основного бизнеса компании от использования сервисного подхода к управлению информационным технологиям является возможность концентрации на основных видах деятельности, а не на управлении ИТ-ресурсами, о которых бизнес-подразделения, как правило, знают весьма немного. Существует множество определений услуги. В частности, библиотека ITIL, один из источников знаний в области ITSM, определяет термин «услуга» следующим образом:

Услуга — способ предоставления ценности заказчику через содействие им в достижении желаемых конечных результатов, которых Заказчики хотят достичь без владения специфическими затратами и рисками.

Комментируя фразу о получении результатов (в оригинале — outcomes) в определении услуги, ITIL поясняет, что речь идёт о бизнес-результатах, которые бизнес стремится получить, обладая определённой производительностью и учитывая действие на него разного рода ограничений. Как правило, по различным причинам заказчики услуг стремятся получить желаемые результаты, но не хотят брать на себя издержки и риски, связанные с их достижением. Например, бизнес подразделению необходимо ведение архива операций с клиентами. Для решения этой задачи подразделению нужны персонал, оборудование и инфраструктура, способные поддерживать контроль над архивом такого объёма. Но это

подразделение, тем не менее, не хочет брать на себя связанные с использованием хранилища риски и издержки, будь они реально существующими или лишь предполагаемыми.

Управление услугами — это множество специализированных организационных способностей для предоставления ценностей заказчикам в форме услуг.

В глоссарии ITIL способность (capability) — это возможность организации, человека, процесса, приложения, конфигурационной единицы или ИТ-услуги осуществлять деятельность. То есть, способности — это нематериальные активы организации, например, такие как менеджмент, процессы и знания.

Несколько упрощая, можно сказать, что в основе управления услугами лежит трансформация ресурсов поставщика в услуги, имеющие ценность для заказчика. Без этого организация, предоставляющая услугу, является всего-навсего набором ресурсов, которые сами по себе несут относительно низкую ценность для потребителя. Отражением тенденции роста значения услуг в экономике в целом, в области ИТ стало появление и рост ценности услуг, которые облегчают взаимодействие или обмен информацией. В результате, ресурсы в области ИТ стали рассматриваться не про-

сто как инструментальные средства для поддержки операционных процессов. Они стали базой для создания ценности, и формой предоставления этой ценности становятся именно ИТ-услуги. Смещение акцента в управлении с продуктов (инструментов автоматизации) на услуги привело к появлению и развитию нового направления в ИТ-менеджменте: ITSM, или управления ИТ-услугами. Современные тенденции в использовании информационных технологий делают управление услугами основной формой управления ИТ.

Основные принципы ITSM можно выразить в двух предложениях:

1. Формой предоставления ценности заказчикам являются услуги.
2. Формой управления услугами являются процессы.

Используя вышеприведённое определение услуги, ITIL так определяет ИТ-услугу:

ИТ-услуга — услуга, предоставляемая одному или многим заказчикам поставщиком ИТ-услуг. ИТ-услуга базируется на использовании информационных технологий и поддерживает бизнес-процессы заказчика. ИТ-услуга задействует людей, процессы, технологии.

Разумеется, при смене акцентов в управлении с продуктов на услуги и с услуг на ценность происходит изменение уровня абстракции, акцент смещается с технологий.

Надо сделать важное замечание относительно термина «ИТ-услуга». С тех пор, как библиотека ITIL второй версии неосторожно назвала ИТ-услугой «одну или более ИТ-систем, позволяющих работать бизнес-процессу», началась самая настоящая путаница. Вроде бы получается, что ИТ-услуги и ИТ системы — суть одно и то же. Однако нет, ИТ-услуга — гораздо более широкое понятие! В случае с внутренним ИТ подразделением — это всё, что нужно для работы конечного бизнес-пользователя: и программа, и АРМ, и сеть, и поддержка Service Desk... Только такой взгляд на ИТ-инфраструктуру и деятельность ИТ-персонала позволяет, к примеру, определить сквозные (end-to-end) требования к доступности ИТ-услуг, от рабочего места пользователя через всю широкую и сложную инфраструктуру до последнего сектора на жёстком диске сервера.

Прошлое, настоящее и будущее сервисного подхода к управлению

Компонентный подход

Исторически первым возник подход к управлению ИТ, который можно условно назвать «компонентным». Его суть – ИТ-отдел предоставляет организации средства автоматизации, программно-аппаратные комплексы, одним словом — **компоненты для поддержки бизнес-операций**.

В этой модели в организации формируется отдельное подразделение, называемое, к примеру, департаментом информационных технологий. Во главе подразделения назначается руководитель — как правило, из числа толковых и опытных ИТ-специалистов. Работа подразделения строится по принципу «получили задание – начинаем работать, а в остальное время следим за техникой».

Задания формируются как функциональные требования к системам автоматизации: программа должна уметь выполнять определённые действия в ответ на определённые управляющие воздействия пользователя. Такого рода требования вполне реально получить от бизнес-заказчиков, которые зачастую так и называются — «функциональные заказчики». Требования к надёжности, доступности, обеспечению непрерывности и проч., как правило, чётко и системно не формируются, либо разрабатываются не для всех программно-аппаратных средств. И уж совсем редко они формируются для ИТ-инфраструктуры целиком, так как это дорого, да и необходимость не очевидна.

Тем не менее, бизнесу разбираться во всём этом не очень хочется, так как есть более важные задачи, поэтому отдел ИТ работает так, как работает, принося пользу в меру своих сил. Честно говоря, даже с согласованием и утверждением ИТ-бюджета больших сложностей обычно нет – закупать можно всё, что захочется, всё равно ведь бизнес в этих серверах, каналах, контрактах не понимает...

Важной особенностью **компонентного** подхода является следующее наблюдение: для того, чтобы ИТ директору сохранить своё рабочее место, ему необходимо уделять самое пристальное внимание двум основным направлениям деятельности:

- обеспечивать стабильность среды эксплуатации – имеющиеся ИТ-системы не должны часто «падать», и если уж «упадут», то должны быстро восстанавливаться;
- постоянно предлагать основному бизнесу новый функционал ИТ-систем, чтобы тот ощущал пользу от ИТ.

Сервисный подход

Альтернативой описанному выше подходу, исторически используемому в большинстве современных организаций в России, является сервисный подход. Важно отметить, что, переходя к применению сервисного подхода, ИТ-подразделение совершает качественное изменение. В таком случае понятие «услуга» вообще, и «ИТ-услуга» в частности, может служить тем самым «мостиком» взаимопонимания. Действительно, любая организация постоянно приобретает какие-либо услуги — в большинстве случаев гораздо активнее, чем товары или материалы. Следствие применения сервисного подхода для ИТ-руководителя — более чёткое понимание собственной зоны ответственности. В пример можно привести высказывание одного из ИТ-директоров, работающего в крупном российском банке — «сервисный подход позволил мне ответить на вопрос «за что меня могут уволить?», а затем стало понятно, что следует делать, чтобы этого избежать».

Перспектива

Сервисный подход эволюционирует. В 2000 году, когда появилась первая книга второй версии ITIL, идея о том, что основной целью рабо-

ты ИТ-отдела является предоставление услуг, а не управление инфраструктурой, выглядела новой и смелой. Спустя семь лет третья версия библиотеки уже рассматривает этот тезис как естественный и очевидный и даже объявляет его недостаточным: «заказчик не заинтересован в услугах, они — лишь форма предоставления ценности». Под ценностью подразумевается помощь в решении задач заказчика — повышение производительности

бизнес-процессов и снижение влияния ограничений.

Таким образом, услуги – не самоцель. Возможно, тем ИТ организациям, которые всё ещё строят свою работу вокруг управления инфраструктурой, такой подход поможет правильно расставить акценты, планируя своё развитие.

Так, COBIT не рассматривает предоставление услуг в качестве цели деятельности ИТ.

ITIL как практика управления услугами

Библиотека ITIL (IT Infrastructure Library) является частью обширных знаний, на которые опирается управление услугами. Этот банк знаний создан по инициативе и контролируется правительством Великобритании. Хотя часть функций по управлению, в частности — развитие сертификации специалистов, организаций и программного обеспечения, отданы в аутсорсинг коммерческой организации AXELOS.

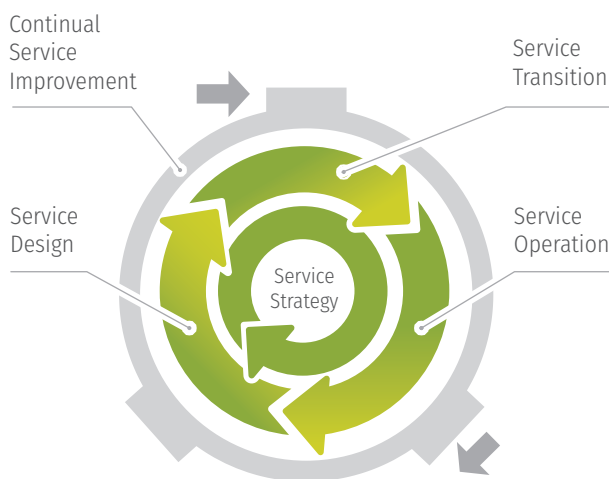
Во многом на базе ITIL был разработан и в 2002 году утверждён британский стандарт в области управления ИТ услугами — BS 15000. В декабре 2005 года он, почти не претерпев изменений, стал основой для международного стандарта ISO/IEC 20000:2005, с тех пор он обновлялся и дополнялся. ISO/IEC 20000 предоставляет формализованный универсальный стандарт для организаций, которым необходим аудит и сертификация своих управленческих способностей.

Библиотека ITIL включает в себя следующие **компоненты**:

- **центральные книги ITIL**: руководство по применению передового опыта для всех организаций, предоставляющих услуги бизнесу;
- **дополнительные публикации ITIL**, содержащие различные полезные материалы: руководства для конкретных отраслей, типов организаций, эксплуатационных моделей и технологических архитектур.

Центральными книгами ITIL являются пять публикаций, которые совпадают с пятью фазами жизненного цикла сервисов (Рис. 2.3.1).

Рис. 2.3.1. Пять фаз жизненного цикла сервиса.



Заметим, что в области управления ИТ-услугами сформировалось и несколько проприетарных сводов знаний – адаптаций общих практик под инструменты, которые предлагает конкретный разработчик – Microsoft Operation Framework, HP ITSM Reference Model. Среди источников знаний и рекомендаций в области управления ИТ-услугами стоит также упомянуть FITS (Framework for Information

Technology Support), USMBOK (Universal Service Management Body Стратегия услуг.

1. Проектирование услуг.

2. Преобразование услуг.

3. Эксплуатация услуг.

4. Постоянное совершенствование услуг.

Организация управления ИТ-услугами

Эксплуатация, поддержка и сопровождение ИТ-услуг — это постоянная, непрерывающаяся деятельность, и основной формой управления этой деятельностью являются **процессы**. Важно заметить, что при предоставлении услуг ответственность поставщика не прекращается в момент передачи решения заказчику. Предоставление и потребление услуг — это **совместная деятельность** обеих сторон, предполагающая постоянный контроль качества услуг и удовлетворённости потребителей.

Поскольку сервисный подход предполагает постоянное взаимодействие поставщика и заказчика услуг, постоянное обеспечение согласованного качества предоставляемых услуг, естественным направлением развития стало формирование системы менеджмента качества (СМК) для ИТ-услуг. В свою очередь, развитие специализированной СМК предполагает использование универсальных принципов управления качеством в приложении к специфике информационных технологий. Поэтому неудивительно, что основной формой организации деятельности по управлению качеством ИТ-услуг во всех подходах стало процессное управление, и каждый подход описывает эту деятельность как более или менее сложную систему процессов.

Общая картина

Общая схема организации деятельности по управлению информационными технологиями следующая. Поставщик ИТ услуг обладает специфическими активами — ресурсами и способностями, определяющими его ИТ-специализацию. Активы сгруппированы по назначению, образуя функции. Функции пре-

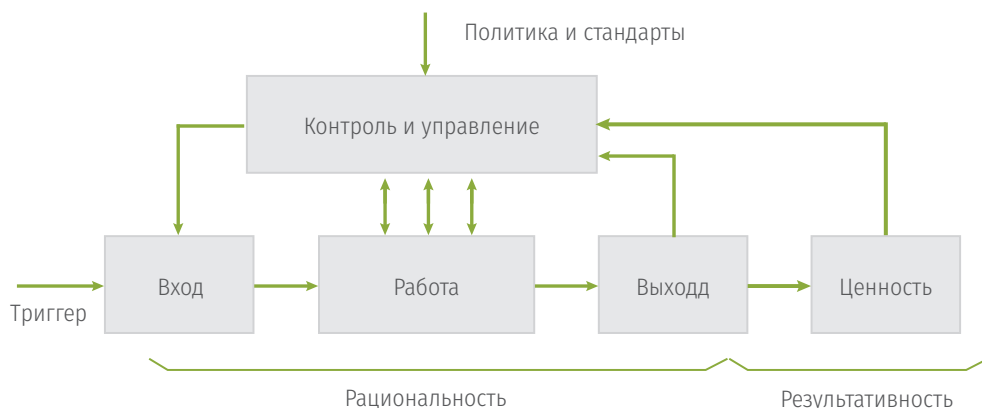
доставляют ресурсы и способности для осуществления деятельности. В большинстве случаев функциональная группировка активов поддерживается организационной структурой. Деятельность поставщика инициируется поступающими требованиями заказчиков (бизнеса). Процессы управления ИТ-услугами обеспечивают предоставление и поддержку услуг на согласованном с заказчиком уровне качества. Результатом этой деятельности являются услуги, следствием — бизнес ценность.

В случае, если услуги не соответствуют текущим или новым требованиям заказчиков, поставщик инициирует изменения в инфраструктуре, компетенциях или организации работ — иначе говоря, в составе своих активов. Эти изменения организуются как проекты. В большинстве случаев продукты, сформированные в результате проектов, не предоставляются заказчику непосредственно, но входят в состав изменённых ИТ-услуг. Таким образом, функции — это основная форма организации активов ИТ-поставщика, а процессы и проекты — основные формы организации деятельности с использованием этих активов.

Главное о процессах

Процесс — это комплекс взаимосвязанных видов деятельности, выполняемых совместно или последовательно, направленный на повторяемое получение определённого измеримого результата. Процесс выстраивает деятельность в логичную, результативную и рациональную последовательность, обеспечивающую достижение запланированного результата. Можно сказать, что процесс — это последовательность

Рис. 2.3.2. Процесс ITOCO.



действий, формирующих результат (output, а затем и outcome), на основе входящей информации (input). Действия в процессе можно разделить на производственные, то есть формирующие результат (throughput activities) и контрольные (control activities). На введённой здесь терминологии основана простейшая модель процесса — ITOCO (Input — Throughput — Output — Control — Outcome) (Рис. 2.3.2):

Процесс инициируется каким-либо триггером. Полученный в результате выполнения деятельности немедленный результат обозначается термином output, в то время как долгосрочный эффект, к которому ведёт выполнение процесса, назван outcome. Контрольные действия направлены на оценку соответствия входов, результатов и производственных действий процессу действующим на него политикам и стандартам, а также инициацию необходимых корректировок при выявлении несоответствий. Контрольные действия обеспечивают выполнение производственных действий своевременно и в правильном порядке.

Оценка результатов (output, а лучше — outcome) позволяет сделать выводы о результативности процесса. Если результаты соответствуют predetermined критериям качества, процесс считается результативным. Важно определить такие критерии на этапе проектирования про-

цесса. Управление процессами подразумевает также, что результаты получаются максимально рациональным образом. Для оценки и обеспечения рациональности также используются контрольные действия и определяемые в рамках планирования процесса критерии и стандарты.

Процедуры и рабочие инструкции

Процессы отвечают на вопрос «что мы делаем?». Для ответов на вопросы «кто, когда и как выполняет работу?» и «как при этом используются инструменты?» создаются процедуры и рабочие инструкции.

Процедура — это определённый способ выполнения деятельности.

Рабочая инструкция — это детальное описание выполнения одного или более действий в составе процедуры, включающее в себя порядок использования технологий и ресурсов.

Эти три уровня детализации соответствуют известной формуле, с которой мы познакомились в проектном управлении (см. главу 2.1 «Управление ИТ-проектами» и главу 2.2 «Управление портфелем проектов»):

Люди ⇔ Процессы ⇔ Технологии

Главное о функциях

Функции — это объединения людей (в том числе и виртуальные) и используемый ими инструментарий, включая знания, методы, средства автоматизации и т.п., специализированные для выполнения определённой деятельности и отвечающие за получение определённых результатов.

Важно отметить, что функции не обязательно совпадают с организационными единицами и могут быть представлены виртуальными командами. В организации функции координируются через различные инструменты управления, в том числе и посредством общих процессов. Функции склонны концентрироваться на внутреннем развитии и достижении собственных целей вне контекста организации, что может привести к их изоляции. Поэтому процессная модель управления помогает организовать кросс-функциональное взаимодействие и контроль. Процессы способствуют повышению эффективности внутри функций и при их взаимодействии.

Существуют различные классификации функций, чаще всего основанные на принципах объединения ресурсов. Так, можно выделить **четыре основных группы функций**:

- **Инфраструктурные** — объединения людей, которые формируются для управления отдельными составляющими информационной инфраструктуры предприятия (например, отделы управления сетями, приложениями, базами данных...).
- **Сервисные** — ориентированные на управление отдельными аспектами качества услуг (отделы или другие группы управ-

ления доступностью, безопасностью, непрерывностью и т.д.).

- **Процессные** — группы людей, объединяющие ресурсы для выполнения определённых видов деятельности (например, команды управления изменениями, или безопасностью, или проектами, отдел разработки ПО или отдел тестирования, группа управления качеством).
- **Организационные** — группы людей, которые формируются для того, чтобы поддерживать структуру организации и могут объединять ресурсы по территориальному («Европа и СНГ») или структурному («штаб-квартира») принципу.

Разумеется, возможны как любые сочетания, так и другие принципы группировки. В любом случае, самое важное — то, что процессы только определяют необходимые действия и порядок их исполнения, в то время как функции обладают всеми необходимыми для этого ресурсами и способностями.

Свод знаний по управлению ИТ-услугами уделяет функциям ITSM ничтожно мало внимания. Так, в предыдущей, второй версии библиотеки ITIL была описана всего одна функция — Service Desk. В ITIL v3 число сущностей, описанных как функции, достигает четырёх. Помимо Service Desk, функциями названы Operations Management (Управление эксплуатацией), Technical Management (Управление технической поддержкой) и Application Management (Управление приложениями). Практически число описанных функций больше, поскольку многие из них рассматриваются в ITIL под заголовком «Процессы».

Основные компоненты системы управления ИТ услугами

Рассмотрим основные процессы и функции системы управления ИТ услугами, согласно ITIL.

Управление уровнем услуг (Service Level Management)

Цели этого процесса — **проектирование новых** (изменяемых) **услуг** в ответ на поступающие требования бизнеса и управление качеством услуг, уже предоставляемых заказчиком. Проектирование услуг предполагает согласование требований к их качеству и определение параметров работы ИТ-инфраструктуры, которые обеспечат выполнение этих требований. Работа по управлению предоставлением услуг включает в себя **постоянный контроль** соответствия фактического качества услуг согласованным требованиям. Чтобы проектировать услуги и управлять их предоставлением, необходимо согласовать параметры, которые позволят определять качество ИТ-услуг и оценивать его на соответствие достигнутому с заказчиками услуг договорённостям. Перечень таких параметров мало отличается от услуги к услуге и от заказчика к заказчику, меняется только их относительная значимость. Можно выделить несколько **ключевых параметров качества услуг**, в большинстве случаев при-

знаваемых основными и определяющими: **функциональность, производительность, доступность и цена**. Есть и другие параметры, управление большинством из которых осуществляется в рамках, соответствующих специализированных сервисных функций.

Функциональность в сочетании с **производительностью** — это то, ради чего услуга потребляется. Функциональность услуги определяет, какие задачи заказчика могут быть решены с помощью этой услуги. Производительность определяет количественные характеристики функциональности (скорость обработки определённого количества информации).

Доступность — это характеристика, позволяющая определить временные и пространственные границы предоставления услуги, то есть ответить на вопрос «где и когда предоставляется услуга?». И наконец, цена — это основной критерий для оценки полезности предоставляемой услуги или принятия решения о её проектировании и передаче в эксплуатацию.

На Рис. 2.3.3 приведена общая схема деятельности по управлению уровнем услуг.

Рис. 2.3.3. Общая схема деятельности по управлению уровнем услуг.



Она начинается с получения требований заказчика, их **обсуждения и формализации** (1). Возможно, первые формулировки будут не очень точны и недостаточно структурированы, и на этом этапе важно придать им чёткую измеримую форму, не теряя при этом связи с ожиданиями заказчика. Такая работа может стать предметом деятельности отдельного процес-

са — управления отношениями с заказчиками, или выполняться как часть процесса управления уровнем услуг.

Разумеется, не все требования заказчиков могут быть выполнены в срок и в полном объёме за те деньги, которые заказчик готов инвестировать в решение соответствующей бизнес-задачи. Поэтому важным шагом, следующим за определением требований, является **оценка их осуществимости** (2). План по выполнению требований с учётом **проведённой оценки** (3) в большинстве случаев требует **согласования с заказчиком** (4).

В ходе этого согласования он может быть одобрен, что часто подразумевает выделение средств на его реализацию. Возможно, впрочем, что план придётся дорабатывать и существенно менять. Более того, часто бывает так, что заказчик корректирует свои требования с учётом технических, финансовых или иных ограничений, выявленных в ходе оценки осуществимости первого плана.

Описанный цикл согласования и планирования услуги может повторяться **многократно**. В итоге либо утверждается видение сторонами новой или изменяемой услуги, а также сроков и методов её проектирования, запуска в эксплуатацию и собственно эксплуатации, либо заказчик отказывается от сформулированных требований или от услуг данного поставщика. Если поставщику удалось согласовать с заказчиком план внедрения для новой или изменяемой услуги, то этот план, включающий в себя спецификацию услуги, а также требования и ограничения по её построению, передаче в эксплуатацию и последующим эксплуатации и поддержке, является основным входящим документом для процессов изменения услуг, а также их эксплуатации. Эти процессы обеспечивают реализацию согласованного плана и сбор данных по основным согласованным параметрам качества новой или изменяемой услуги.

Данные мониторинга анализируются для

оценки соответствия фактического качества услуги согласованным на **этапе проектирования целевым значениям** (5.1). Выявленные отклонения становятся основой для **планирования корректировок** (6). Кроме устранения отклонений от целей в области качества, корректировки могут планироваться для того, чтобы обеспечить соответствие качества услуги ожидаемым новым требованиям — например, в случае, когда прогнозируется рост нагрузки на инфраструктуру в связи с увеличением числа потребителей. Разумеется, работа по оценке фактического качества услуги и планированию улучшений не может проводиться без участия представителей заказчика. Они не только получают **отчётность о качестве услуг** (5.2), но и активно участвуют в **планировании улучшений** — в особенности тех, что направлены не на исправление недостатков, а на повышение «планки качества». Анализ информации о фактическом качестве услуг и управление улучшением услуг — основные составляющие деятельности по управлению уровнем качества предоставляемых услуг.

Отметим, что важным инструментом управления предоставлением услуг является **каталог услуг**. Поставщик ИТ-услуг обычно предоставляет какое-то множество услуг какому-то множеству заказчиков. Для этого он обычно выступает и сам в роли заказчика услуг — как внутри организации, так и в отношениях с внешними контрагентами.

Все услуги, контролируемые ИТ-службой, описываются в каталоге услуг. Та часть каталога, которая содержит информацию о предоставляемых заказчикам услугах, обычно называется бизнес-каталогом. Часть, в которой описаны потребляемые ИТ-службой поддерживающие услуги — внешние и внутренние — часто называется **техническим каталогом**. На практике термин «каталог услуг» часто используется для обозначения бизнес-каталога, то есть перечня и описания всех предоставляемых услуг.

Управление поставщиками (Supplier Management)

Процесс выстраивания отношений с поставщиками – один из ключевых процессов в деятельности СІО, поэтому вопросу Управления отношениями, в том числе и с поставщиками, в Учебнике отведена отдельная глава 2.5 «Управление отношениями». Поэтому здесь рассмотрим только суть данного процесса.

Современные ИТ-организации все больше зависят в своей работе от поставщиков услуг, как правило — **инфраструктурных**. Многие аналитики прогнозируют стремительное сокращение числа и значения собственных активов, используемых для предоставления ИТ-услуг. Увеличение числа и повышение сложности связей с внешними услугами требует выделенного контроля, что часто требует специализированного процесса. Процесс управления поставщиками отвечает за

выбор поставщиков, управление договорами, контроль качества предоставляемых услуг, поддержание соответствия предоставляемых услуг текущим и будущим требованиям ИТ-услуг и бизнес процессов.

В рамках этого процесса осуществляется взаимодействие с юридическими и коммерческими функциями организации, что обеспечивает выполнение внешних требований, рациональность отношений с поставщиками и финансовый контроль.

В организациях, где связи с поставщиками не играют существенной роли при оказании ИТ-услуг пользователям, процесс управления поставщиками может быть реализован в рамках управления уровнем услуг.

Управление изменениями (Change Management) и управление релизами (Release Management)

Рис. 2.3.4. Общая схема деятельности по управлению изменениями.



Процесс управления изменениями осуществляет общую координацию всех планируемых и проводимых изменений.

Изменение — это добавление, модификация или удаление чего-либо, способного оказать влияние на ИТ-услуги.

Релиз — набор аппаратного, программного обеспечения, документации и других компонентов, необходимых для внедрения одного или нескольких утверждённых изменений в ИТ-услугах.

Результаты построения и тестирования согласуются с комитетом по изменениям или назначенными комитетом лицами, и по результатам согласования принимается решение о передаче результатов построения в эксплуатацию. После завершения развёртывания и проведения оценки начального этапа эксплуатации принимается решение о признании изменения успешным и закрытии.

Важно отметить, что в ходе управления релизами обеспечивается не только вза-

имодействие с потребителями обновлённых решений и службой поддержки и эксплуатации, но и с разработчиками и поставщиками компонентов для этих решений. Таким образом, процессы управления изменениями и релизами формируют интерфейс между проектной и операционной составляющими ИТ деятельности.

Контроль операционного управления ИТ (IT Operations control)

Необходимым условием стабильной работы компонентов ИТ-инфраструктуры, а, следовательно, и стабильного качества ИТ услуг, является контроль состояния инфраструктуры и согласованных показателей качества ИТ-услуг, а также обнаружение, анализ и обработка событий.

Событие — это изменение состояния, которое имеет значение для управления конфигурационной единицей или ИТ-услугой.

Контроль предоставления ИТ-услуг осуществляется **на трёх уровнях**: инфраструктурном, сервисном и уровне пользователей. Работу компонентов **ИТ-инфраструктуры** в той или иной степени контролируют почти в се инфраструктурные функции ИТ, иногда более детально, иногда — менее. Для того, чтобы связать данные мониторинга инфраструктуры с показателями качества ИТ-услуг, используются так называемые модели здоровья услуг, позволяющие трактовать информацию о компонентах

в контексте выполнения обязательств по качеству услуг.

Но даже если установленным нормам соответствуют формальные показатели как компонентов ИТ-инфраструктуры, так и услуг соответствуют установленным нормам, поставщик услуг должен собирать информацию об **удовлетворённости пользователей и заказчиков услуг**. Расхождения между оценкой качества услуг, сделанной на основании мониторинга инфраструктуры, и субъективной оценкой этого качества потребителями могут служить поводом для изменения услуг, корректировки соглашений о качестве услуг или подхода к мониторингу (изменения точек контроля).

Служба поддержки пользователей (Help Desk, Service Desk)

Для того, чтобы обеспечить результативность и рациональность взаимодействия поставщика ИТ-услуг с конечными пользователями, создаётся специализированная функция, большинство источников называет её **Service Desk** или **Help Desk**. Назначение Service Desk — обеспечение коммуникаций между поставщиком ИТ-услуг и конечными пользователями. Для этого Service Desk использует средства связи, маршрутизации обращений, оповещение пользователей. Специалисты Service Desk должны обладать развитыми навыками коммуникации, хорошо ориентироваться в предоставляемых услугах, технической и организацион-

Обеспечить для пользователей удобные и понятные процедуры взаимодействия с ИТ невозможно без создания внутри самого ИТ-подразделения чётких и эффективных процедур по работе с обращениями пользователей. Наиболее эффективным для пользователей ИТ-услуг является такое взаимодействие со службой технической поддержки, при котором доступны удобные способы подачи обращений, а информация о предоставляемых пользователям услугах доступна, понятна и прозрачна пользователю... Пользователи понимают: какие ИТ-услуги они могут получить, их обращения решаются качественно и в срок, а само ИТ перестаёт быть для пользователей «черным ящиком» и обеспечивает обратную связь, предоставляя своевременное информирование о статусе обращений.

Георгий Ованесян

ной структуре поставщика, знать зависимости от субподрядчиков. Эффективно работающая функция **Service Desk** — важный интерфейс, обеспечивающий коммуникации с пользователями в рамках таких процессов, как управление инцидентами, управление изменениями, управление релизами, а также во всех случаях, когда требуется выполнить оповещение пользователей или провести опрос по какому-либо вопросу. В приведённой выше классификации функция Service Desk может быть отнесена как к организационным, так и к сервисным функциям.

Управление запросами на обслуживание (Request Fulfilment)

Часть работ по эксплуатации — обработка запросов на обслуживание.

Запросы на обслуживание — это обращения пользователей, инициирующие выполнение сотрудниками ИТ-службы заранее согласованных регламентированных операций, являющихся частью нормальной работы ИТ-услуг. Эти запросы можно условно разделить на две группы:

- запросы первой группы инициируют выполнение операций по обслуживанию инфраструктуры, с которой работает пользователь (типичные примеры — замена картриджа в принтере, обновление лицензии);
- запросы второй группы инициируют выполнение операций, формирующих дополнительную ценность для заказчика, например, формирование нестандартного отчёта, выполнение других операций с данными, недоступных по каким-то причинам пользователю, но необходимых для решения бизнес задач.

Запросы первой группы часто являются следствием недостаточных возможностей мониторинга инфраструктуры самой ИТ-службой, связанных с техническими или финансовы-

ми ограничениями. Некоторые запросы на обслуживание могут инициировать проведение изменений в инфраструктуре, чаще всего — стандартных, типовых (например, организацию нового рабочего места). Запросы второй группы могут быть одним из факторов формирования различных уровней предоставления услуги в каталоге поставщика. Широкий спектр таких запросов может стимулировать заказчика выбрать именно высокий уровень из нескольких возможных.

Управление инцидентами (Incident Management)

Инцидент — событие в инфраструктуре ИТ, оказывающее негативное влияние на качество предоставляемых услуг. Этот же термин используют для обозначения проявлений негативного влияния таких событий на уровне потребителей.

Цель процесса управления инцидентами — скорейшее восстановление согласованного качества предоставляемых услуг в случаях, когда это качество снижается или услуги незапланированно прерываются.

Для достижения этой цели в рамках процесса решаются следующие задачи:

1. Накопление и организация повторного использования знаний о нарушениях в работе услуг, способах их диагностики и восстановления нормальной работы — задача, выполнение которой повышает рациональность процесса. При наличии базы знаний поиск решений инцидентов проходит быстрее и может выполняться специалистами службы поддержки без привлечения экспертов и поставщиков. Работа по накоплению знаний об инцидентах способствует и результативности процесса, ведь инциденты, о которых накоплена такая информация, более качественно и быстро диагностируются и решаются.

2. Своевременное информирование пользователей о временной недоступности ИТ-у-

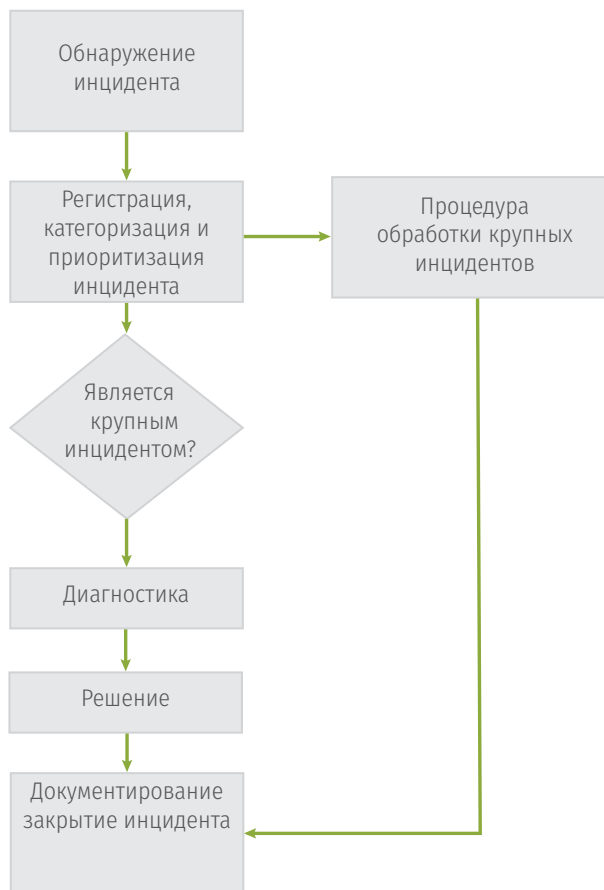
луг помогает снизить негативное влияние инцидентов на бизнес-процессы во время работ по расследованию и устранению этих инцидентов. Получая от ИТ-службы информацию о случившихся инцидентах и плановом времени их устранения, пользователи могут более рационально планировать свою работу. На Рис. 2.3.5 приведена общая схема деятельности по управлению инцидентами.

Приоритизация — один из важнейших механизмов управления инцидентами, его работа поддерживает уверенность бизнес-заказчиков в том, что ИТ-служба распределяет свои ограниченные ресурсы с учётом интересов бизнеса, а каждый отдельный инцидент будет решён в сроки, позволяющие свести к минимуму негативное влияние на бизнес-процессы. Аналогичный механизм применяется и в других процессах сервис-менеджмента в случаях, когда между задачами возникает ресурсный конфликт.

Управление конфигурациями (Configuration Management)

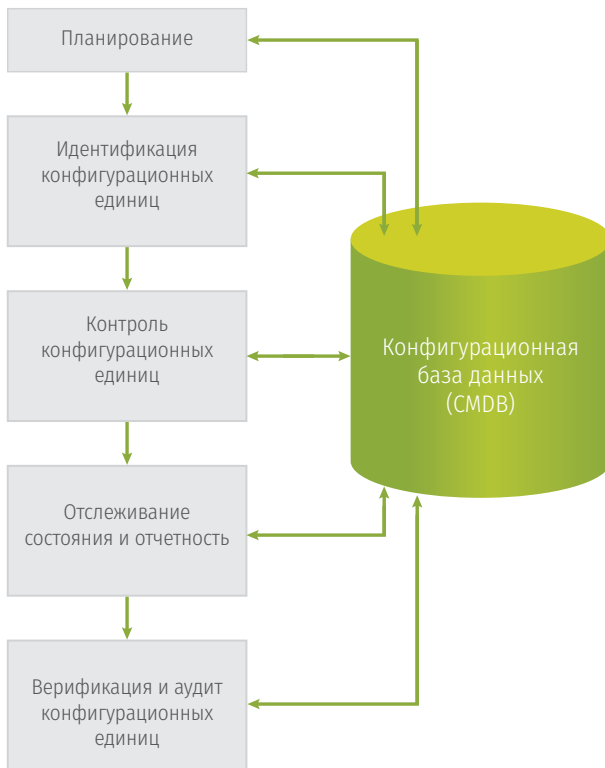
Цель процесса управления конфигурациями — **хранить и предоставлять** информацию о составе и связях компонентов инфраструктуры ИТ, задействованных в предоставлении услуг. При проведении изменений, чтобы обеспечить корректность и целостность информации об изменяемых компонентах инфраструктуры, а также возможность восстановления в случае нештатного прохождения изменений, процесс управления конфигурациями фиксирует начальное, итоговое и промежуточные состояния инфраструктуры. Этот же процесс обеспечивает управление изменениями и релизами информацией о влиянии проводимых изменений на окружающую ИТ и бизнес-инфраструктуру. Эта информация помогает планировать изменения, формировать представительство в комитете по изменениям, возвращаться к предыдущим состояниям, если что-то идёт не так.

Рис. 2.3.5. Общая схема деятельности по управлению инцидентами.



Процесс управления конфигурациями предоставляет заинтересованным лицам информацию не только о статусе и связях компонентов инфраструктуры между собой, но также и о том, как эти компоненты участвуют в предоставлении ИТ услуг. Процессы поддержки используют данные о конфигурации для оценки влияния событий и расследования обстоятельств и причин этих событий. Разумеется, такая информация незаменима и при планировании и проектировании услуг. Знание зависимости услуг от характеристик отдельных компонентов инфраструктуры позволяет лучше планировать развитие услуг и понимать ограничения и возможности инфраструктуры.

Рис. 2.3.6. Общая схема деятельности по управлению конфигурациями.



Важнейшим понятием управления конфигурациями является конфигурационная единица. Конфигурационные единицы — это все значимые для предоставляемых услуг компоненты инфраструктуры ИТ.

Планируя процесс управления конфигурациями, необходимо определить, какие компоненты будут находиться под контролем процесса, на каком уровне детализации будут идентифицироваться, и какие их атрибуты и свойства будут учитываться в системе управления конфигурациями. На Рис. 2.3.6 приведена общая схема деятельности по управлению конфигурациями.

Управление проблемами (Problem Management)

Проблема — ошибка в инфраструктуре ИТ, способная стать или являющаяся причиной инцидентов. Свойство отдельных элемен-

тов инфраструктуры или их взаимодействия, (потенциально) вредное для предоставляемых услуг.

Процесс управления инцидентами сосредоточен на скорости восстановления нормальной работы бизнес-процессов. Поэтому в рамках этого процесса не выполняется поиск причин инцидентов. Тем не менее, очевидно, что без такого анализа невозможно устранить корневые причины нарушений в работе инфраструктуры, и инциденты будут возникать вновь. Выявление корневых причин инцидентов и инициация их исправления для предотвращения повторения инцидентов — это основная деятельность процесса управления проблемами.

Рис. 2.3.7. Общая схема деятельности по управлению проблемами.



Однако возможна ситуация, когда полное устранение выявленных ошибок в и инфраструктуре невозможно или нерационально. В таких случаях управление проблемами может инициировать изменения в инфраструктуре, которые снизят негативное влияние этих ошибок на бизнес процессы до приемлемого уровня без полного устранения ошибки.

В любом случае процесс управления проблемами в первую очередь предоставляет в процесс управления инцидентами информацию о том, каким образом следует устранять инциденты, являющиеся проявлениями найденной ошибки. Эта информация может использоваться при решении инцидентов до тех пор, пока ошибка не будет устранена или не утратит влияние.

На Рис. 2.3.7 приведена общая схема деятельности по управлению проблемами.

Поиск причин инцидентов, осуществляемый как реакция на их возникновение, характеризует «реактивное управление проблемами». Его основная задача — предотвращение повторения уже случившихся инцидентов. При наличии необходимых для этого ресурсов осуществляется также так называемое «проактивное управление проблемами», в ходе которого ошибки в инфраструктуре выявляются на основе другой информации: данных мониторинга, информации поставщиков и производителей ИТ-систем, опыта других потребителей и так далее. Устранение выявленных таким образом ошибок помогает предотвратить не повторения, а возникновение соответствующих инцидентов.

Границы применения модели управления ИТ услугами (ITSM)

ITSM шествует по миру ИТ уже почти 30 лет, вышли несколько версий сводов в этой области, сформировалась целая сеть профессиональных сообществ, армия сертифицированных специалистов, растёт осведомлённость бизнеса о преимуществах, принципах и практиках ITSM. И, тем не менее, масштаб реального использования бизнесом модели управления ИТ-услугами оказывается не столь велик.

Однако различные причины сдерживают широкое распространение ITSM. Наверное, было бы сильным упрощением это связывать с недостаточной квалификацией консультантов, слабой маркетинговой активностью профессионального сообщества или тотальной незрелостью бизнеса. Похоже, у этого явления есть более глубокие причины. Не понимая границ адекватного применения той или иной методологии, технологии или инструмента, мы не поймём, как можно их использовать эффективно. Универсальны только неработающие методологии.

ITSM – модель

«покупатель — продавец»

ITSM — это такая модель управления, которая позволяет построить внутреннюю организацию ИТ-службы и её взаимодействие с бизнесом как единое целое. В этой модели взаимодействие ИТ-службы и бизнеса построено по принципу оказания услуг, а организация деятельности ИТ-службы — как система взаимосвязанных процессов управления услугами. Принцип оказания услуг означает, что исполнитель (ИТ служба) отвечает не за работоспособность ИТ-ресурсов, а за получаемый заказчиком (бизнес-подразделениями) результат. И ИТ-услуга — это спецификация этого результата, которая включает его описание в терминах бизнеса, минимум 3 разделов:

- функции информационных систем, используемые бизнесом для решения своих задач, например, электронная почта, печать или даже call-центр;

- условия и способ нормального доступа к этим функциям, например, график предоставления и способ доступа пользователей;
- уровни предоставления и поддержки, например, диапазон уровней производительности, доступности или скорости разрешения инцидентов.

Чтобы ИТ-служба могла обеспечивать предоставление и поддержку ИТ-услуг в требуемой спецификации, её работа должна быть организована по определённым процессам, например, ITSM.

Модель ITSM имеет важную особенность. Для того, чтобы ИТ служба могла гарантировать предоставление и поддержку ИТ-услуги, она должна иметь возможность самостоятельно решать — как, когда и какие ИТ-ресурсы она будет использовать, чтобы обеспечить требования, заданные спецификацией этой ИТ-услуги. В этом случае ИТ-сервис становится границей раздела прав между бизнесом и ИТ службой: бизнес отдаёт ИТ-службе право управления ИТ ресурсами, а себе оставляет только право пользования и владения ими. Например, бизнесу нужна корпоративная электронная почта требуемого уровня, но его не должно волновать, как организована её работа и обслуживание ресурсов, поддерживающих систему.

Передача в рамках модели ITSM части полномочий ИТ-службе приводит к тому, что бизнесу теперь необходимо с ней договариваться, а не «строить» её, как прежде. Модель ITSM подталкивает отношения между бизнесом и ИТ-службой к форме рыночных отношений (отношений типа «покупатель — продавец»). Понятно, что в случае с полностью зависимой внутренней ИТ-службой (отношений типа «начальник — подчинённый») форма войдёт в противоречие с реальным содержанием отношений. И при таких отношениях из всех процессов ITSM в ИТ-службе смогут прижиться только некоторые процессы и Service

Desk. В этом случае ИТ-служба по-прежнему продолжает выполнять свои функциональные обязанности и поручения бизнеса, бизнес с ней не торгуется — он её «строит».

По-настоящему форма и содержание ITSM модели совпадают только в случае вывода ИТ-службы на аутсорсинг (см. главу 3.4 «ИТ-аутсорсинг») с передачей на её баланс ИТ-активов. Между этими крайностями существует огромное разнообразие промежуточных вариантов инсорсинга и аутсорсинга.

Существует хороший признак того, когда бизнес и ИТ-служба начинают реально переходить к отношениям «покупатель — продавец», то есть начинают торговаться. Это появление соглашения об уровне услуг (SLA), в которых определён диапазон уровней сервиса. Если бизнес хочет, чтобы ИТ-служба предоставила ИТ-сервис с уровнем 1, то бизнес должен нести одни обязательства, если с уровнем 2 — то другие. В случае аутсорсинга обязательства определяют тариф, за которым стоят конкретные деньги. Хорошо спроектированный ИТ-сервис должен быть построен так, что большинство изменений требований к нему со стороны бизнеса должно приводить не к разработке новой функциональности, а к изменению уровня, заданного в спецификации ИТ-сервиса. В противном случае каталог сервисов быстро наполняется короткоживущими ИТ сервисами, эксплуатация сростётся с «ползучей» разработкой, и вся сервисная модель в организации быстро деградирует. Там, где в SLA нет возможности выбора уровня сервиса, там нет и торга.

Что получает бизнес от внедрения модели ITSM, когда отношения с ИТ-службой переводятся из «начальник — подчинённый» в «покупатель — продавец»? Отдавая право управления, бизнес превращает ИТ-службу в «чёрный ящик» с хорошо контролируемым входом и выходом. На входе — деньги (ресурсы), на выходе ИТ-сервис с заданным уровнем. Правила управления этим «чёрным

ящиком» определяются SLA. С одной стороны, он теряет прямой контроль за своими ИТ-ресурсами, но с другой — бизнес получает три мощных рычага не прямого, а косвенного управления ИТ:

- спецификация ИТ-сервиса;
- деньги (ресурсы);
- угроза конкурентного выбора поставщика ИТ-сервиса (приобретения ИТ-сервиса у сторонней организации).

Что получает ИТ-служба от внедрения модели ITSM?

1. Руководители ИТ службы получают большую самостоятельность в принятии решений, которая позволяет создать более эффективную систему управления ИТ.

2. Руководители ИТ-службы получают возможности для глубокой стандартизации не только деятельности своих подразделений, но и регламентации взаимоотношений с бизнесом. Для руководителя ИТ-службы это всегда повышение его статуса в организации и укрепление его позиции в профессиональном сообществе.

3. С выводом ИТ-службы на аутсорсинг у неё появляется возможность получения прибыли и создания полноценного бизнеса. Последовательное внедрение модели ITSM создаёт мощные стимулы для обеих сторон в:

- повышении уровня ИТ-сервиса до уровня лидеров рынка;
- снижении объёма затрат на ИТ.

Однако, внедрение модели ITSM создаёт революционную ситуацию, связанную с тем, что раньше в рамках отношений «начальник-подчинённый» все обязательства лежали на ИТ службе и определялись организационными положениями и технологическими регламентами, теперь в рамках отношений «покупатель — продавец» обязательства становятся взаимными и регулируются прямыми SLA соглашениями между бизнесом и ИТ-службой. То есть ITSM — это модель партнёрства бизнеса и ИТ, основанная на рыночных прин-

ципах, когда бизнес готов делиться властью со своим функциональным подразделением.

Если у обеих сторон от внедрения ITSM-модели появляются такие большие выгоды, то почему она так трудно внедряется? Дело в том, что у модели ITSM есть серьёзные объективные ограничения:

- далеко не всякий бизнес готов выстраивать отношения со своим функциональным подразделением на рыночных принципах «покупатель — продавец»;
- Даже тогда, когда бизнес готов двигаться в сторону рыночных отношений со своей ИТ-службой, на пути ITSM встают технические ограничения, связанные с особенностями ИТ-архитектуры.

Способность к использованию модели «покупатель — продавец»

Для того, чтобы понять, при каких условиях бизнес готов к построению отношений со своей ИТ-службой на рыночных принципах «покупатель — продавец», нужно ответить на вопрос: «При каких условиях бизнес готов поделить управленческой властью со своим подчинённым подразделением?».

Рассмотрим такую аналогию. Пользуясь мобильным телефоном, я покупаю телекоммуникационные услуги у оператора. С ним я договариваюсь, и мы в прямом соглашении определяем взаимные обязательства. Он создаёт для меня определённую ценность — гарантию мобильной связи с определённым уровнем качества в любой точке России и мира. Представим, что в моем аппарате мобильного телефона работает команда человечков, которая каким-то образом обеспечивает работу этого аппарата. Аппарат — мой, я его купил. Команду я нанял для того, чтобы они выполняли свои функции. Если что-то будет работать не так, я не буду с ней договариваться, я буду ей командовать. При каких условиях эта команда человечков может соз-

давать для меня какую-то дополнительную ценность, чтобы у меня появилась необходимость с ней договариваться?

Факторы, подталкивающие бизнес к модели ITSM, формируются с обеих сторон: как в бизнесе, так и в самих ИТ. Далее для обеих сторон рассмотрены группы факторов, с которыми встречался автор.

Факторы со стороны бизнеса

1. Где бизнес начинает стандартизацию своих бизнес процессов, там появляются унифицированные и широко используемые бизнес-задачи. Здесь большое значение имеет масштаб использования унифицированных операций и степень изменчивости бизнеса. Нет унификации задач – все ИТ-решения будут уникальными. Например, в холдинге есть небольшое предприятие мелкосерийного производства, обладающее специфической технологией. Для поддержки MES-системы своей собственной разработки предприятие держит свою ИТ-службу и никаких ИТ-услуг в этой области нет. У этого предприятия есть планы внедрения покупной, более унифицированной MES-системы, но при этом ИТ служба останется на предприятии в том же статусе функционального подразделения. Никому в холдинге, кроме этого предприятия, такая локальная задача не нужна. Другое дело в торговой сети, где, например, кассовое обслуживание — задача не только унифицированная, но массово используемая. Там под неё сделана корпоративная ИТ-услуга и вся специфика кассового обслуживания «уложена» в различные уровни этой услуги. Кстати, в ходе кризиса, когда многие компании начали искать новые продукты, технологии и рынки, они начали бурно перестраивать свой бизнес и стали отказываться от многих корпоративных ИТ-сервисов, загружая свои функциональные ИТ-службы. В этих компаниях остались только те корпоративные ИТ сервисы, где сохранились унифицированные задачи, например, такие как интернет, элек-

тронная почта или корпоративный документооборот.

2. Организация хочет фокусироваться на своих критически важных бизнес-процессах и поэтому в виде общей ИТ-услуги готово потреблять только поддержку второстепенных и не критичных задач. Например, представим себе, CIO управляющей компании спросил генерального директора нашего мелкосерийного предприятия о потребности передачи сопровождения своей MES-системы как корпоративной ИТ-услуги. Разумеется, тот ответил бы отказом, ведь MES-система критически важна для его предприятия, и он требует от своего менеджмента сохранять прямой контроль за своей ИТ-службой. С другой стороны, поддержку других бизнес-задач, например, бухгалтерского и кадрового учёта, он готов покупать в виде корпоративной ИТ-услуги, как и документооборот, почту.

3. Некоторые крупные компании выбирают бизнес-стратегию кооперации своих дивизионов, построенной по принципу рынка «покупатель — продавец». В рамках этой стратегии происходит централизация вспомогательных служб и превращение их в корпоративные сервисные центры. В этом случае бизнес сам вынуждает ИТ-службы внедрять ITSM. Например, именно это произошло в РАО ЕЭС и РЖД.

Факторы со стороны ИТ

1. К переходу на модель ITSM может подтолкнуть такая сложность и растущие затраты на ИТ, при которых бизнес уже не в состоянии осуществлять «ручное управление» своей ИТ-службой. В этом случае ИТ-служба просто подбирает право управления ИТ-ресурсами, которое бизнес де-факто уже потерял. На пример, когда бизнес двух торговых сетей холдинга начал стандартизацию своих бизнес-процессов с целью более глубокой их интеграции, обнаружилось, что главным «узким местом» стало ИТ. Оно было похоже даже не на «поскутное одеяло», а на «винегрет» из информационных си-

стем, и поглощало затраты как чёрная дыра. Бизнесу ничего не оставалось, как эту чёрную дыру превратить в «чёрный ящик», управляемые входы и выходы которого смогла обеспечить модель ITSM.

2. В последнее время получила распространение практика принудительного (шокового) выведения своей ИТ-службы на аутсорсинг, исходя из самых разных соображений, имеющих лишь косвенное отношение к ИТ. Представим себе, что генеральный директор говорит своему ИТ-директору: «Со следующего квартала ИТ выводится на аутсорсинг. Теперь вы будете получать не зарплату в нашей организации, а зарабатывать деньги в своей компании, продавая нам ИТ сервисы». При этом бизнес часто не подразумевает, что за словом зарабатывать деньги лежат отношения «покупатель — продавец», а не «начальник — подчинённый». Вывести на аутсорсинг функциональные обязанности принципиально невозможно, вывести можно только сервис. А это значит, что такие шоковые условия подталкивают обе стороны к форсированному внедрению модели ITSM. В противном случае бизнес ждёт мучительная лихорадка с функционированием информационных систем, а ИТ-службу потеря квалифицированного персонала.

Способность к ИТ-сервисам в различных ИТ-архитектурах

Даже когда бизнес и ИТ-служба готовы двинуться к рыночным отношениям «покупатель — продавец», на их пути могут возникнуть ограничения, связанные с особенностями ИТ архитектуры. В зависимости от сочетаний этих двух факторов: готовности к партнёрству и особенностей ИТ архитектуры, могут наблюдаться различные сценарии управления ИТ, среди которых встречается и ITSM-модель. Сильное влияние на формирование такого сценария оказывает стиль ИТ-архитектуры, определяющий логику организации информационного пространства предприятия.

В целом можно выделить четыре сценария развития отношений между ИТ службой и бизнесом.

1. «Пожарная команда». К этому сценарию ИТ-служба приходит в условиях, когда бизнес не хочет с ней договариваться. ИТ-служба для бизнеса — ресурс, который он использует. Если такое использование ИТ службы бизнесу кажется неэффективным, то «виновата», как правило, в этом сама ИТ-служба. Данный сценарий не исключает наличия Service Desk и каких-либо процессов поддержки, например, управления инцидентами и проблемами, но совершенно исключает появление ИТ-услуг.

2. «Серый кардинал». Сценарий часто возникает, например, при масштабном внедрении ERP-системы в организации, в которой реально появились лишь фрагментарные бизнес процессы, и вся основная деятельность так и осталась построена на личных отношениях и прецедентах. Это внедрение «на вырост». Удачный запуск системы в эксплуатацию заслуженно приближает руководителя ИТ службы к первым лицам бизнеса. При этом все руководители бизнес-подразделений обнаруживают новый центр влияния в отношениях, который базируется на экспертном знании ИТ-специалистов и близостью к первым лицам. Иногда это становится сюрпризом и для самих первых лиц. Если в этой ситуации бизнес оказывается не готов к движению в направлении построения модели отношений с ИТ-службой «покупатель — продавец», то «серый кардинал» быстро эволюционирует в «пожарную команду» с последующей эволюцией ИТ-архитектуры к типу «лоскутное одеяло».

3. «Центр компетенции». К этому сценарию ИТ-служба приходит часто с масштабным внедрением сильно интегрированной ERP-системы в организации, в процессе которого бизнес сам систематически регламентирует свою основную деятельность в виде бизнес-процессов. В архитектуре «сильной интеграции» вы-

сокая эффективность оплачивается высокой сложностью системы. Ценой высокой сложности всегда становится высокая квалификация тех, кто умеет работать с этой сложностью. Работу такой команды необходимо хорошо оплачивать, да ещё и суметь удержать, так как после завершения проекта они на ИТ-рынке нарахват. Это одна из скользких сторон крупных и даже успешных проектов по созданию сильно интегрированных корпоративных систем. Если бизнес в такой ситуации не готов делиться полномочиями с ИТ-службой, то «центр компетенции» начинает балансировать на грани «серого кардинала». Для того, чтобы не сорваться в этот сценарий, бизнес должен отчётливо понимать те выгоды, которые он получает от архитектуры «сильной интеграции», и те полномочия ИТ службы, на которые ему лучше не покушаться. В этом сценарии в ИТ-службе можно встретить большинство процессов ITSM, но полноценная модель ITSM все равно не формируется. Связано это с тем, что функциональность сильно интегрированной ERP-системы не может быть разделена на независимые ИТ-услуги. В результате весь каталог услуг сворачивается до одной большой монолитной услуги, называемой, к примеру, «доступ к ERP системе». В этом случае бизнес и ИТ-служба могут выстраивать отношения по типу «покупатель – продавец», но только продаются не ИТ-услуги, а компетенции команды.

4. «Сервисный центр». Именно этот сценарий идеально ложится на модель ITSM и хорошо описывается ITIL. Он имеет свои небольшие модификации. В архитектуре «слабой интеграции» каталог услуг может точно соответствовать прикладным услугам (почта, групповая работа, поиск, регистрация документов, согласование, хранение и так далее). В архитектуре «лоскутное одеяло» каталог услуг выстраивается вокруг бизнес-приложений (лоскутов). В любом каталоге услуги должны быть независимы друг от друга. Это принципиально отличает данный сценарий от сце-

нария «центр компетенции».

Именно в сценарии «сервисный центр» ИТ-услуги хорошо выносятся на аутсорсинг. Однако, если бизнес не хочет видеть в ИТ-службе партнёра, то ИТ услуги в лучшем случае существуют в технической поддержке элементов ИТ-инфраструктуры, куда бизнес редко заглядывает.

В каждом бизнесе реальная архитектура всего корпоративного информационного пространства представляет собой уникальный симбиоз различных типов архитектур, который оказывается чрезвычайно чувствителен к изменениям самого бизнеса и статуса его ИТ службы. В этом случае построение реальных отношений между ИТ-службой и бизнесом также будет синтезом различных вышеприведённых сценариев. Поэтому реальное превращение функциональных обязанностей ИТ-службы в ИТ-услуги — это процесс не быстрый, а иногда и обратимый.

Видимо, с этим связаны и реальные масштабы использования модели ITSM. И это нормально. Наличие ограничений применения — это свойство любой практически значимой методологии. Если мы не понимаем её ограничений, то мы, соответственно, не понимаем, как эффективно её использовать. Универсальны только неработающие методологии.

Задача перехода на модель ITSM должна рассматриваться в контексте всей ИТ-стратегии бизнеса, которая должна анализировать долгосрочные тенденции изменения самого бизнеса. Особенно остро проблема ИТ-услуг возникает с появлением у бизнеса планов аутсорсинга. На аутсорсинг можно вынести только сервисы. Многочисленные попытки вытолкнуть на аутсорсинг свою ИТ-службу, просто назвав её функциональные обязанности ИТ-услугами, всегда заканчивались серьёзными проблемами для бизнеса. Представленный анализ показывает также, что вывод на аутсорсинг — процесс тоже постепенный, к тому же он очень даже обратим.

Часть 2. Управление ИТ деятельностью

Глава 2.4

Управление разработкой программного обеспечения



Марина
Аншина

История разработки программного обеспечения

Практически ни одна организация не может обойтись без разработки или доработки программного обеспечения под свои потребности.

Разработка программного обеспечения (software development) — это род деятельности и процесс, направленный на создание и поддержание работоспособности, качества и надёжности ПО.

Википедия

Программирование — деятельность по написанию программ — инструкций для компьютеров, в соответствии с которыми они выполняют определённые действия.

Трудно сказать, когда возникло программирование. На это могут претендовать три даты.

1. Создание первого программируемого устройства, Жаккардового ткацкого станка, который был построен ещё в 1804 году Жозефом Мари Жаккаром для программирования узоров на тканях при помощи прообраза перфокарт.

2. Создание проекта аналитической машины, первого программируемого вычислительного устройства, придуманного, но так и не построенного, Чарльзом Беббиджем в сентябре 1834 г.

3. Написание 19 июля 1843 года дочкой поэта Джорджа Байрона графиней Адой Августой Лавлейс для этой аналитической машины первой программы для решения уравнения Бернулли.

Интересно, что графиня Ада Лавлейс сделала для программирования ещё много хорошего. Например, она впервые высказала принцип экономии ячеек памяти, использовала цикл, ввела понятия подпрограммы, библиотеки подпрограмм, индексного регистра. В знак признания её заслуг перед программированием в 80-х годах 20 века появился язык программирования Ада, названный в её честь. Ада Лавлейс прекрасно определила смысл программирования: «Аналитическая машина не претендует на то, чтобы создавать

что-то действительно новое. Машина может выполнить все то, что мы умеем ей предписать». Ада Лавлейс, 1843

Первый работающий программируемый компьютер был построен Джорджем Шотцем из Стокгольма и показан на Всемирной выставке в Париже в 1855 году. Это была машина, сделанная по принципу Бэббиджа, но намного более простая. Она выполняла арифметические действия четвертой степени и печатала результат с точностью до восьмого знака после запятой. Первым электронным компьютером принято считать «Колосс 1», созданный в военном ведомстве Великобритании во время Второй Мировой войны. Он начал работать в декабре 1943 года и был предназначен для дешифрования перехваченных сообщений противника. Этот компьютер позволил английской разведке расшифровать считавшийся супернадёжным немецкий код «Энигма». Так как «Колосс 1» был сверхсекретным проектом, первым электронным компьютером часто называют ЭНИАК — Electronic Numerical Integrator And Computer, созданный для Департамента артиллерии армии США в феврале 1946 года. Первые программы для компьютера писались в машинных кодах и представляли собой последовательность нулей и единиц. Конечно, такое программирование было весьма непростым делом, но зато позволяло максимально приблизиться к тому, как «думает» машина. Эти программы вводились в компьютеры через перфокарты (карточки из тонкого картона) или перфоленты (бумажные ленты), на которых информация в двоичной форме соответствовала наличию или отсутствию отверстий в позициях, и обрабатывались процессором.

По мере превращения программирования в повседневную деятельность возникла необходимость в создании языков, программирование на которых было бы проще. Так появился Ассемблер, а затем его российский аналог — Мнемокод. В основе Ассемблера лежит английский язык, Мнемокода — русский.

Assembler (Ассемблер) (от англ. assembly language) — это язык низкого уровня — т. е. он относится к языкам, близким к программированию в машинных кодах. По сравнению с программами, написанными в машинных кодах, программы на Ассемблере легче писать и читать. Вариантов Ассемблера существует великое множество — для каждого типа процессора свой. Преимуществами языков низкого уровня является максимальное использование возможностей конкретного типа процессора. Несомненными недостатками — сложность программирования и соответственно длительность отладки (выявления и устранения ошибок программы), большое количество диалектов.

Появление языков высокого уровня ознаменовало новый этап в развитии программирования. Эти языки были разработаны для упрощения процесса написания и отладки программ и абстрагировались от конкретного процессора. С их появлением профессия программиста становится по-настоящему массовой и доступной. Языки высокого уровня «оторваны» от реализации конкретного процессора, поэтому не могут так эффективно использовать его особенности, как языки низкого уровня. Однако и они имеют свои варианты и диалекты, которые в той или иной мере учитывают особенности если не конкретного процессора, то семейства процессоров.

Примеры языков высокого уровня всем хорошо известны. Это Алгол, Фортран, С, С++, С#, Java, Паскаль, Delphi, Python и многие другие. Выполнение программ, написанных на языках высокого уровня, может осуществляться принципиально двумя способами. Либо программа обрабатывается сначала компилятором (транслятором), а потом компоновщиком, либо используется интерпретатор.

Транслятор — программа или техническое средство, осуществляющее перевод программы с одного языка на другой. Транслятор также осуществляет проверку программы

и выявляет синтаксические ошибки. Результат работы транслятора называют объектным кодом.

Компилятор — это разновидность транслятора, который формирует машинный код, готовый для исполнения компьютером. Компилятор также выполняет объединение программ, относящихся к одной задаче.

Компоновщик собирает несколько программ, обработанных транслятором, библиотеки для конкретного типа процессора, и генерирует машинный код, который может быть исполнен компьютером. Компоновщик также выявляет ошибки, но другого рода: неопределённые функции, несоответствия в скомпонованных программах. На вход компоновщика поступает объектный код, а на выходе получается исполняемый модуль (программа).

Интерпретатор — это программное обеспечение, которое умеет выполнять программу на языке высокого уровня на компьютерах. Обычно это делается с помощью «виртуальной машины», которая запускается на компьютере и пошагово интерпретирует программу.

Существуют различные «промежуточные» варианты, например, когда интерпретатор перед исполнением программы транслирует её на другой язык, более удобный для интерпретации (например, таковы языки PYTHON и Perl). История языков высокого уровня началась в конце 40-х годов, когда Джон Моучли, один из создателей ЭНИАКа, разработал первый примитивный интерпретатор под названием «Short Code». В 1951 г. ученица Моучли Грейс Хоппер создала первый компилятор и придумала сам термин. В 1954 году группа под руководством Грейс Хоппер разработала язык программирования и его компилятор, которые называли MathMatic. В ходе его усовершенствования в 1958 г. появился язык FlowMatic, который считают первым языком для обработки коммерческих данных. На его основе как язык для бизнеса в 1960 году по-

явился Кобол (COBOL — Common Business Oriented Language), который достаточно широко используется и по сей день, особенно в США. В 1954 г. появился Фортран (FORTRAN, от FORMula TRANslator — переводчик формул), разработанный группой программистов IBM и направленный на осуществление сложных математических расчётов, который также используется по сей день.

В 1964 г. был создан специализированный язык программирования, состоящий из простых слов английского языка. Его называли «универсальным символическим кодом для начинающих» (Beginner All Purpose Symbolic Instruction Code, или, сокращённо, BASIC). Широкому распространению БЭЙСИКа способствовало то, что в конце 70х годов его начали использовать как встроенный язык персональных компьютеров.

Однако большие программы на Фортране читать и понимать достаточно сложно. А на БЭЙСИКе их написать практически невозможно. Поэтому в 1958 г. появился Алгол (ALGOrithmic Language), название которого подчёркивает, что он предназначен для записи алгоритмов. Алгол считается первым языком структурного программирования. К ним также относятся Паскаль (1970) и Си (1972).

Структурное программирование, что следует из его названия, предлагает более структурированный, системный подход к написанию программы, наличие блоков и подпрограмм, отсутствие безусловных переходов, выделение локальных переменных. Однако по мере усложнения программ и структурное программирование оказалось недостаточно эффективным.

В начале 1980 годов появилось объектно-ориентированное программирование (ООП), основанное на понятиях объекта и класса.

Объект в ООП — сущность, обладающая определённым состоянием и поведением, имеющая заданные значения свойств (атрибутов) и операций, которые над ним можно выпол-

нить (методов). Как правило, в ООП объекты группируются в классы, которые определяют поведение объекта, а объект также называют экземпляром класса.

На практике ООП обычно сводится к созданию некоторого количества классов, включая интерфейс и реализацию, и последующему их использованию. ООП основано на четырёх (иногда выделяют только три первых) принципах: инкапсуляция, наследование, полиморфизм, абстракция.

Инкапсуляция — свойство, позволяющее объединять данные в объект и скрывать его реализацию. При этом взаимодействие с объектом осуществляется только через спецификацию (интерфейс) объекта.

Наследование позволяет описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса наследуются. Поэтому с классом-наследником можно взаимодействовать по тому же интерфейсу, что и с родительским.

Полиморфизм определяет возможность объектов с одинаковой спецификацией (интерфейсом) иметь различную реализацию. Например, реализация класса-наследника отличается от реализации класса-родителя.

Абстракция — это отделение объекта от других объектов по конкретным характеристикам (атрибутам). Абстракция позволяет работать с объектами как с черными ящиками, используя их инкапсуляцию.

Примерами объектно-ориентированных языков могут служить Object Pascal, C++, Java, PYTHON. ООП организует программы, разбивая их на составные части, и позволяет работать над ними группам разработчиков.

В заключение краткого экскурса в историю программирования необходимо сказать про интегрированные средства разработки.

Интегрированная среда разработки (англ. IDE, Integrated development) — система программных средств, используемая для раз-

работки программного обеспечения. Такие среды обычно включают в себя:

- текстовый редактор;
- компилятор и/или интерпретатор;
- средства автоматизации компоновки;
- отладчик;
- систему управления версиями;
- средства создания графического интерфейса пользователя;
- графические средства построения классов.

Именно среды разработки являются наиболее популярным средством разработки программ в настоящее время. Наиболее известные среди них: Eclipse, Microsoft Visual Studio, в которых можно вести разработку на различных языках программирования, и Visual Basic, Delphi (Pascal), IDLE (PYTHON), предназначенные для одного языка.

Software Engineering

Разработка ПО является одним из сложнейших видов деятельности в ИТ, связанным с рядом проблем и рисков. Задача улучшения продуктивности, качества и надёжности разработки стоит уже в течение нескольких десятилетий.

Согласно наиболее распространённой точке зрения, чтобы своевременно производить и использовать качественное ПО, необходимо применять современные принципы, методы и стандарты жизненного цикла ПО, которые объединены в область знания Software Engineering.

Программная инженерия (Software Engineering) — применение систематического, упорядоченного, количественного подхода к разработке, эксплуатации и поддержке программного обеспечения, т.е. приложение профессионального инженерного подхода к разработке программного обеспечения.

IEEE 610.12

Программная инженерия подходит к разработке ПО именно как к инженерной дея-

тельности, она достаточно хорошо изучена и стандартизирована. Ниже приведены и кратко описаны основные стандарты и методологии этой дисциплины. Сначала введём несколько терминов:

Жизненный цикл (Life Cycle) — это эволюция системы, продукта, услуги, проекта или других изготовленных человеком объектов, начиная со стадии разработки концепции и заканчивая прекращением применения.

Модель жизненного цикла (Life Cycle Model) — это рамочная модель процессов и действий (или работ), связанных с жизненным циклом и организованных в стадии, которая также служит в качестве общей основы для коммуникаций и взаимопонимания сторон.

ГОСТ Р ИСО/МЭК 12207-2010

Методология (метод) — это конкретное описание работ в рамках определённой модели жизненного цикла.

Другими словами, модель жизненного цикла определяет основные стадии (фазы) жизненного цикла и принципы перехода между ними, но не детализацию работ проекта. Детальное содержание работ и ролевую ответственность специалистов на всех этапах выбранной модели жизненного цикла задаёт методология управления проектом разработки.

Внутренняя разработка и аутсорсинг

Для выполнения разработки или доработки ПО у организации есть 2 способа:

- Сформулировать требования к ПО и отдать проектирование, разработку, тестирование и сопровождение системы или какую-то часть этих процессов внешней компании (аутсорсинг или частичный аутсорсинг функций — например, тестирования ПО);
- Выполнять вышеуказанные процессы самостоятельно, силами собственных специалистов, взяв на себя ответственность за качество этих процессов и управление ресурсами, которые обеспечивают их выполнение.

Как всегда, бывает, однозначного ответа на вопрос «что лучше» дать нельзя; и тот, и другой подход имеют свои положительные и отрицательные стороны, плюсы и минусы, возможности и риски, которые приведены в Табл. 2.4.1.

Отметим, что распространённый тезис о низкой стоимости внутренних разработок весьма спорен, вопрос стоимости надо решать конкретно для каждого проекта.

Табл. 2.4.1. Сравнение аутсорсинга и внутренней разработки ПО.

Вариант	Плюсы	Минусы	Возможности	Риски
Аутсорсинг разработки ПО	Выведение непрофильной функции за рамки предприятия. Не надо управлять высококвалифицированным персоналом. Определённые гарантии качества ПО и сроков разработки.	Потеря гибкости. Необходимость чётко сформулировать требования. Необходимость контроля исполнителя. Зависимость от аутсорсинговой компании.	Построение долговременных отношений с аутсорсером для развития и эксплуатации ПО. Повышение качества ПО. Уменьшение стоимости владения полученной системой.	Получение ПО бесполезного или с невостребованным функционалом (при ошибке в требованиях или их устаревании). Трудности в организации сопровождения и эксплуатации.

Собственная (внутренняя) разработка	Гибкость. Возможность непрерывного тесного контакта с заказчиком. Хорошее знание разработчиками архитектуры предприятия, его бизнес-процессов, существующего ПО. Ориентация разработчиков на длительное взаимодействие с заказчиком. Как правило в той или иной степени налаженные вспомогательные процессы (управление изменениями, управление конфигурациями), в которых уже участвует заказчик. Сохранение ценных ИТ-кадров в компании.	Необходимость управления непрофильной высокотехнологичной функцией. Относительно низкий уровень собственных ИТ-специалистов. Низкий (как правило) уровень регламентации процессов разработки и управления качеством.	Управляемость и контроль. Минимальные или полностью отсутствующие дополнительные затраты на управление. Получение первого результата в кратчайшие сроки.	Уход ключевых специалистов. Низкое качество ПО. Превышение установленных сроков проекта. Высокая стоимость владения полученной системой, сложность её технической поддержки, управления изменениями.
-------------------------------------	--	--	--	--

Основные стандарты и методологии разработки ПО

Software Engineering Book of Knowledge

Software Engineering Book of Knowledge (SWE-BoK) — это свод знаний по программной инженерии, подготовленный в 2004 году IEEE Computer Society и принятый как стандарт ISO/IEC 9579:2004. SWEBOOK является основополагающим документом, отражает мнение многих зарубежных и отечественных специалистов в области программной инженерии и согласуется с процессами жизненного цикла ПО стандарта ISO/IEC 12207. SWEBOOK описывает 10 областей знаний:

1. Программные требования (Software requirements) — извлечение (сбор), анализ, специфицирование и утверждение требований.

2. Проектирование ПО (Software design) — определение архитектуры, компонентов, ин-

терфейсов и других характеристик системы или её компонентов. Результат процесса проектирования — дизайн системы.

3. Конструирование ПО (Software construction) — создание рабочей системы с привлечением методов верификации (тестирования), тестирования модулей (unit testing), интеграционного тестирования и отладки.

4. Тестирование ПО (Software testing) — деятельность, выполняемая для оценки и улучшения качества ПО, в общем случае, базируется на обнаружении дефектов и проблем в программных системах.

5. Сопровождение ПО (Software maintenance) — совокупность действий по обеспечению работы ПО, по внесению изменений в случае обнаружения ошибок в процессе эксплуатации, по адаптации ПО к новой среде функционирования.

ния, а также по повышению производительности или других характеристик ПО.

6. Управление конфигурацией ПО (Software conguration management) — идентификация конфигурации системы, с целью контроля её изменений конфигурации, а также поддержки и сопровождения целостной конфигурации на протяжении всего жизненного цикла системы.

7. Управление инженерией ПО (Software engineering management) — управление (планирование, координация, количественная оценка, мониторинг, контроль и отчётность) для систематического, упорядоченного и количественно измеряемого обеспечения разработки и сопровождения программных систем.

8. Процесс программной инженерии (Software engineering process) — область знаний, существующая на двух уровнях:

- техническая и управленческая деятельность в течение процессов жизненного цикла ПО, включающих приобретение, разработку, сопровождение и вывод из эксплуатации;
- определение, реализация, оценка, измерение, управление, изменение и совершенствование самих процессов жизненного цикла ПО.

9. Инструменты и методы программной инженерии (Software engineering tools and methods) — инструменты, предназначенные для обеспечения поддержки процессов жизненного цикла ПО, а также методы, обеспечивающие проектирование, реализацию и выполнение ПО на процессах, а также достижение качества процессов и продуктов.

10. Качество ПО (Software quality) — область знаний, связанная с характеристиками ПО и его способностью удовлетворить установленным или предполагаемым потребностям заказчика. Кроме того, SWEBoK включает обзор смежных дисциплин, связь с которыми важна для программной инженерии:

- компьютерная инженерия (Computer engineering);

- применение вычислительной техники (Computer science);
- управление (Management);
- математика (Mathematics);
- управление проектами (Project management);
- управление качеством (Quality management);
- системная инженерия (Systems engineering).

В соответствии со своим названием, SWEBoK предлагает ряд принципов и лучших практик организации процесса разработки ПО и может послужить основой корпоративных стандартов разработки ПО.

Project Management Book of Knowledge

Project Management Book of Knowledge (PMBoK) — стандарт управления проектами, который может быть адаптирован для проектов разработки ПО (подробнее см. главу 2.1 «Управление ИТ-проектами»).

PRINCE2 (PProjects IN Controlled Environments 2)

PProjects IN Controlled Environments 2 (PRINCE2) — это один из наиболее известных и популярных стандартов ведения ИТ проектов. Он был разработан в 1989 году Central Computer and Telecommunications Agency (CCTA) Великобритании как стандарт для руководства проектами в области информационных технологий. В настоящее время он широко используется и в других областях и является «de facto» стандартом для руководства проектами в Великобритании. Стандарт описывает принципы организации программы проектов и проектов в рамках программы.

В нем выделены 7 принципов управления проектами:

1. Постоянная оценка соответствия проекта потребностям бизнеса.
2. Учёт уроков проекта, «самообучение» проекта.
3. Выделение определённых ролей с ответственностью и правами.

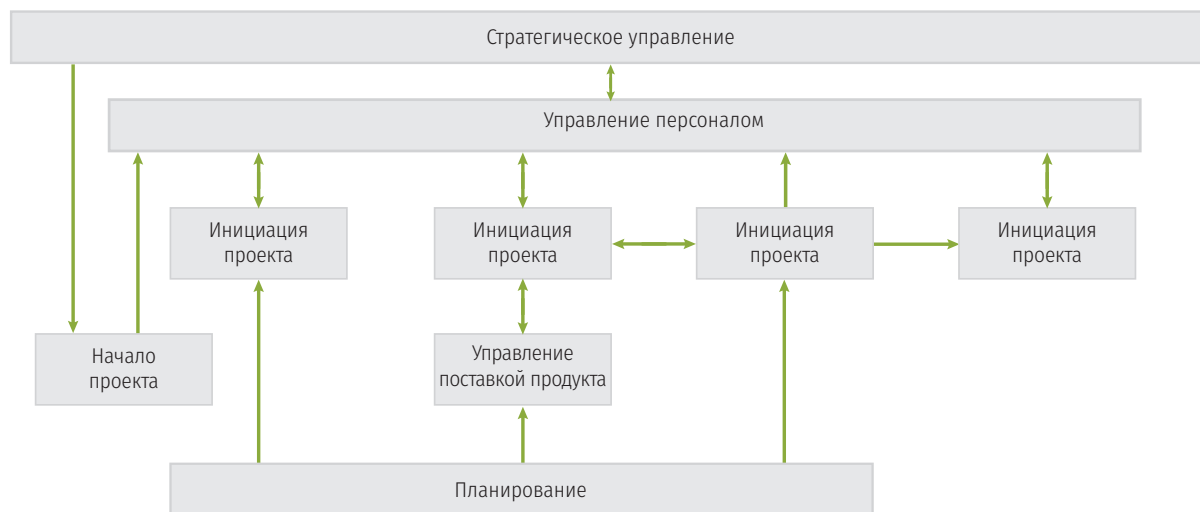
4. Управление по фазам. Текущая фаза завершается планированием следующей.
5. Управление по отклонениям. Каждая роль имеет полномочия на утверждение конкретных отклонений.
6. Ориентация на результат. Главное — имен-

но результат, а не процессы его достижения (в отличие, например, от PMBoK).

7. Адаптация стандарта под конкретную организацию и внешнюю среду.

Стандарт разбивает проект на 8 стадий (Рис. 2.4.1):

Рис. 2.4.1. Стадии проекта в соответствии с PRINCE2.



1. Начало проекта (Starting Up a Project) — определение того, надо ли выполнять проект и выбор подхода к его реализации.

2. Инициация проекта (Initiating a Project) — создание бизнес-кейса проекта, формирование оргструктуры, плана и бюджета проекта.

3. Планирование (Planning) — детальное планирование и перепланирование проекта, выполняется в течение всего проекта.

4. Руководство проектом (Directing a Project) — принятие Комитетом проекта решений по контрольным точкам и ситуационное управление по значительным проблемам и отклонениям. Выполняется в течение всего проекта.

5. Систематический контроль стадий (Controlling a Stage) — непосредственная работа менеджера проекта по ежедневному управлению проектом (выдача и приёмка заданий, фиксация сложностей и рисков, принятие решения об эскалации и отчётность).

6. Управление поставкой продукта (Managing Product Delivery) — действия по созданию нужных для проекта продуктов, выполняемые менеджером проекта.

7. Управление границами стадий (Managing Stage Boundaries) — анализ исполнения плана стадии, промежуточное планирование следующей стадии, обзор рисков и предоставление информации Комитету проекта.

8. Завершение проекта (Closing a Project) — действия по закрытию проекта. Стадии в стандарте не являются последовательными, зависимости между ними более сложные. Каждая стадия детализирована по активностям (всего их 45), входящим и исходящим артефактам, критериям, позволяющим начать активность. Кроме того, стандарт выделяет 7 артефактов (документов и тем), на которых надо фокусироваться при выполнении проекта.

1. Соответствие потребностям бизнеса (бизнес-кейс).

2. Оргструктура проекта.
3. Обеспечение качества (по 3-м направлениям: финансовое состояние проекта, выполнение требований пользователей, технологическое качество результата — продукта проекта).
4. Планирование.
5. Управление рисками.
6. Управление изменениями.
7. Отчётность (в частности, план/факт).

Большим преимуществом стандарта является его глубокая проработанность и универсальность. В целом PRINCE2 является методологией, готовой к использованию для выполнения ИТ-проектов и проектов по разработке ПО. Главным недостатком PRINCE2 является высокая сложность применения. Распространение его в России сдерживает также отсутствие официального перевода на русский язык.

Стандарты жизненного цикла программных систем

Как и SWEBoK, основу для организации процесса разработки ПО составляют стандарты ИСО/МЭК жизненного цикла систем и программных систем: ISO/IEC 14764:2006, ISO/IEC 15288:2008 и ISO/IEC 12207:2008. В России приняты как ГОСТ Р их аналоги:

- ГОСТ Р ИСО/МЭК 14764:2002 «Сопровождение программных средств» (развитие стандарта 12207 на один из этапов ЖЦ);
- ГОСТ Р ИСО/МЭК 15288:2005 «Информационная технология. Системная инженерия. Процессы жизненного цикла систем»;
- ГОСТ Р ИСО/МЭК 12207:2010 «Системная и программная инженерия. Процессы жизненного цикла программных средств». Стандарты ISO/IEC 12207:2008 и его российский эквивалент ГОСТ Р ИСО/МЭК 12207:2010 специально предназначены для разработки ПО и определяют процессы жизненного цикла ПО, начиная с формиро-

вания требований к будущей системе и заканчивая выводом ПО из эксплуатации. В них ПО рассматривается в двух вариантах (Рис. 2.4.2).

- как цельная система, слабо зависящая от внешних систем — процессы в контексте системы;
- как специфическая программная часть более комплексной ИТ системы — специальные процессы программных средств.

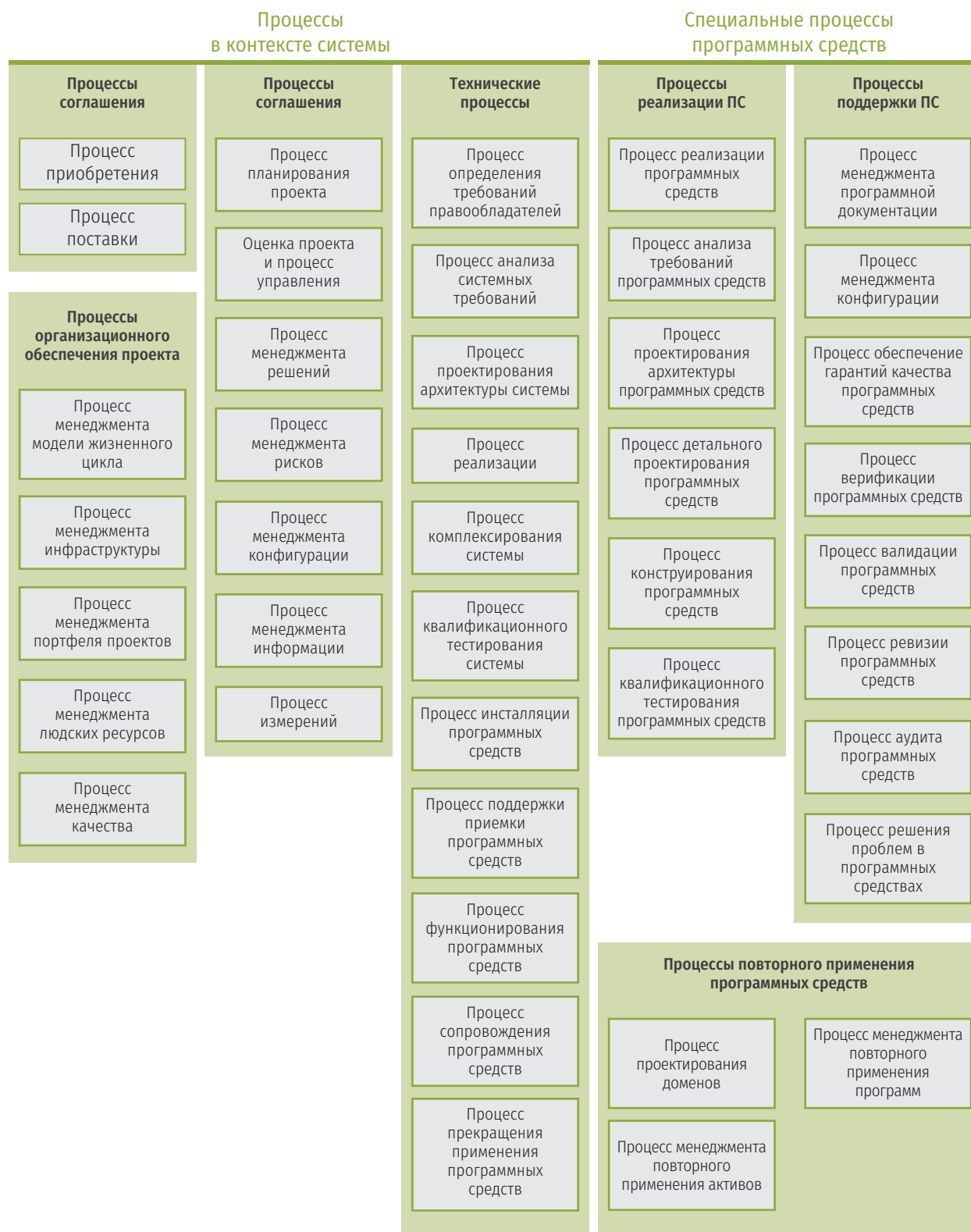
Процессы жизненного цикла ПО как системы состоят из 4 групп, включающих 25 процессов. Процессы жизненного цикла ПО как части системы состоят из трёх групп, включающих 18 процессов.

Стандарты ISO/IEC 12207:2008 и ISO/IEC 15288:2008 лежат в основе практически всех современных промышленных технологий создания и эксплуатации ПО. Отметим, что эти стандарты определяют только рамочную структуру процессов и работ, общее понимание их содержания и терминологию, но не определяют конкретный набор и последовательность стадий жизненного цикла ПО и входящих в них процессов (это определяет модель жизненного цикла). Их положения являются общими для любых моделей жизненного цикла, методов и технологий создания ПО. Стандарты ISO/IEC 12207:2008 и ISO/IEC 15288:2008 и соответствующие им ГОСТ-Р допускают, что в проектах не нужно будет применять все определённые в них процессы. Выбор ряда процессов, подходящих для организации или проекта, называется процессом адаптации стандарта. Для определения документов и их структуры, сопровождающих жизненный цикл систем и программных систем существует стандарт ISO/IEC 15289:2006, переведённый на русский язык в 2011 г.

Серия стандартов ГОСТ 34

Серия стандартов ГОСТ 34 относится к жизненному циклу автоматизированных си-

Рис. 2.4.2. Процессы жизненного цикла ПО по стандарту ГОСТ Р ИСО/МЭК 12207 2010.



стем и содержит отдельные элементы управления проектами. Например, стадии и этапы создания АС, состав, правила оформления документа «Техническое задание на создание системы, виды испытаний АС. В соответствии с ГОСТ 34, создание АС состоит из следующих стадий и этапов:

- Формирование требований к АС;
- Разработка концепции АС;
- Техническое задание;
- Эскизный проект (допускается исключать);
- Технический проект;
- Рабочая документация;
- Ввод в действие;
- Сопровождение АС.

В свою очередь в соответствии с ГОСТ 34.602.89 техническое задание состоит из следующих частей:

1. Общие сведения.
2. Назначение и цели создания системы.
3. Характеристика объекта автоматизации.
4. Требования к системе.
 - 4.1. Требования к системе в целом.
 - 4.1.1. Требования к структуре и функционированию системы.
 - 4.1.2. Требования к численности и квалификации персонала системы и режиму его работы.
 - 4.1.3. Показатели назначения.
 - 4.1.4. Требования к надёжности.
 - 4.1.5. Требования к безопасности.
 - 4.1.6. Требования к эргономике и технической эстетике.
 - 4.1.7. Требования к транспортабельности для подвижных АС.

Модель зрелости разработки

Существует модель оценки зрелости процессов разработки ПО: Capability Maturity Model (CMM) - модель зрелости создания ПО, позже развитая в Capability Maturity Model Integration (CMMI), интегрированная модель зрелости создания ПО.

4.1.8. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы.

4.1.9. Требования к защите информации от несанкционированного доступа.

4.1.10. Требования по сохранности информации при авариях.

4.1.11. Требования к защите от влияния внешних воздействий.

4.1.12. Требования к патентной чистоте.

4.1.13. Требования по стандартизации и унификации.

4.1.14. Дополнительные требования.

5. Состав и содержание работ по созданию системы.

6. Порядок контроля и приёмки системы.

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.

8. Требования к документированию.

9. Источники разработки.

Стандарты ГОСТ 34 до сих пор являются наиболее востребованными отечественными стандартами в области разработки ПО. В основном, для создания технического задания. Хотя на данный момент они в значительной мере устарели и не подходят для гибких методик разработки ПО, именно они чаще всего выступают в качестве требований в открытых и закрытых конкурсах, особенно в государственных организациях. ГОСТ 34 также не требует выполнения всех стадий разработки и наличия всех частей ТЗ, что позволяет использовать его и в наше время.

Модель CMM была создана в 1991 году Институтом программной инженерии Университета Карнеги Меллона (Software Engineering Institute, SEI) для разработки программных продуктов. С течением времени было выпущено целое семейство моделей: SWCMM — для программ-

ных продуктов, SECM — для системной инженерии, Acquisition CMM — для закупок, People CMM — для управления людскими ресурсами, ISCM — для интеграции продуктов.

Разнообразные модели оказались достаточно сложными для понимания и внедрения. Поскольку они были созданы разными группами специалистов, содержание этих моделей не всегда согласовывалось друг с другом, а также с требованиями международных стандартов. Поэтому в 2002 году SEI опубликовал новую модель CMMI, объединяющую ранее выпущенные модели и учитывающую требования появившихся к этому времени международных стандартов. CMMI включает три модели:

- CMMI для разработки (for Development) CMMI-DEV,
- CMMI для сервисов (for Services) CMMI-SVC
- CMMI для поставок (for Acquisition) CMMI-ACQ.

Наиболее известной является модель CMMI for Development, ориентированная на организации, занимающиеся разработкой ПО, аппаратного обеспечения, а также комплексных систем. Модель CMM/CMMI выделяет пять уровней зрелости компании-разработчика ПО, каждый из которых является этапом в совершенствовании процессов разработки ПО. Описание уровней приведено в Табл. 2.4.2.

Табл. 2.4.2. П. Уровни зрелости модели CMMI.

Наименование уровня зрелости	Наименование CMMI	Описание
Оптимизирующий	Optimizing	Непрерывное улучшение процессов на основании количественного анализа улучшений. Проактивное управление процессами.
Управляемый на основе количественных данных	Quantitatively Managed	Устанавливаются количественные цели в области качества и цели, связанные с выполнением процессов. Процессы контролируются статистически, что даёт возможность предсказать их поведение.
Определённый	Defined	Процессы детально и строго описаны. Установлены и применяются единые правила проектной деятельности для всей организации. Отслеживается качество процессов и предпринимаются корректирующие действия
Управляемый	Managed	Процессы планируются, оцениваются и контролируются. Часть процессов прозрачны для управления. Однако стандарты и описания процессов могут отличаться в разных проектах. Отсутствует проактивное управление процессами.
Начальный	Initial	Процессы обычно не регламентированы и хаотичны. Результаты зависят от профессионализма и опыта сотрудников и различаются между реализациями процесса. Проекты редко выполняются в рамках запланированных сроков и бюджетов.

Использование модели CMMI позволяет организации оценить эффективность процессов, установить приоритетные направления и способы их усовершенствования, а также внедрить данные усовершенствования. CMMI также определяет 22 процессные области, ряд целей, которые должны быть достигнуты при использовании CMMI в данной области и набор рекомендаций в виде практик. Сертификацию на соответствие определённому уровню CMM/CMMI производят специализированные компании. Отметим, что в России есть несколько компаний-разработчиков, сертифицированных по пятому уровню CMMI.

Модели и методы проектов разработки ПО

Напомним, что лишь стандарты PRINCE2 и ГОСТ 34 определяют конкретный набор стадий (этапов) жизненного цикла проекта по разработке ПО. Стандарт ISO/IEC 12207:2008 и его российский эквивалент ГОСТ Р ИСО/МЭК 12207 2010, а также SWEBoK предполагают, что набор стадий должна определять модель жизненного цикла. Модели жизненного цикла разработки ПО широко применяются уже достаточно давно. Помимо моделей жизненного цикла конкретное описание работ определяется методологией или методом разработки. Все модели и методы разработки ПО можно разделить на три группы:

1. Классические последовательные модели, которые состоят из чётко определённых по-

следовательных этапов, готовый программный продукт получается после успешного выполнения последнего этапа. К ним можно отнести:

- программирование и исправление ошибок(Code-and-Fix);
- каскадную модель (модель водопада).

2. Итеративные модели разработки, которые предполагают разбиение проекта не только на фазы, но и на итерации, и эволюционное развитие ПО, при котором рабочая версия ПО может быть получена на промежуточных итерациях. К итеративным моделям разработки можно отнести:

- модель Rational Unified Process (RUP);
- спиральную модель.

3. Гибкие методологии разработки (agile programming), из которых наиболее известны в России:

- Scrum;
- Extreme Programming (XP).

В Табл. 2.4.3 дано сравнение классических и гибких методов разработки ПО. Вывод: будьте готовы к применению гибких методов, но действуйте с осторожностью. Гибкие методы разработки предполагают постоянное активное участие в процессе разработки пользователей и заказчиков, что не всегда возможно. Один из важных факторов определения необходимого уровня планирования в проекте разработки ПО — количество и объём рисков, которые готова принять организация.

Табл. 2.4.3. Сравнение классических (последовательных и итеративных) и гибких методов разработки ПО

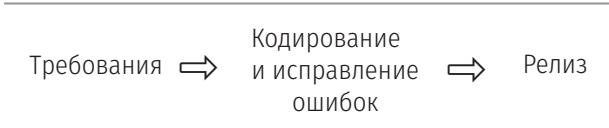
Область	Гибкие методы	Классические методы, направляемые планом (последовательные и итеративные)
Разработчики	Ориентированы на коммуникацию, профессиональны, хорошо осведомлены, находятся в одном офисе, умеют работать коллективно	Непрерывное улучшение процессов на основании количественного анализа улучшений. Проактивное управление процессами.

Заказчики	Активно участвуют в проекте, хорошо осведомлены, находятся в одном офисе, умеют работать коллективно	Непрерывное улучшение процессов на основании количественного анализа улучшений. Проактивное управление процессами.
Требования	Большей частью возникают внезапно, быстро меняются	Известны заранее, большей частью стабильны
Архитектура	Спроектирована для текущих требований	Спроектирована для текущих требований и требований, появление которых можно предсказать
Улучшение кода	Обходится не дорого	Обходится дорого
Размер	Небольшие команды и продукты	Большие команды и продукты
Первичные цели	Быстрые достижения	Высокая надёжность

Классические последовательные модели разработки ПО

Программирование и исправление ошибок (Code-and-Fix)

Это простейшая, наиболее древняя модель, неформальная и неструктурированная. Состоит из последовательно выполняемых шагов:



Преимущества, недостатки и возможное применение показаны в Табл. 2.4.4.

Табл. 2.4.4. Преимущества, недостатки и возможные применения модели Code-and-Fix.

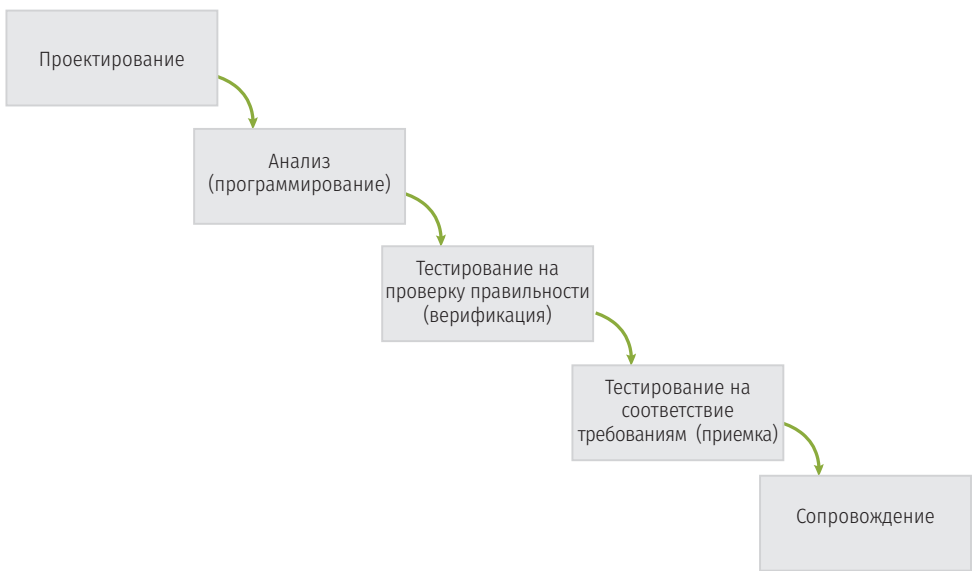
Преимущества	Недостатки	Возможные применения
• Не требует дополнительных расходов на управление. • Не требовательна к квалификации разработчиков.	• Трудно оценить параметры проекта (сроки, длительность, стоимость). • Непредсказуемое качество - Отсутствует управление рисками.	• Создание прототипов системы • Создание демо-версий. • Пилотные проекты.

Каскадная модель (модель водопада)

Основная характеристика — строго упорядоченная последовательность этапов. Каждый этап выполняется один раз и в том порядке, как они представлены на схеме (Рис. 2.4.3).

Следующий этап не может начаться до окончания предыдущего. По окончании каждого этапа осуществляется проверка, достигнута ли поставленная цель. Результаты каждого этапа документируются. Основные этапы (процессы) разработки и внедрения ПО

Рис. 2.4.3. Каскадная модель (модель водопада).



показаны на Рис. 2.4.3. Кроме них возможно также выделение других этапов:

- Разработка пользовательской документации;
- Сборка, интеграция, внедрение;
- Настройка;
- Обучение пользователей.

Каскадная модель предполагает, что работы на каждом этапе будут выполнены настолько качественно, что после её завершения и пе-

рехода к следующему этапу возвращения к предыдущему не потребуется, а все накопившиеся ошибки можно будет устранить на этапе тестирования. Отметим, что в каскадной модели огромное значение имеет этап формулировки требований и составления спецификаций — цена ошибки здесь наиболее велика. Преимущества, недостатки и возможное применение модели показаны в Табл. 2.4.5.

Табл. 2.4.5. Преимущества, недостатки и возможные применения каскадной модели.

Преимущества	Недостатки	Возможные приложения
• Возможность обнаружения ошибок на ранних стадиях. • Минимальные затраты на планирование. • Простота управления жёсткой структурой этапов проекта. • Все возможности системы реализуются одновременно	• Отсутствие практических результатов до окончания процесса – выполнения последнего этапа. • Сложность описания требований в начале проекта, когда они ещё не до конца понятны, полученный продукт может оказаться непригодным для использования из-за неправильного понимания требований. • Отсутствие гибкости, трудность, а иногда и невозможность реализации изменений, учёта изменений в технологии.	• Автоматизация стабильных процессов. • Наличие чётких, хорошо формализованных требований. • В ситуации, когда качество результата важнее стоимости и длительности проекта. • Создание первой версии ПО для проверки реализованных в нём функций.

- Система может быть слишком большая по объёму, чтобы быть реализованной в целом.
- Риск ограничения по ресурсам, угроза переработок и перерасходов.
- Чрезмерное количество документов.

Существуют вариации каскадной модели. То, что не всегда удаётся детально проработать проект будущей системы, выяснилось достаточно давно. И для учёта возможности исправлений сделанной на предыдущем этапе работы были предложены вариации каскадной модели, предполагающие возможность возврата к предыдущему этапу (эта модель получила название «возвратной»). Однако при этом резко (многократно) возрастает стоимость переделок.

Итеративные модели разработки

В условиях активного развития компании или в период серьёзной трансформации бизнес-процессов в организации, практически невозможно создавать сложное ПО последовательно, т. е. сначала выявлять все проблемы, затем принимать проектные решения, потом формировать программное обеспечение и, наконец, проверять полученное изделие. Итеративный подход обеспечивает большую гибкость при изменяющихся требованиях, позволяет последовательно улучшать понимание проблем, более эффективно и заблаговременно идентифицировать, и снижать проектные риски. В общем виде при итеративной модели этапы разработки (включая определение требований, кодирование и т.д.) могут содержать несколько итераций. Продукты этих итераций могут быть различными – как перечнем требований, так и промежуточными вариантами системы. Таким образом, при итеративной модели мы постепенно приближаемся к желаемой цели.

Модель жизненного цикла *Rational Unified Process*

Rational Unified Process (RUP) — это методология гибкой разработки ПО, созданная компанией Rational Software (с 2003 г. – подразделение IBM) и поддерживаемая комплексом инструментов Rational Suite. RUP предлагает двигаться итеративно, создавая за одну итерацию (длящуюся от 2 до 6 недель) значимый для заказчика результат (но не обязательно — готовую часть продукта). Весь цикл разработки состоит из 4 фаз (Рис. 2.4.4):

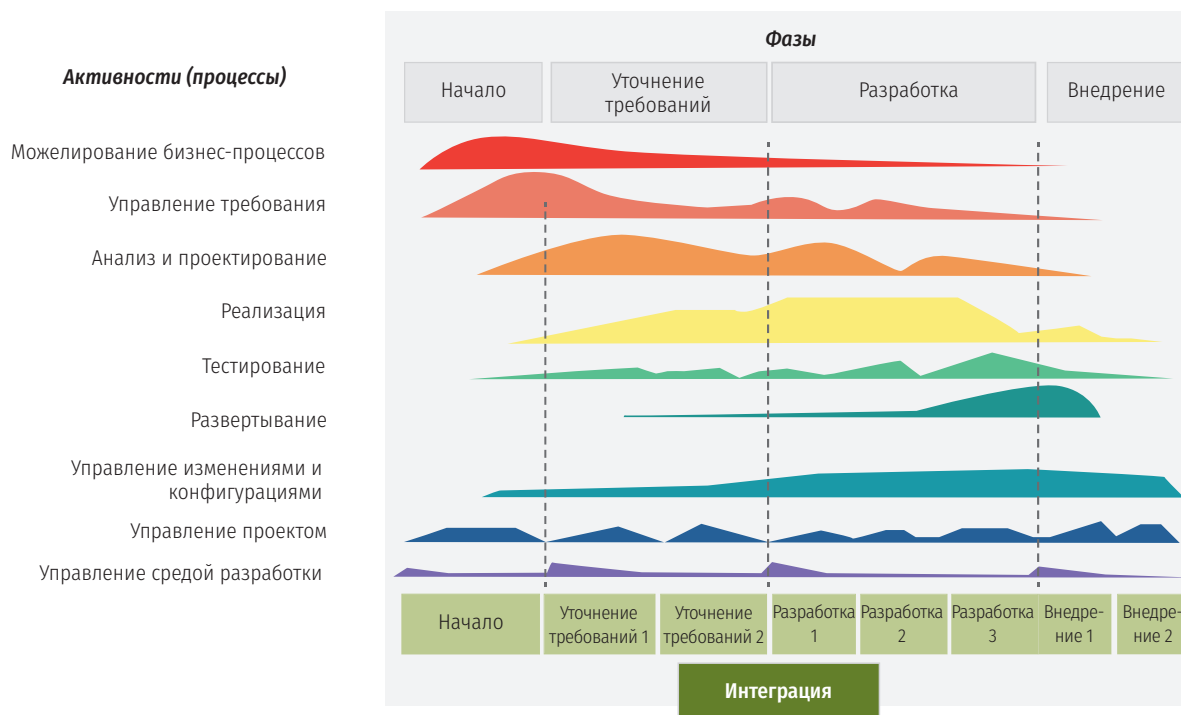
- Начало, инициация (inception);
- Уточнение требований, детальная проработка (elaboration);
- Разработка, создание ПО (construction);
- Внедрение, ввод в эксплуатацию (transition).

Переход с фазы на фазу возможен только после выполнения задач предыдущей фазы и представляет собой контрольную точку проекта. Внутри каждой фазы могут быть несколько итераций, которые увеличивают вероятность достижения задач фазы (на Рис. 2.4.4 они показаны по оси X или оси времени). Кроме того, RUP определяет набор активностей или процессов (в RUP они называются «дисциплины»), которые повторяются в разном объёме на разных фазах и итерациях (на Рис. 2.4.4 они показаны по оси Y):

Основные активности (процессы):

- Моделирование бизнес-процессов;
- Управление требованиями;
- Анализ и проектирование;

Рис. 2.4.4. Цикл разработки в соответствии с методологией RUP.



- Реализация;
- Тестирование;
- Развёртывание.

Вспомогательные активности (процессы):

- Управление изменениями и конфигурациями;
- Управление проектом;
- Управление средой разработки.

Примерный объем активностей на каждой фазе и отдельных итерациях показан на Рис. 2.4.4 (высота «горбов» по вертикальной оси отражает объем работ).

RUP чётко определяет, какие артефакты (документы, результаты) должны быть созданы на каждой фазе. Однако методология не требует, чтобы все они обязательно создавались, тем более, на каждой итерации, и оставляет проектной команде свободу действий в выборе уровня строгости использования методологии.

Кроме того, RUP определяет набор принципов разработки:

- **Раннее определение рисков** и принятие соответствующих мер. Каждая итерация должна начинаться анализом рисков и подготовкой списка мер, устраняющих или смягчающих их. Перечень рисков зависит от фазы.
- **Выполнение требований заказчика.** Требования заказчика описываются Use Case (вариантами использования).
- **Изменчивость.** Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.

Это неотъемлемое свойство всех методологий итеративной и гибкой разработки. Итеративность позволяет как исправлять дефекты, так и отзываться на изменения внешней среды и потребностей заказчика. Управление изменениями является важной дисциплиной RUP.

- **Сборка системы из компонентов.** Компонентная архитектура позволяет вести параллельную разработку, использовать компоненты повторно, т.е. позволить ускорить создание ПО и повысить его качество. Конечно, для реализации этой возможности необходимо приложить дополнительные усилия.
- **Визуальное моделирование.** Использование графических моделей UML для описания требований и проектирования.
- **Качество на всех стадиях.** Все артефакты на всех стадиях предлагается подвергать тщательной экспертизе. Качество ПО предлагается достигать за счёт метода разработки через тестирование (Test Driven Development, TDD) — сначала пишется тест, затем пишется код ПО так, чтобы тест прошёл.
- **Создание работающего ПО.** Ход проекта оценивается по тому, какая часть ПО работоспособна. Несомненные достоинства методологии RUP принесли ему большую популярность среди российских разработчиков. Преимущества, недостатки и возможное применение модели показаны в Табл. 2.4.6.

Табл. 2.4.6. Преимущества, недостатки и возможное применение методологии RUP.

Преимущества	Недостатки	Возможные применения
• Достаточно строгое и подробное описание процесса разработки • Адаптивность, не требуется создание всех артефактов (документов) • Система программных продуктов, существенно облегчающих использование RUP	• Сложность и избыточность • Возможности итераций ограничены рамками фаз • Когда разработчики используют другое ПО, RUP становится и не вполне удобен	• Можно использовать как для сложных распределённых проектов, так и для относительно простых

Спиральная модель

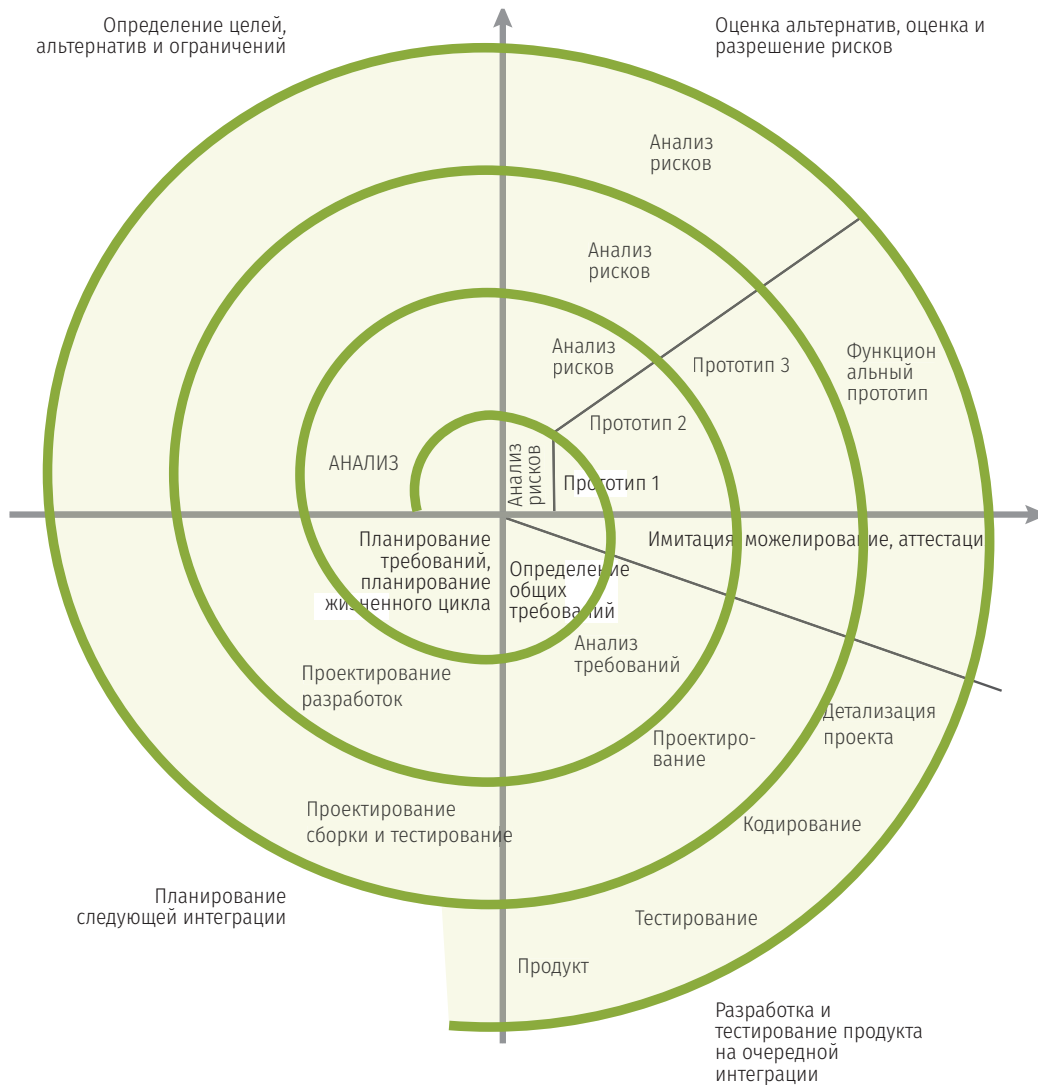
Спиральная модель разработки ПО была создана для обеспечения возможности внесения неоднократных изменений, как в процесс разработки, так и в создаваемый продукт. Процесс разработки разбит на итерации — витки спирали (Рис. 2.4.5).

Рис. 2.4.5. Спиральная модель разработки ПО
Каждый виток разбит на четыре сектора:

- Определение целей каждой итерации, ограничений, вариантов разработки;
- Оценка вариантов с точки зрения целей и ограничений, выявление и смягчение рисков;
- Разработка, начинающаяся с прототипирования, изучения лучших практик;

- Планирование следующего витка, если принято решение о его реализации, или опуститься на нижний для доработки.
- Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для создания нового продукта: планирование, анализ требований, проектирование, кодирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что версия системы (возможно, прототип) готова к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки.

Рис. 2.4.5. Спиральная модель разработки ПО.



Модель сочетает в себе возможности прототипирования и каскадной модели. Отличие спиральной модели от каскадной состоит в многократном возвращении к формулированию требований и повторной разработке. Фактически работы выполняются параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Кроме того, можно переходить на следующий этап, не дожидаясь полного завершения работы на текущем — оставшу-

юся работу можно будет выполнить на следующем витке.

Спиральная модель ориентирована на большие, дорогостоящие и сложные проекты. Особенно, когда бизнес-цели таких проектов могут измениться, но требуется разработка стабильной архитектуры, удовлетворяющей высоким требованиям по нагрузке и устойчивости. Преимущества, недостатки и возможное применение спиральной модели показаны в Табл. 2.4.7.

Табл. 2.4.7. *Преимущества, недостатки и возможное применение спиральной модели.*

Преимущества	Недостатки	Возможные применения
• Сокращение рисков по мере увеличения стоимости проекта • Контроль со стороны руководства • Раннее обнаружение наиболее существенных рисков • Гибкость	• Сложность, модель ориентирована на большие, дорогостоящие и сложные проекты • Необходимость в грамотном управлении проектом, действия разных групп (архитекторов, программистов, тестировщиков) должны быть синхронизированы • Сложность планирования проекта по срокам и бюджету • Необходимость использования специальных инструментов для специфицирования, визуализации архитектуры, создания и документирования	• Проекты, характеризующиеся высокими рисками • Проекты с недостаточно чётко изложенными требованиями или в условиях, когда бизнес цели таких проектов могут измениться • Проекты с недостаточно хорошо проработанной архитектурой • Проекты с потенциальными проблемами с производительностью и базовыми технологиями

Гибкие методологии (Agile Programming)

Однако сегодня мы сталкиваемся со все большей изменчивостью практически во всех областях деятельности. Кроме того, многолетние попытки вытащить из заказчика требования к будущему ПО слишком часто оказывались и оказываются неудачными. Именно поэтому возник такой широкий интерес к гибким (agile) методам разработки ПО, которые направлены на создание ПО итеративно, максимально подстраиваясь под текущие потребности заказчиков, но при этом создавая работающие и работоспособные варианты. Agile – методы теперь активно используют и в других проектах, в частности, особенно популярны они стали в банковском секторе.

Отцами-основателями гибких методологий были сформулированы основные принципы подхода:

- Получение готового к применению ПО уже на ранних стадиях разработки;
- Высший приоритет — удовлетворение потребностей заказчика;

- Пользователи и взаимодействие с ними важнее процессов и инструментария;
- Программное обеспечение важнее подробной документации;
- Сотрудничество с заказчиком важнее переговоров при заключении договора;
- Реагирование на изменения важнее следования плану.

Помимо опоры на итеративную модель разработки у гибких методологий разработки ПО есть ещё две общие особенности:

- большинство гибких методологий нацелены на минимизацию рисков путём сведения разработки к серии коротких циклов, программный продукт готов уже на ранних стадиях, далее происходит его развитие и усовершенствование;
- упор делается на непосредственное общение лицом к лицу заказчиков и разработчиков, ориентированное на динамическое формирование требований в результате постоянного взаимодействия внутри рабочих групп, в рабочую группу

обязательно включаются представители заказчика.

Таким образом, гибкие методологии повышают продуктивность работы разработчиков и улучшают качество ПО.

Методология Scrum

Методология Scrum - это один из видов гибкой разработки ПО, набор принципов, на которых строится процесс командной разработки. Она ориентирована на то, чтобы в небольшие жёстко фиксированные промежутки времени (называемые спринты) предоставлять заказчику работающее ПО с новыми возможностями, для которых определён наибольший приоритет.

Два основных термина Scrum:

- **Бэклог продукта (Product Backlog)** — приоритизированный список требований с оценкой трудозатрат. Обычно он состоит из бизнес-требований, реализация которых приносит конкретную бизнес-ценность.
- **Спринт** — короткий (от 1 до 4 недель) период проведения работ, в конце которого получается законченный функционал.

Основные принципы Scrum:

- Команда является единственным полностью вовлечённым участником разработки и отвечает за результат как единое целое;
- Никто, кроме команды не может вмешиваться в процесс разработки на протяжении спринта;
- Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

Главные роли в Scrum:

- **Scrum Master** — тот, кто ведёт Scrum-митинги и следит, чтобы при этом соблюдались

все принципы Scrum (роль не предполагает ничего, кроме корректного ведения самого Scrum-проекта, руководитель проекта, скорее, относится к Product Owner и не должен являться Scrum Master);

- **Владелец продукта (Product Owner)** человек, который представляет интересы конечных пользователей и других заинтересованных в продукте сторон;
- **Кросс-функциональная команда (Scrum Team)**, состоящая как из разработчиков, так и из тестировщиков, архитекторов, аналитиков и т. д. (при этом размер команды в идеале составляет 7±2 человека).

Процесс разработки в Scrum состоит из следующих действий:

1. Владелец продукта информирует о запросах на выполнение работ, которые должны быть выполнены. Определяется список требований к ПО (product backlog).
 2. В ходе совета по планированию спринта (sprint planning meeting) команда определяет, сколько из желаемого они могут выполнить на будущем спринте, из списка требований выделяется набор функциональности, которая планируется к разработке. Таким образом, определяется запланированный на спринт список заданий (sprint backlog).
 3. Команда выполняет запланированный на спринт список заданий. На протяжении спринта никто не имеет права менять список требований.
 4. В конце каждого спринта проводится его обзор для получения обратной связи от владельца продукта, изменения требований и приоритетов. Успех Scrum'a зависит во многом до деятельности Scrum Master-a. В числе его главных задач и особенностей его деятельности:
- Удалять барьеры между заказчиком и разработчиками так, чтобы заказчик напрямую управлял разработкой;

- Разрешать конфликты, доверять своей команде, делать все возможное, чтобы раскрыть креативность каждого её члена;
- Увеличивать продуктивность команды любыми возможными способами.

Методология Extreme Programming (EX)

Экстремальное программирование (Extreme Programming, XP) — это один из видов гибкой разработки ПО, набор принципов и приёмов, позволяющих повысить продуктивность работы разработчиков и улучшить качество ПО. При создании методологии экстремального программирования были выделены 12 практик и приёмов, сгруппированных в 4 группы:

1. Короткий цикл обратной связи (Fine scale feedback). Как и в других методологиях гибкой разработки, создание продукта происходит путём быстрых итераций в постоянном общении с пользователями.

- **Единая команда** (Whole team) или заказчик всегда рядом. Разработчики и заказчики представляют собой единую команду, заинтересованную в разработке качественного и полезного ПО, и работают в постоянном тесном контакте друг с другом.
- **Игровое планирование** (Planning game). Основная цель «игры в планирование» — быстро сформировать приблизительный план работы и постоянно обновлять его по мере того, как условия задачи становятся всё более чёткими.
- **Разработка через тестирование** (Test driven development). В XP тестирование является важнейшим процессом. Методология выделяет 2 типа тестов: Unit testing — тестирование элементов (иногда называют — модульное тестирование) и Acceptance testing (приёмочные испытания). Первые позволяют убедиться в работоспособности ПО, вторые — в том, что ПО соответствует

потребностям заказчика (требуемой функциональности). XP использует подход разработки через тестирование (Test Driven Development, TDD) — сначала пишется тест, затем пишется код ПО так, чтобы тест прошёл.

- **Парное программирование** (Pair programming). Написанием кода занимается пара разработчиков, которые работают за одним рабочим местом. Один из них собственноручно записывает код, а другой просматривает его работу и подсказывает идеи. Время от времени сотрудники меняются ролями. Участников пар в XP также рекомендуется периодически «тасовать», так что каждый разработчик оказывается в курсе всего проекта.

2. Непрерывный процесс (Continuous process).

- **Небольшие (и частые) релизы** (Small releases). Общая идея гибких методологий проектирования состоит в создании рабочей версии ПО, удовлетворяющей некоторые (основные) потребности заказчика. XP не исключение. Он предлагает «есть слона по частям», и при этом постепенно «насыщаться».
- **Непрерывная интеграция** (Continuous integration). Очень часто в проектах разработки ПО вопросы интеграции откладывают на конец проекта, что приводит к серьёзным проблемам. XP предлагает начинать ставить и решать вопросы интеграции с самого начала проекта, постоянно наращивания и совершенствуя интеграционное ПО.
- **Рефакторинг или модернизация проектного решения** (Refactoring or design improvement). Улучшение кода при сохранении функциональности или рефакторинг является существенным компонентом XP. Код всегда можно улучшить. XP предлагает заниматься этим на протяжении всего проекта.

3. **Общее понимание** (Shared understanding).

- **Простота дизайна** (Simple design). Как все технологии гибкой разработки, XP исходит из того, что требования к ПО постоянно меняются, поэтому грамотно спроектировать систему со всеми подробностями в начале проекта невозможно. Надо двигаться итеративно, соблюдая условие простоты дизайна. Иначе в какой-то момент система окажется неработоспособной.
- **Метафора системы** (архитектура). Метафорой в XP называют по сути архитектуру программной системы. XP требует, чтобы метафора системы была актуальна и постоянно находилась в поле внимания разработчиков. Это очень важный принцип XP.
- **Стандарты кодирования** (Coding standards). При групповой разработке ПО все члены команды должны соблюдать стандарты кодирования. Это позволяет гибко формировать пары для разработки, эффективно проводить рефакторинг, в це-

лом повысить производительность и качество разработки.

- **Коллективная ответственность за код** (Collective code ownership). Т.е. члены команды несут коллективную ответственность за результат разработки: его функциональность и качество.

4. **Благополучие программиста** (Programmer welfare).

- **Устойчивый шаг** (sustainable pace) проекта. На каждой итерации необходимо разрабатывать максимально качественное и работоспособное ПО. Это не полуфабрикат, который невозможно использовать. Кроме того, XP настойчиво рекомендует придерживаться 40-часовой рабочей недели для разработчиков, мотивируя это требование тем, что переработки ухудшают качество разрабатываемого ПО.

Преимущества, недостатки и возможное применение гибких методологий разработки ПО (Scrum и XP) показаны в Табл. 2.4.8.

Табл. 2.4.8. Преимущества, недостатки и возможное применение гибких методологий разработки ПО (Scrum и XP).

Преимущества	Недостатки	Возможные применения
• Максимальная гибкость. • Максимальное соответствие автоматизируемому процессу и функции. • Возможность прекращения проекта после любой фазы. • Относительно невысокая компетенция менеджера проекта. • Возможность сократить затраты и сроки разработки.	• Повышенные риски, невозможность прогнозирования сроков и бюджета проекта. • Ограниченный размер команды проекта, как правило 7±2 человека. • Необходимость тесного взаимодействия команды проекта, в которую входят и представители заказчика. • Необходима активность в коммуникациях и высокая скорость принятия решений заказчиком. • Повышенные требования к компетенции команды проекта.	• Принято считать, что Scrum и XP подходят только для малых и средних проектов. Однако никто не мешает разделить работу над крупным проектом и использовать их и в этом случае. • Географически локализованные проекты. • Проекты, связанные с высокой изменчивостью объекта автоматизации. • Методы гибкой разработки ПО не рекомендуется применять для разработки критически важных систем, сбои которых могут повлечь тяжёлые последствия.

Характеристики качества ПО

Как мы писали выше, одна из проблем разработки ПО – проблема качества. Существует серия стандартов SQuaRE, относящихся к качеству программного обеспечения. В модель SQuaRE входят следующие разделы:

- ИСО/МЭК 2500n — Раздел «Менеджмент качества»;
- ИСО/МЭК 2501n — Раздел «Модель качества»;
- ИСО/МЭК 2502n — Раздел «Измерение качества»;
- ИСО/МЭК 2503n — Раздел «Требования к качеству»;
- ИСО/МЭК 2504n — Раздел «Оценка качества».

Номера с ИСО/МЭК 25050 по ИСО/МЭК 25099 зарезервированы для расширения международных стандартов SQuaRE и/или технических отчётов, буква «n» означает номер стандарта в разделе.

Характеристики качества ПО согласно SQuaRE делятся на определённые области:

1. Функциональность — степень, в которой продукт или система обеспечивают выполнение функции в соответствии с заявленными и подразумеваемыми потребностями при использовании в указанных условиях:

- Функциональная полнота;
- Функциональная корректность;
- Функциональная целесообразность.

2. Уровень производительности — производительность относительно суммы использованных при определённых условиях ресурсов:

- Временные характеристики;
- Использование ресурсов;
- Потенциальная возможность.

3. Совместимость — способность продукта, системы или компонента обмениваться информацией с другими продуктами, системами или компонентами, и/или выполнять требуемые функции при совместном использовании

одних и тех же аппаратных средств или программной среды:

- Сосуществование;
- Интероперабельность.

4. Удобство использования — степень, в которой продукт или система могут быть использованы определёнными пользователями для достижения конкретных целей с эффективностью, результативностью и удовлетворённостью в заданном контексте использования:

- Определимость пригодности;
- Изучаемость;
- Управляемость;
- Защищённость от ошибки пользователя;
- Эстетика пользовательского интерфейса;
- Доступность.

5. Надёжность — степень выполнения системой, продуктом или компонентом определённых функций при указанных условиях в течение установленного промежутка времени:

- Завершённость;
- Готовность;
- Отказоустойчивость;
- Восстанавливаемость.

6. Защищённость — степень защищённости информации и данных, обеспечиваемая продуктом или системой путём ограничения доступа людей, других продуктов или систем к данным в соответствии с типами и уровнями авторизации:

- Конфиденциальность;
- Целостность;
- Неподдельность;
- Отслеживаемость;
- Подлинность.

7. Сопровождаемость — результативность и эффективность, с которыми продукт или система могут быть модифицированы предполагаемыми специалистами по обслуживанию:

- Модульность;
- Возможность многократного использования;
- Анализируемость;
- Модернизируемость;
- Тестируемость.

8. Переносимость — степень простоты эффективного и рационального переноса системы, продукта или компонента из одной среды (аппаратных средств, программного обеспечения, операционных условий или условий использования) в другую:

- Адаптируемость;
- Устанавливаемость;
- Взаимозаменяемость.

Характеристики качества ПО

Тенденции в разработке ПО

Разработка ПО постепенно обогащается различными инструментами, как управленческими, так и техническими, которые позволяют добиться повышения качества кода, облегчить сопровождение и эксплуатацию ПО, улучшить показатели его использования.

Сбор требований к ПО

Одним из важнейших этапов разработки ПО является сбор и описание требований ПО. Именно от качества этого этапа в большей степени зависит качество ПО, его полезность и применимость в организации. В помощь этому процессу существует большое количество нотаций моделирования. Наиболее известны из них:

IDEF — семейство стандартов функционального моделирования сложных программных систем, основанный на методологии SADT (Structured Analysis and Design Technique или методология структурного анализа и проектирования), в 1993 году принятый в качестве федерального стандарта в США, а в 2000 году — в качестве руководящего документа по стандартизации в Российской Федерации. Всего известно 14 ме-

Какой бы стиль ведения проекта не был выбран: классический (например, «водопадный») или одна из гибких методологий, необходимо проанализировать все характеристики и вместе с заказчиком определить их целевые значения. При итеративной разработке эти значения должны быть определены для каждой итерации. Не следует думать, что гибкие методы разработки позволяют небрежнее относиться к качеству. Это одно из заблуждений, которое стоит на пути активного использования этих методов в России. Как раз наоборот, большинство этих методологий ставит качество во главу угла.

тодов IDEF, среди которых наиболее популярны IDEF0 - средство описание функциональной модели организации и IDEF3 для описания бизнес-процессов и рабочих процессов.

ARIS – (Architecture of integrated Information Systems), методология моделирования бизнес-процессов организации, была разработана Августом Шеером в 1992 г. Существует также одноименное семейство программных продуктов, которые покрывают многие потребности разработки ПО, в частности вопросы проектирования архитектуры AC.

UML — (Unified Modeling Language, унифицированный язык моделирования) был разработан консорциумом OMG в 1997 г. для графического моделирования объектно-ориентированного ПО, в частности, как средство для последующей автоматической генерации кода.

Методы анализа программного кода

К числу методов анализа программного кода, о которых необходимо упомянуть, относятся:

- Анализ кода, просмотр кода или рецензирование кода (code review) — анализ программного кода без запуска программы,

выполняемый специалистом или группой специалистов, но не тем(и), который(ые) писал(и) этот код, что позволяет осуществить не просто формальную и статическую обработку кода, но и проанализировать также архитектуру решения, выбранный алгоритм и его оптимальность для поставленной задачи. И не просто улучшить качество отдельной программы, но и в целом обеспечить развитие и улучшение качества разработки, и повышение квалификации разработчиков.

- Статический анализ кода — анализ кода без запуска программы, выполняемый специальным ПО на основании общих принципов и специфики используемого языка программирования.
- Динамический анализ кода — анализ кода путём выполнения программы на реальном или виртуальном процессоре, обычно, с помощью специальных утилит, которые позволяют проверить код на большом количестве входных данных и проанализировать результаты выполнения программы.

Тестирование

Тестирование обычно принято разбивать на следующие виды:

- Регрессионное тестирование — собирательное название для всех видов тестирования ПО, направленных на обнаружение ошибок в уже протестированных участках программного кода;
- Дымовое тестирование — минимальный набор тестов на явные ошибки, проводится при внесении незначительных изменений в код;
- Системное тестирование — выполняется после окончания разработки ПО и направлено на тестирование как отдельных модулей АС, так и взаимосвязей этих модулей;

- Интеграционное тестирование — направлено на тестирование взаимодействия между подсистемами;
- Тестирование сквозного процесса — выполняется после успешного завершения системного и интеграционного тестирования и направлено на тестирование всего автоматизируемого процесса;
- Интеграционно-функциональное тестирование (ИФТ) — состоит из 3-х типов тестирования: системное, интеграционное, тестирование сквозного процесса;
- Нагрузочное тестирование — обычно выполняется после успешного завершения ИФТ и направлено на получение количественных показателей быстродействия АС и обнаружения проблем производительности.
- Тестирование документации — направлено на выявление несоответствий в технической спецификации и требованиях;
- Приёмочное тестирование — выполняется после успешного завершения всех этапов тестирования, процесс проверки соответствия АС заданным требованиям, выполняемый рабочей группой на специально оборудованном испытательном стенде.

Общие тенденции

Все более популярными становятся программные платформы, которые используются в различных функциональных областях: от ERP-систем (Compiere) до создания Help Desk (Remedy). Среди российских программных систем наиболее популярное платформенное решение — это несомненно 1С. Платформенные системы сводят процесс разработки ПО к сборке решения из готовых программных компонентов подходящей функциональности и позволяют существенно сократить, а иногда полностью исключить, объем собственно программирования. Термин программист все чаще заменяется термином разработчик ПО.

С точки зрения управления разработкой ПО доминирование классических последовательных моделей разработки было обусловлено ростом сложности создаваемого ПО и относительной стабильностью объектов автоматизации.

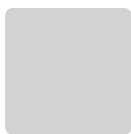
Сейчас наблюдается увеличение внимания к гибким методикам разработки ПО Agile Programming, что обусловлено ростом изменчивости и неопределённости среды, в кото-

рой работают многие организации. Однако, ни одна модель не будет полностью доминирующей на рынке и, вероятнее всего, интерес к методологиям и моделям будет развиваться по спирали. Это значит в будущем нас, возможно, ждёт возвращение к классическим подходам на каком-то новом уровне или совершенно новые методы, которые будут органично сочетать гибкие и классические методы разработки ПО.

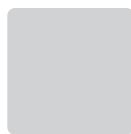
Часть 2. Управление ИТ деятельностью

Глава 2.5

Управление отношениями



Павел
Пестряков



Алексей
Часников

Всегда оставайтесь самим собой, по крайней мере, это будет понятно тем, с кем вы поддерживаете отношения.

Отношения. Что это? Часто мы слышим «у меня отличные отношения с...», «у него не сложились отношения». Насколько отношения с окружающими влияют на нашу профессиональную жизнь? Как это происходит, и что необходимо делать, чтобы сделать отношения с людьми и организациями своими помощниками, а не противниками в жизни и работе?

Систематизация отношений

Для начала попробуем систематизировать отношения, в которые вступает CIO. Понятно, что в своей профессиональной деятельности CIO необходимо руководствоваться интересами организации. Организация, в свою очередь, сталкивается с окружением, которое влияет на деловой результат. В теории управления принято разделять сферы взаимоотношений на внутреннюю и внешнюю. Во внешней сфере

Отношения и человеческие коммуникации — основные навыки в деятельности менеджера и руководителя. Развитие этих навыков — основа успеха в профессиональной и личной жизни. Постараемся разобраться в сути предмета и дать практические рекомендации.

Социальный психолог Дерил Бем определяет отношения как «то, что нравится, и то, что не нравится, как нашу неприязнь или привязанность к предметам, людям группам или к любым аспектам окружающей нас среды»¹.

организация, взаимодействует с четырьмя основными группами, которые оказывают влияние на деловой результат.

Для определения внешнего окружения, в котором работает любая организация, и CIO, как один из представителей её высшего, часто используются названия на английском языке (и их сокращения) для обозначения областей управления отношений:

¹ Daril J.Bem, Beliefs, Attitudes and Human Affairs (Monterey California; Dooks/Cole, 1970) p. 14.

- с общественностью — Public Relations (PR);
- с прессой — Media Relations (MR);
- с инвесторами — Investor Relations (IR);
- с государственными и регулирующими органами — Government Relations (GR);
- с потребителями — Customer Relations Management (CRM);
- с поставщиками — Supplier Relations Management (SRM);

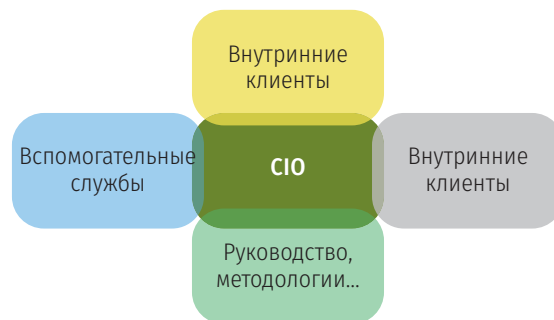
Для управления отношениями с конкурентами устоявшегося термина нет, но всегда нужно помнить, что «вопреки бытующему мнению, не гиены пытаются овладеть добычей львов, а львы часто отнимают добычу у гиен». На построение структуры отношений внутри организации существенно влияют организационная форма, сфера её деятельности, распределение функциональных зон ответственности между подразделениями, стиль управления высших руководителей, уровень корпоративной культуры, размеры, территориальная распределённость организации, и ряд других факторов.

Если проводить аналогию со структурой внешнего окружения, то для подразделений, которыми руководит CIO, также можно условно выделить четыре основных группы, с которыми ИТ-специалисты выстраивают и поддерживают отношения (Рис. 2.5.1).

Условность такого разделения обусловлена тем, что границы между этими группами не всегда могут быть чётко определены. Так, внутренними клиентами – пользователями ИТ-услуг и информационных систем являются как сотрудники основных бизнес-подразделений (например, производство или продажи), так и сотрудники вспомогательных служб (например, бухгалтерия или управление персоналом) и руководство организации.

Бухгалтерия и кадры, в свою очередь являются для CIO поставщиками специфических услуг

Табл. 2.5.1. Внутреннее окружение CIO и его подразделений.



по бухгалтерскому учёту и управлению кадрами. Подразделение закупок и складские службы также могут быть внутренним поставщиками услуг по доставке и хранению ИТ-оборудования. И даже внутри крупных ИТ-подразделений CIO вовлечён в отношения между службой ИТ-инфраструктуры и ЦОД, которые являются поставщиками услуг по предоставлению и обслуживанию вычислительных мощностей для служб поддержки и сопровождения информационных систем. Ограничения и требования, накладываемые высшим руководством организации, владельцами бизнес-процессов, методологами порой становятся не менее приоритетными, чем, например, задачи соблюдения закона о персональных данных, при этом с коммуникации с регуляторами по ФЗ 153 происходят куда реже, чем с руководителями организации. Со всеми этими группами, CIO требуется выстроить понятные и эффективные профессиональные отношения, которые будут обеспечивать как решение целей и задач бизнеса, так и ИТ/ИС служб.

«Трудности возникают из-за несоответствия реальности ожиданиям — бизнес рассчитывает на получение ИТ-услуг без лишней суеты, непосредственное участие ИТ в повышении эффективности бизнеса и реальный стратегический вклад в ИТ в бизнес»².

² Терри Уайт «Чего хочет бизнес от ИТ». Минск. Гревцов Пабlishер. 2007.

Проблема внутренней конкуренции всегда будет актуальна для СІО крупного холдинга, в котором не выстроена вертикаль управления ИТ/ИС-службами. Обособленные отношения между руководителем дочерней организации и локальным ИТ директором могут порой идти вразрез с корпоративными целями и программой развития. И, как следствие, негативно отражаться на отношениях между СІО холдинга и локальным ИТ директором. При этом практика корпоративных ИТ-проектов показывает, что если первый руководитель дочерней организации имеет влияние на основного акционера, то интересы локального ИТ-директора могут свести на нет разумные инициативы СІО головного офиса, направленные на повышение эффективности ИТ-решений в рамках всей организации.

И здесь мы подходим к самой сути успеха любых отношений: наличие или отсутствие стремления достичь общей цели, доверия и взаимопонимания между ИТ-руководителем и его окружением.

Какие же факторы влияют на успех выстраивания отношений. Перечислим несколько основных:

- Знания и навыки, которыми обладает СІО;
- Тип личности и стиль руководства (авторитарный, демократический, либеральный);
- Уровень корпоративной культуры в организации;
- Язык общения;
- Позиция и роль СІО в организации.

Знания и навыки: для темы отношений одним из ключевых является навык ведения переговоров. Этой теме посвящено немало публикаций. Так, например, Пол Т. Бриз и Том Бризор определяют переговоры как «процесс, в ходе которого стороны продвигаются от их первоначально противоположных позиций к такому положению, когда может быть достигнуто соглашение».

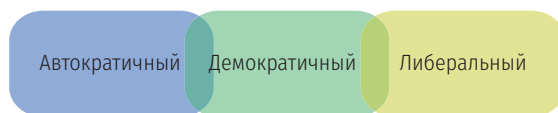
Знание типа личности и стиля руководства позволяет более аккуратно выстраивать

отношения как с представителями внешнего, так и внутреннего окружения. «По традиционной системе классификации стиль может быть автократичным (это одна крайность) и либеральным (другая крайность) или это будет стиль, сосредоточенный на работе, и стиль, сосредоточенный на человеке».

В организации, где доминирует демократический стиль, «руководитель тратит сравнительно большую часть своего времени, действуя как связующее звено, обеспечивая соответствие целей производственной группы целям организации в целом, заботясь о том, чтобы группа получила необходимые ей ресурсы».

Рис. 2.5.2. Автократично-либеральный континуум стилей руководства

Рис. 2.5.2. Автократично-либеральный континуум стилей руководства.



Корпоративная культура оказывает существенное влияние на построение взаимоотношений с не только с внутренним окружением СІО, и его подчинёнными, но в ряде случаев и с представителями внешнего окружения. Так, например, отсутствие в организации проектной культуры существенно ограничивает возможности СІО правильно организовать и привлечь ресурсы для внедрения информационной системы, и более того, создаёт дополнительные проблемы для поставщиков услуг внедрения. В частности, за счёт изменения содержания проекта, переноса сроков и т.д.

Законодательство РФ регламентирует достаточно большую часть отношений СІО с представителями внешнего и внутреннего окружения. В первую очередь это касается сферы закупок, и соответственно отношений между поставщиками и заказчиками. В РФ закупочная деятельность регламентируется двумя основными законами: 44-ФЗ и 223-ФЗ,

а также целым рядом соответствующих подзаконных актов обязательных к исполнению госструктурами, госкомпаниями, организациями с более чем 50% участием государства, естественными монополиями, а также предприятиями в регулируемых видах деятельности. Существенные ограничения накладываются также на деятельность ИТ в банках.

Законодательство в отношении результатов развития высоких технологий изменяется гораздо медленнее, медленно, чем возникают новые технические возможности. При этом регуляторы периодически накладывают различные требования, которые вынуждают CIO с осторожностью применять последние достижения.

В этой связи, чтобы не «подставить» свою организацию в отношениях с законодательством CIO должен не забывать поддерживать отношения с юридическими службами или юрисконсультантами своего работодателя согласовывать с ними действия касающиеся отношений с внешними организациями.

Язык общения: Эффективность отношений существенно зависит от того, на каком языке CIO общается со своим окружением. Наряду со знанием основных положений законодательства, терминология «лучших практик» управления ИТ, позволяет CIO найти понимание не только с представителями внешнего окружения организации (например, поставщиками, аудиторами или представителями регулирующих органов), но и с внутренним окружением, в первую очередь с руководством и собственниками. Так, например, использование рекомендуемых «лучшими практиками» согласованных показателей эффективности (или метрик для управления ИТ-Услугами) позволяет выбрать необходимый набор параметров деятельности ИТ, которые будут понятны собственнику и вышестоящему руководителю, оценить предоставляемые ИТ-услуги в рамках соглашения с бизнесом об уровне обслуживания (SLA – Service Level Agreement).

«Важно понимать какие задачи ставит перед собой бизнес и согласовывать ними процессы измерения, мониторинга и контроля»

Но при этом основным языком общения с бизнесом остаётся язык денег. Именно он позволяет исключить претензии бизнеса и других вспомогательных подразделений, о том, что ИТ говорит на своём «птичьём» языке, позволяет оцифровать и сервисы ИТ, и инвестиции, и затраты, необходимые для предоставления таких услуг.

Позиция и роль: Кроме того, что CIO — это должность в корпоративной структуре с соответствующими функциями, обязанностями и правами. Необходимо помнить, что эту должность занимает, как правило, человек, обладающий неотъемлемыми для руководителя личными качествами в умении построения и поддержки отношений с различными группами внешнего и внутреннего окружения.

На практике очень важно понимать, что движет людьми в той или иной ситуации, какие приоритеты и ценности лежат в основе того или иного поступка. Поэтому с точки зрения основных приоритетов и ценностей предлагаем также разделять следующие виды отношений:

- Деловые (профессиональные, корпоративные, контрактные) отношения;
- Личные отношения (дружеские, семейные) отношения.

Настоятельно рекомендуется при выполнении своих должностных обязанностей не путать (не совмещать) деловые и личные (в первую очередь семейные и дружеские) отношения. Следует придерживаться принципа: «дружба – дружбой, а служба – службой». Подмена понятий приводит к неестественности, последующей напряжённости и может привести к серьёзным конфликтам. Безусловно, если взаимное проникновение деловых и дружеских отношений не происходит естественным образом. В этом случае создаётся очень комфортная среда для работы. Однако при этом надо всегда помнить, что причина, а что след-

ствие: отношения нужны для работы, а не работа — для отношений. Поэтому ответ на вопрос, сокращать или нет дистанцию в деловых отношениях, надо всегда давать не торопясь, чётко понимая возможные негативные последствия. Для организаций, ценные бумаги которых размещаются на зарубежных биржах, могут существовать специальные требования, разделяющие деловые и семейные отношения. Так, например, закон Сарбейнса-Оксли (англ. Sarbanes-Oxley Act, SOX), помимо серьёзных требований к системам внутреннего контроля и, соответственно, к аудиту и организации ИТ-деятельности и ИТ-систем, существенно ограничивает возможности членов семей занимать руководящие должности в таких организациях.

Наконец, ещё один параметр для систематизации отношений — их качество. В шкале «негатив-позитив» можно выделить такие типы отношений:

- хорошие – плохие;
- устойчивые – минутные, случайные;
- доверительные – подозрительные, отстранённые (не доверительные);

Советы здесь простые — стройте отношения на позитивных основах. Относитесь к окружающим так, как вы хотели бы, чтобы они относились к вам. Всячески избегайте недоверия в отношениях — вам должны доверять. При малейшем проявлении недоверия постарайтесь разобраться в его сути и причинах и приложите усилия для его устранения.

Решение конфликта вокруг ИТ — пример вовлечения

Рассмотрим крупную компанию, деятельность которой связана с торговлей. Компания занимается логистикой и обладает своей автоматизированной системой. Коммерческий директор привёл некую компанию, которая, по его мнению, лучше всех может выполнить работу по автоматизации процесса его деятельности. В прошлом он с ней успешно работал, и достиг, по его словам, выдающихся результатов. Однако эксперты ИТ-департамента и ИТ-директор уверены, что предлагаемая система по ряду технических причин непригодна и, к тому же, устарела. Налицо конфликт. Как сделать так, чтобы коммерческий директор решил свою задачу и остался доволен, а ИТ-директор и его сотрудники с удовольствием принимали участие в этом проекте?

Основные шаги, в данном случае могут быть следующими:

- 1.** Несмотря на потенциальные проблемы, ИТ-директор соглашается на данный проект, тем самым постепенно располагая к себе и к своим решениям коммерческого директора. Спорить с коммерческим директором в такой ситуации — значит нажить себе врага. С коммерческим директором необходимо договориться о поддержке в процессе внедрения данной системы.
- 2.** ИТ-директору следует придерживаться следующей стратегии: мы внедряем автоматизированную систему, но с необходимыми для нас доработками со стороны исполнителя.
- 3.** Далее ИТ-директору необходимо оценить необходимые ресурсы (временные, денежные, человеческие) для внедрения системы. При этом в оценку надо заложить максимальное количество необходимых улучшений и доработок системы.
- 4.** Все предполагаемые решения необходимо вынести на совет директоров (правление). Перед этим желательно вынести этот вопрос на заседание комитета по ИТ с обязательным приглашением коммерческого директора для участия в нем. Вовлечение коммерческого директора в работу комитета по ИТ повысит его лояльность к предлагаемым ИТ-директором решениям, и их проще будет согласовать. Если комитета по ИТ в организации нет, необходимо его создать и проводить регулярно.

5. Чтобы повысить лояльность сотрудников по отношению к внедряемой системе, необходимо мотивировать их за счёт средств проекта, объяснить, что мы существенно дорабатываем и расширяем систему, а также вовлечь их в процесс написания технических заданий на доработку системы.

Однако в ряде случаев CIO стоит отказать коммерческому директору, даже ценой конфликта, — если предлагаемая коммерческим директором система радикально не соответствует перспективной ИТ-архитектуре организации и может нанести вред как самой организации, так и профессиональной репутации CIO, а также в случае наличия серьёзных политических оснований.

Профессиональное окружение CIO

Давайте более подробно рассмотрим первую систематизацию отношений — по профессиональному окружению.

Коллеги «по цеху»: Люди, занимающиеся одинаковой профессиональной деятельностью, исторически склонны объединяться в сообщества по профессиональному принципу. Сейчас во время взрывного роста числа социальных сетей объединение профессионалов происходит на клубной (по территориальному принципу) и профсоюзно-союзной основе. Клубы и союзы полезны для обмена опытом, профессиональных коммуникаций, да и просто личной дружбы (подробнее об отношениях в рамках клуба ИТ-директоров см. в конце Учебника «Участие в ИТ-сообществе»). Наши рекомендации в построении отношений в остальных профессиональных группах ниже.

Отношения с подчинёнными. Создавайте рабочую, дружескую атмосферу в коллективе, приучайте сотрудников к конструктивному общению. Использование практик «управляемого конфликта» и прочих методов манипуляции с подчинёнными ведёт к повышению непроизводительных расходов основного ресурса — людей, имеющих в вашем распоряжении, что понижает эффективность работы подразделения под вашим руководством. Существенным фактором выстраивания отношений с подчинёнными является соблюдение психологического контракта, который

можно определить, как набор взаимных не прописанных в каком-либо документе ожиданий, существующих между работниками и работодателем. Психологический контракт подразумевает, что существует постоянно действующий набор ожиданий как со стороны работодателя (в нашем случае CIO), так и работника (ИТ-специалиста), который затрагивает предположения, ожидания, обещания и взаимные обязательства. Нарушения психологического контракта, как правило являются причиной конфликтов и увольнений поскольку дают ответы на два ключевых вопроса трудовых отношений, которые волнуют каждую из сторон: «Чего я могу ожидать от организации/от работника» и «Какой вклад и в каком объёме ожидается взамен?».

Отношения с регуляторами (чиновниками).

Мы рекомендуем исключительно деловые отношения с регуляторами. Чиновников надо обеспечивать необходимым объёмом информации, позволяющим выполнять их работу. Однако не более того. Необходимо помнить, что ответственность за результат несёт не регулятор, а управленческая команда организации. Поэтому во взаимоотношениях с чиновниками необходимо иметь свою конструктивную и обоснованную позицию, если вопрос касается интересов организации.

Отношения с коллегами-руководителями в подразделениях организации. Выстраивание отношений с коллегами, основанных на дове-

Рис. 2.5.3. Профессиональное окружение CIO.



рии и взаимопонимании — одна из основных задач CIO. Для этого используются различные методы, часть из которых описана в главе 1.1 «Профессия CIO». Основная цель выстраивания отношений с коллегами — поддержка проектов и работа с внутренними заказчиками. Их надо вовлекать в совместную работу как равноправных партнёров.

Самое страшное в этих отношениях — появление внутреннего конкурента ИТ-службы. Часто выделение различных направлений и разделение на управленческое, производственное, развитие и сопровождение без общего квалифицированного управления приводит к серьёзной деградации функции или неэффективному использованию ИТ. Отметим, что вместе с низким уровнем корпоративной культуры это может стать началом конца бизнеса особенно на высоко конкурентных, розничных рынках. Эти же причины могут привести и к появлению внутреннего конкурента ИТ-службы — ИТ-систем и ИТ-ре-

шений, построенных без контроля и участия ИТ-службы (часто такое явление называют «теневое ИТ», shadow IT). Надо всячески избегать такого сценария — борьба с внутренним конкурентом ИТ-службы может отнимать колоссальные ресурсы, которые в этот момент используются непродуктивно. Второй важный момент в отношениях с коллегами — проблема доминирования. Финансовые, технические директора, руководители службы безопасности и прочие руководители часто считают работу CIO простой и пытаются либо занять позицию сверху по отношению к CIO в иерархической структуре организации, либо перетянуть на себя часть функциональных обязанностей вместе с ресурсами. Особенно это касается отдельных контуров управления финансами (контроллингом) или производственной автоматизацией уровня MES и АСУТП.

Что делать в таких случаях? Во-первых, не ввязывайтесь в прямой конфликт. Докажите свою состоятельность и профессионализм через достижение быстрых результатов. Quick wins (быстрые победы) — это лучший «ответ» CIO. Ищите узкие места в бизнесе, «расшивайте» их с использованием современных технологий, помогайте бизнесу достигать бизнес-результатов, не тратя время и ресурсы на склоки внутри организации. Формируйте свой образ как профессионального менеджера и специалиста в области управления организацией, а не «хитрого политикана», искушённого в корпоративной борьбе.

Во-вторых, если ваш коллега уже совсем «распоясался» — можно сделать тактический шаг назад. Отдайте на время область ответственности и ресурсы. Ведь в 90% случаев тот, кто на них претендует, слабо разбирается в предмете. У каждого должен быть шанс провалиться — пусть он поруководит, раз очень хочет. Мы с вами знаем, что это совсем

не так просто, как кажется. Зато потом, когда все увидят результаты его управления, можно будет вернуть все назад.

Отношения с акционерами, инвесторами и руководителями. Как правило, акционеры и руководители организации требуют прозрачных целей использования ИТ, синхронизированных с стратегическими целями бизнеса и демонстрации ценности, которую использование информационных технологий приносит организации. О методах построения адекватной ИТ-стратегии рассказывается в главе «ИТ-стратегия» полной версии Учебника. О подходах к демонстрации ценности ИТ мы рассказали в главе 1.3 «Бизнес-ценность ИТ».

Основой для общения на уровне руководства организации и собственников бизнеса может быть вовлечение CIO в работу структур операционного и стратегического управления бизнесом. Участие в стратегических сессиях, правлении, комитетах и прочих органах управления организацией, позволяет получить CIO ясную картину целей, методов и потребностей бизнеса и сформировать у бизнеса потребность в использовании информационных технологий. Для систематизации отношений с акционерами и руководи-

телями очень полезным будет ознакомление с рекомендациями COBIT 4.1, который систематизирует права и обязанности участников системы корпоративного управления по использованию информационных технологий и особенно с документом Board Briefing on IT Governance (об этом подробнее рассказано в главе «Корпоративное управление ИТ» полной версии Учебника).

Основное отличие акционеров и инвесторов от генеральных директоров: генеральный директор и акционеры стоят по разные стороны от собственности. Убытки для акционеров — расставание с собственными деньгами, в то время как для генерального директора (как и для CIO) — лишь карьерная неудача. Поэтому с генеральным директором и другими менеджерами часто проще договариваться в вопросах изменения сроков и стоимости проектов, в случае ресурсного конфликта и т. п. Наконец, отметим, что для построения отношений CIO с акционерами и руководителями организации, важно построение системы финансового планирования и отчётности, совместимой с основной финансовой системой, используемой в бизнесе (об этом рассказано в главе 3.2 «Управление финансами»).

Управление отношениями с поставщиками и вендорами

Типы и виды поставщиков

Если коротко по Ожегову, то «поставщик — лицо, организация, поставляющие какие-нибудь материалы, товары». Поставщиком может быть как внешняя организация, предоставляющая товары, работы, услуги, в соответствии с заключённым договором, так и внутреннее подразделение организации (например, Служба ИТ-инфраструктуры, предоставляющая услуги поддержки серверов).

Заметим, что в качестве ИТ поставщика может также выступать физическое лицо (фри-

лансер, внештатный работник), которого организация нанимает только для выполнения определённого перечня работ без заключения долгосрочного трудового договора.

Вендор (англ. vendor) — это поставщик брэнд-продуктов и услуг, выпускающий их под своей торговой маркой. В ИТ-среде термин «вендор» как правило, применяется для идентификации организации — разработчика оборудования и программного обеспечения, отличая тем самым от его партнёров, которые зачастую выступают в роли реселлеров (поставщиков) программ-

ного обеспечения. Стоит отметить, что вендор — это не всегда непосредственный производитель продукта, эту особенность можно увидеть в модели работы организаций, выпускающих ПО с открытым кодом.

Существует также термин «независимый вендор программного обеспечения», (Independent Software Vendor, ISV) который применяется к организациям, специализирующихся на создании или продаже программного обеспечения.

Value-added reseller (VAR) — это организация, которая модифицирует/расширяет возможности уже существующего продукта (т. е. создаёт добавочную стоимость), а затем перепродает его, извлекая прибыль из создаваемой добавленной стоимости. Например, термин VAR широко используется в качестве определения производителей компьютеров, которые закупают отдельные комплектующие и на их основе собирают готовые компьютеры.

На российском рынке распространён термин **«Системный интегратор»**. Он охватывает более широкий круг организаций — поставщиков комплекса профессиональных ИТ-услуг, таких как системная интеграция, консалтинг, обучение и внедрение бизнес-приложений, построения ИТ-инфраструктуры, в том числе включая поставку оборудования и ПО.

Аутсорсинговая компания — это специализированный поставщик услуг, который исполняет **на основании договора** определённые виды или функции производственной предпринимательской деятельности **другой компании (Заказчика)**. Подробнее об аутсорсинге смотрите в главе 3.4 «ИТ-аутсорсинг».

Роли и типы ИТ-поставщиков

Перед тем, как говорить об общих принципах построения взаимоотношений с поставщиками, полезно посмотреть, в каких ролях предстают перед заказчиками ИТ-поставщики. ИТ-поставщика можно рассматривать как:

- магазин, в который приходят купить пусть дорогие, но необходимые товары;

- сервисную структуру, которая оказывает организации определённые услуги;
- собрание компетенций и экспертизы, которое можно использовать в случае недостатка компетенций внутри организации;
- исследовательскую лабораторию, которая может разобраться со сложной проблемой, беспокоящей организацию;
- бизнес-партнёра, с которым нужно выстраивать отношения не в плоскости поставщик-потребитель, а как с организацией, которая вместе с вами работает над единой задачей.

В зависимости от того, в какой роли выступает перед вами ИТ-поставщик, по-разному будут выстраиваться отношения с ним. В одном случае CIO может и даже должен жёстко требовать от поставщика соблюдения простых правил работы (например, когда он выступает как «магазин»), а в другом случае выстраиваются отношения доверия равных партнёров. Дело в том, что поставляемые продукты и услуги удобно разделять на два типа:

- закупки простых продуктов (типовое оборудование, простое ПО) и в некоторых видах простых услуг;
- приобретение сложных продуктов и комплексных услуг (внедрение новой или существенная модернизация текущей информационной системы, либо комплексный инфраструктурный проект).

При поставках этих двух типов поставщики выступают в различной роли, и взаимоотношения должны выстраиваться по-разному. Поэтому удобно аналогично делить ИТ-поставщиков.

ИТ-поставщик типа 1 — это поставщик простых продуктов (типовое оборудование, простое ПО) и в некоторых видов простейших услуг. При поставке таких продуктов и услуг от поставщика требуется минимальный уровень собственной экспертизы в области ИТ, и создаваемая поставщиком при такой рабо-

Поставщики Двора Его Императорского Величества

РИдея отмечать качество российских товаров особым знаком принадлежала Петру I. Именно при его правлении продукцию уральских заводчиков Демидовых впервые в стране стали клеймить «Знаком соболя». С 1824 года предприятия, получившие право постоянно доставлять свою продукцию к императорскому двору, получали диплом и звание «Поставщик Двора Его Императорского Величества», а также могли украшать свои торговые щиты императорским гербом. А в 1856 году был учреждён знак «Поставщик Двора Его Императорского Величества». Столь высокое звание промышленникам и купцам присваивал сам император «за состояние производства и влияние на жизнь страны», а их товарам — «за весьма чистую отделку, новейший фасон, доступные цены».

Для того чтобы получить право ставить на этикетках и упаковках знак с гербом Российской империи и надписью: «Поставщик Двора Его Императорского Величества», претенденты должны были в течение минимум восьми лет участвовать во всех губернских выставках, утверждённых Министерством финансов России и императором лично, попадать в официальный «Список экспонатов, удостоенных Похвальной грамоты».

те добавочная стоимость находится в области логистики и понимания того, как работают вендоры.

1. ИТ-поставщик типа 2 — это поставщик сложных продуктов и комплексных услуг. Такие поставщики интересны прежде всего своей собственной экспертизой в области ИТ и уровнем предоставляемых услуг.

Заметим, что это не столько характери-

ки самих поставщиков, сколько поставляемых продуктов и услуг. В зависимости от типа поставляемых продуктов и услуг один и тот же ИТ-поставщик может выступать как поставщик типа 1 и как поставщик типа 2. В Табл. 2.5.1 показано, каким образом надо проводить разделение поставщиков на типы. Тем не менее, каждый CIO должен сам проводить такое разделение в зависимости от конкретной ситуации.

Табл. 2.5.1. Критерии разделения поставщиков на типы.

Критерий	Скорее, поставщик типа 1	Скорее, поставщик типа 2
Что?	Простые и средней сложности продукты, простые услуги	Сложные продукты и большинство услуг
От кого?	Поставка от производителя оборудования	Поставка производителем ПО или поставщиком услуг
Сколько?	Поставка на небольшие суммы	Поставка на значительные суммы
Время?	Срок поставки – несколько месяцев	Срок поставки от полугода и более

Для оптимизации работы с поставщиками продуктов и сервисов желательно принимать во внимание правило Парето.

Применяя это правило к задачам управления поставщиками, мы можем логично предположить, что на долю 20% поставщиков должно приходиться примерно 80% от общего объёма расходов или поставок. Надо понимать, что «80/20» — это не магическая формула и фактическое соотношение очень редко составляет точно 80/20.

С организациями, входящими в состав 20% поставщиков, как правило, устанавливают долгосрочные партнёрские взаимоотношения, и они становятся стратегическими поставщиками. Чаще всего организация-стратегический партнёр выступает в роли генерального подрядчика, передавая другим организациям выполнение работ по частным задачам и обеспечивая их контроль в рамках договоров субподряда.

Какую — моно- или мультивендорную политику нужно вести? Основной совет — в большинстве случаев мультивендорность более предпочтительна, так как это позволяет выбрать оптимального партнёра, при этом не переплачивая вендору за оборудование или услугу.

Но в каких-то случаях правильнее использовать моновендорный подход. Моновендорный подход могут использовать небольшие и средние организации, но для крупной организации более предпочтительна мультивендорность, так как при выборе политики моновендорности есть риск попасть в зависимость от вендора — редкий вендор не воспользуется своим монопольным положением. Поэтому, как правило, моновендорный подход ведёт к увеличению стоимости создаваемого решения.

Такой подход можно использовать в тех случаях, когда функционал и специфические черты продуктов важнее, чем их стоимость.

С другой стороны, при моновендорном подходе, если поддержка решения организована собственными силами, она, как правило, стоит меньше. Конкретный выбор моно- или мультивендорной политики диктуют развитие технологий и экономика.

Стандарт качества ISO-9000

Некоторые организации требуют от своих поставщиков наличие сертификатов на соответствие стандартам ISO-9001 или 9002. Сертификат, как правило, служит подтверждением того, что поставщик располагает соответствующей системой качества, эффективность которой регулярно оценивается независимыми аудиторами. «Система обеспечения качества, соответствующая стандарту ISO, гарантирует следующее:

- поставщик предпринимает меры по обеспечению качества, согласованного с заказчиками;
- руководство регулярно оценивает работу системы обеспечения качества и, по мере необходимости, использует результаты внутреннего аудита для улучшения её функционирования;
- процедуры работы поставщика задокументированы и переданы тем лицам, кто зависит от них;
- претензии заказчика регистрируются, рассматриваются в течение разумного срока и, по мере возможности, принимаются во внимание при усовершенствовании услуг;
- поставщик контролирует производственные процессы и может их совершенствовать.

Сертификат ISO не является абсолютной гарантией качества предоставляемых услуг, тем не менее, он служит показателем серьёзного отношения поставщика к вопросам обеспечения качества и его готовности обсуждать эти вопросы».

Принципы построения взаимоотношений с ИТ-поставщиками

В соответствии с рекомендациями ITIL, «цель управления поставщиками и предоставляемыми ими сервисами состоит в обеспечении непрерывного уровня качества ИТ-сервисов для бизнеса».

Согласно ITIL, «процесс управления поставщиками должен гарантировать, что каждый поставщик ясно понимает свои обязательства как поставщика услуг. Требования должны быть ясно определены и согласованы. Также должны быть обеспечены гарантии того, что все изменения этих соглашений производятся в соответствии с процессом управления изменениями».

Однако, необходимо смотреть на этот вопрос несколько шире рамок ITIL. Для CIO цель управления поставщиками состоит также и в том, чтобы упростить и сделать более эффективными не только процессы взаимодействия между ИТ-подразделением и его поставщиками, но и логично встроить разнообразные услуги и продукты поставщиков и вендоров в единую схему процессов предоставления качественных услуг бизнес-подразделениям и при этом оптимизировать затраты организации и стоимость услуг ИТ-подразделения для внутренних клиентов.

В целом, при построении взаимоотношений с ИТ-поставщиками необходимо опираться на четыре общих принципа.

1. Принцип оптимизации политик взаимодействия. Разрабатывая или применяя общие политики взаимодействия с поставщиками, желательно постараться избежать при этом типичной ошибки: «стремления работать со всеми поставщиками одинаково». Степень универсальности и регламентация корпоративных политик по работе с поставщиками должны учитывать особенности ИТ-продуктов и услуг. Совершенно необходимы различные политики для поставщиков типов 1 и 2. Кроме того, жёсткая регламентация процесса не

позволяет адекватно реагировать на требования бизнеса, ставит барьеры на обеспечении своевременного и качественного выполнения услуг ИТ-подразделением.

2. Принцип взаимного баланса интересов заказчика и поставщика (в наибольшей степени используется для ИТ-поставщиков типа 2). Взаимоотношения надо строить на принципе обоюдной заинтересованности сторон. Важно помнить, что выгодные условия — это понятие, работающее в обе стороны. При этом, баланс должен быть достигнут не только и не столько по деньгам, сколько по другим результатам сотрудничества — интересности и нарабатываемом опыте и компетенций, референса и т. д. Кроме того, баланс должен быть найден не только на уровне денег. У поставщика должен быть интерес к проекту не только с точки зрения денег, но и с точки зрения других результатов — получаемых компетенций, референса и т. д.

Самое главное во взаимоотношениях с поставщиками и вендорами — быть весомым заказчиком для них. Если вы весомы, значит, вы правильно построили взаимоотношения со своим партнёром.

Павел Пестряков

Этим можно и нужно пользоваться при построении взаимоотношений не в меньшей степени, чем денежными инструментами.

Отношения с ИТ-поставщиком типа 1 носят, как правило, краткосрочный характер и должны строиться на принципе чёткой формализации правил совместной работы. Здесь взаимодействие с поставщиком короткое, и не нужно выстраивать каких-то специальных отношений.

Отношения с ИТ-поставщиком типа 2 сложнее. Здесь чёткая формализация правил со-

вместной работы также очень важна, однако надо понимать, что в случае сложной поставки или предоставления ИТ-услуг формализовать удастся далеко не все. Чем более сложные задачи решаются (от построения ИТ-инфраструктуры, до автоматизации бизнес-процессов), тем с большим уровнем неопределённости и количеством случайных факторов придётся столкнуться ИТ-директору. Поэтому, в случае таких взаимоотношений, помимо более серьёзных проверок поставщика на надёжность, компетентность и т. д., необходимо опираться на 2 дополнительных принципа:

3. Принцип партнёрства. В случае построения взаимоотношений с поставщиками типа 2 важно выстроить партнёрские отношения, взаимное понимание и доверие, которое позволит минимизировать риски при возникновении непредвиденных проблем.

Прочность и эффективность отношений, как между организациями, так и между людьми, определяется степенью совпадения их ожиданий относительно друг друга. У каждого руководителя, как на стороне потребителя, так и на стороне поставщика ИТ-продуктов и/или ИТ-услуг, могут быть по-своему сформулированные ожидания и своё понимание ролей, которые могут играть поставщики и потребители на рынке информационных технологий. Несовпадение взглядов на логику взаимоотношений и ожиданий является одной из основных проблем выстраивания партнёрских отношений с поставщиками. CIO должен заранее объяснять, что поставщик может получить, решив ту или иную задачу, обрисовать перспективы партнёрства и т. д. Только взаимное понимание и терпеливое отношение друг к другу может привести к прочным взаимоотношениям. Для построения долгосрочных отношений в них надо бережно и терпеливо инвестировать. Только тот, кто готов терпеливо находить взаимный баланс интересов, сможет стать наилучшим партнёром для CIO.

4. Принцип соответствия размеров. Простые закупки (ПК или нижних серверных линеек) можно осуществлять и у поставщиков типа 1. Однако, при работе над сложными проектами с поставщиками типа 2 важное значение имеет соотношение размеров заказчика и ИТ-поставщика. Не стоит выбирать самого большого поставщика, равно как не стоит работать и с маленьким. Опыт показывает, что нужно стремиться к тому, чтобы в обороте поставщика пакет заказов от вас составлял от 3-5%. Говоря о соответствии размеров поставщика и заказчика при поставке сложных продуктов или услуг, можно дать несколько советов:

Три основных принципа работы с поставщиками услуг: последовательность (каждый интегратор должен знать, что его «поляна» защищена, пока он работает эффективно и добросовестно), твёрдость, поскольку нужно ограничивать «полет фантазии» интеграторов в части ценообразования, и наконец, прозрачность всех взаимоотношений в рамках сотрудничества.

Дмитрий Назипов

- заказчик должен быть более крупной организацией, чем поставщик, с поставщиком не следует разговаривать «снизу-вверх»;
- необходимо адекватно оценивать ресурсы поставщика, его способность вести проекты необходимого для заказчика масштаба (именно оценивать, а не полагаться на его заявления);
- необходимо адекватно оценивать финансовую надёжность поставщика.

Следует особо отметить риски привлечения малых компаний и фрилансеров. С одной стороны, эти поставщики быстро и оперативно решают текущие и частные задачи и дают достаточно явный результат для отдельных

задач бизнеса. С другой, если фрилансер или несколько ключевых сотрудников небольшого ИТ-поставщика прекращают работу в организации, это грозит полной потерей экспертизы в той или иной ИТ-области, так как небольшой поставщик не сможет адекватно заместить вышедшие ресурсы.

Наконец, нужно помнить, что ИТ-подразде-

ление традиционно остаётся внутренним поставщиком ИТ-услуг для своей организации, и выстраивая процедуры управления и цепочки взаимодействия с ИТ поставщиками и вендорами, СІО стоит подумать о том, на каких принципах он хотел бы выстраивать отношения между ними и своим собственным заказчиком — бизнесом.

Процесс управления поставщиками

Построение системы управления поставщиками СІО должно начинаться с анализа требований законодательства и корпоративных стандартов.

Необходимая для этого информация может содержаться в бизнес-стратегии и ИТ-стратегии организации, в общекорпоративных политиках и регламентах управления закупками (Положение о Закупках), шаблонах заключения договоров (у юристов могут иметься стандартные договоры для различных направлений деятельности).

На мой взгляд, не стоит выбирать максимально большого поставщика, равно как не стоит работать и с маленьким. Лучше стремиться к тому, чтобы в обороте поставщика пакет заказов Вашей компании составлял бы не менее 3-5%. Тогда Ваша компания будет интересна для поставщика, и Вы сможете рассчитывать на серьёзное отношение к Вашим задачам и на возможность мобилизации дополнительных ресурсов при необходимости

Сергей Кирюшин

Требования, предъявляемые к организации закупочных процедур в соответствии с упомянутыми 44-ФЗ и 223-ФЗ будут полезны не только для организаций, деятельность которых они регулируют, но и ко всем остальным организациям. Также, в целях подтверждения обоснованного отказа в выборе и мониторинга поставщиков не стоит пренебрегать

знакомством с положениями Федерального закона 115-ФЗ «О противодействии легализации (отмыванию) доходов, полученных преступным путём, и финансированию терроризма». Немало полезных идей, не только в части взаимодействия с ИТ-поставщиками, но и для бизнеса в целом может найти в Федеральном законе «О государственном оборонном заказе» от 29.12.2012 N 275-ФЗ.

Хотя действующее законодательство не содержит исчерпывающего перечня торговых и неторговых способов закупок, однако положениями ФЗ №223 (ч.3. ст. 3) предусмотрена возможность для организаций заказчиков указать в своём Положении о закупках иные способы закупок, помимо конкурса и аукциона (они в соответствии с Гражданским кодексом РФ являются торгами и должны быть обязательно отражены в Положении о закупках, независимо от того, будет ли использоваться эти способы или нет).

Способы закупок:

1. Конкурентные:

- путём проведения торгов (конкурс, аукцион)
- без проведения торгов (например, запрос предложений, запрос котировок, конкурентные переговоры)

2. Неконкурентные (закупка у единственного поставщика/источника).

Единственный поставщик — способ закупки при котором контракт заключается с конкретным поставщиком без применения кон-

курентных способов осуществления закупки, т.е. организация-заказчик предлагает заключить контракт исполнителю (физическому или юридическому лицу) принимая на себя большую часть рисков некачественного или несвоевременного исполнения условий. Необходимо отметить, что законодательство накладывает существенные ограничения для данного способа закупки.

Кроме того, СІО важно понимать, как в организации обеспечен процесс бюджетирования (см. подробнее в главе 3.2 «Управление финансами») и учёт закупаемого ИТ-оборудования, расходных материалов, нематериальных активов и на какие статьи плана счетов относятся расходы по проектам построения инфраструктуры и внедрения информационных систем.

Планирование и организация процесса

Процесс управления поставщиками может рассматриваться в тесной связи со сквозным процессом «от закупки до оплаты» и предусматривает прохождение следующих этапов:

1. Определение потребностей.

2. Разработка требований к поставщику и формирование перечня аккредитованных поставщиков.

3. Оценка и выбор поставщика-победителя.

4. Определение условий и механизмов мониторинга и контроля исполнения договорных обязательств и заключение договора.

5. Мониторинг и контроль исполнения договорных обязательств и изменений состояния поставщика, и управление изменениями (новыми требованиями).

6. Приёмка и закрытие работ по договору.

7. Обработка платежей.

Далее мы последовательно опишем все эти этапы.

1. Определение потребностей

Процесс управления поставщиками начинается со сбора и формализации требований. Требования могут поступать либо от бизнеса (прикладные запросы на автоматизацию и модернизацию бизнес-процессов), либо от сотрудников ИТ-департамента и проектных рабочих групп (инфраструктурные запросы на обновление и развитие ИТ-инфраструктуры). Следует отметить, что это два существенно разных вида требова-

Очень важно соотношение размеров заказчика и поставщика. Когда мы стартовали внедрение биллинговой системы, Билайн был небольшой компанией, всего около 1 млн. абонентов и вендор был втрое больше нас. Нам тяжело было с ними работать — за ними стояла большая машина и приходилось подстраиваться. Теперь, когда мы во много раз больше, чем вендор, отношения, конечно же, другие, иногда даже приходится действовать с позиции силы.

Важно реально оценивать возможности поставщиков, чтобы понимать, смогут ли они обеспечить твои потребности. Если ты понимаешь, что компания может, то есть шанс, что ты это получишь. Иначе ты можешь бесполезно ломиться в закрытую дверь.

Например, вендоры с доходом от 10 до 100 млн. долл. — это небольшие производители, которые случайно попали на рынок, выехав на одном крупном заказчике. И с ними очень тяжело работать: когда ты начинаешь, то видишь хороших продавцов, когда продолжаешь подготовку к проекту — тебе выставляют лучших аналитиков и архитекторов, а потом на проект приходит непонятно кто, выясняется, что серьёзных ресурсов у компании нет, терпеть она не умеет, а наверху у неё стоит «безумный» финансист, который требует оплаты. А если они выиграют ещё один тендер, то разговаривать с ними, как показывает мой опыт, становится совершенно бесполезно.

Владимир Филиппов

ний и должны быть утверждены конкретными заказчиками:

- требования бизнеса должны быть подтверждены Заказчиком, желательно в ранге руководителя бизнес-направления.
- требования, направленные на развитие инфраструктуры утверждаются либо самим СIO, либо вице-президентом, курирующим ИТ.
- для прикладных задач, которые не содержат единого заказчика, требования должны быть утверждены либо руководством, либо комитетом.

На основании этих первичных требований, СIO и группа ИТ-планирования (если такая существует) делают первичную оценку потребностей по объёмам и денежному эквиваленту, привязывая их к статьям бюджета.

2. Разработка требований и формирование перечня аккредитованных поставщиков

В последние годы широкое распространение получила процедура аккредитации, которая проводится в целях защиты от недобросовестных поставщиков и оптимизации и повышения эффективности механизмов управления закупочной деятельностью. Общий перечень поставщиков по каждой группе статей бюджета, или перечень, подготовленный специально для проведения конкурса, называют «длинным списком», из которого будет впоследствии формироваться перечень потенциальных поставщиков, так называемый «короткий список». Для такого отбора подготавливается предквалификационная документация, в ней устанавливаются необходимые требования к участникам, а также критерии отбора, а к участию в таком отборе приглашаются любые заинтересованные лица (процедура открытая). Количество отбираемых лучших потенциальных поставщиков и сроки аккредитации должны быть указаны в предквалификационной документации.

По итогам такого предварительного квалификационного отбора формируется Перечень

квалифицированных (аккредитованных) поставщиков (подрядчиков, исполнителей), которые будут приглашаться на конкурс на выполнение работ/поставку товаров.

Как мы указывали выше, всех ИТ-поставщиков и закупки в ИТ удобно разделить на два вида:

- закупки простых продуктов (типовое оборудование, простое ПО) и некоторых видов простых услуг (поставщики типа 1);
- приобретение сложных продуктов или средних по сложности и комплексных услуг (поставщики типа 2).

Этим двум типам продуктов и услуг соответствуют два подхода к выбору:

- выбор по критерию минимальной цены из «короткого списка»;
- выбор через сложный и многокритериальный анализ различных характеристик продукта и услуги.

Оценка поставщика типа 1. В случае закупки простых продуктов у поставщиков типа 1 формирование «короткого списка» и выбор поставщика проходит по укороченному перечню параметров/требований:

- репутация поставщика и наличие соответствующих прав, сертификатов (от вендоров) и, в ряде случаев, проведение экспертизы;
- финансовое состояние, опыт поставок и репутация.

В короткий список попадают поставщики, которые отвечают вышеприведённым требованиям. Окончательный выбор поставщика далее идёт только по цене, и главная задача при этом — оптимизация расходов на ИТ. Причём на поставщиков можно оказывать давление в отношении цены, так как рынок таких услуг велик, и заменить одного поставщика на другого достаточно просто.

Оценка поставщика типа 2. При работе над сложными решениями, критичными для бизнеса, или касающимися прикладных задач,

отношения с поставщиком долгие, и важно их правильно выстроить. В этом случае, помимо вышеприведённых требований к ИТ-поставщику, должны быть чётко сформированы расширенные требования, которые включают гораздо более подробные параметры/требования:

- повышенные требования к финансовой устойчивости;
- экспертиза поставщика и квалификация сотрудников поставщика (в том числе и в дополнительных услугах);
- реальный и доказанный опыт работы поставщика или требования к организации референс-визитов и предоставления информации;
- знание и понимание организации-заказчика и рынка, на котором он работает;
- срок работы на рынке и в отрасли, репутации, партнёрские статусы и т.д.;
- география точек присутствия;
- требования к проведению обучения (при необходимости) и другим дополнительным услугам и пр.

Наличие у интегратора хорошей экспертизы важно, но сегодня экспертиза зачастую быстро покупается и мигрирует из компании в компанию. Сегодня в одной компании, через полгода — в другой. Та экспертиза, в которую «вложился» и создал интегратор, может очень быстро перейти к его конкурентам. Поэтому сегодня критерий наличия экспертизы может оказаться вторичен по отношению к размеру компании, её устойчивости и умению реализовывать проекты нужного масштаба.

Сергей Кирюшин

В случае выбора сложного продукта или услуги понять, кто является оптимальным поставщиком, намного тяжелее. В этом случае необходимо применение таких инструментов, как

Не стоит выбирать поставщиков только по принципу максимальной экономической выгоды на текущий момент времени. Выбор партнёров, с помощью которых будет создаваться информационная система предприятия, имеет долгосрочные последствия.

Борис Славин

«длинный» и «короткий список» и «матричной» модели оценки поставщиков, с присвоением весов каждому из заданных параметров (образец матрицы оценки поставщиков смотрите в Табл. 2.5.1). В этом случае на Конкурсной комиссии сначала утверждаются веса параметров, а уже затем проводится сама оценка. И цена, которая превалирует при выборе простых продуктов, может не только оказаться не на первом месте, но даже опуститься на последнее.

Можно говорить о следующей общей тенденции — чем сложнее продукт и критичнее проект для бизнеса, тем меньший вес будет иметь критерий цены.

3. Оценка и выбор поставщика-победителя.

В управлении поставщиками одной из сложных и важных задач является выбор поставщиков. При этом есть два различных способа, о которых мы упоминали выше: проведение конкурса и выбор поставщика не на конкурсной основе. В случае организации конкурса организация-заказчик (организатор конкурса) назначает Рабочую группу (куда ИТ-специалисты входят наравне с другими представителями бизнес-подразделений) и Конкурсную комиссию (куда входит CIO), которая и несёт ответственность за результаты конкурса.

Рабочая группа в ходе подготовки конкурса знакомится с подготовленным ИТ-подразделением «длинным списком», результатами предквалификационного отбора, проводит сопоставление каждого аккредитованного поставщика с исходными требованиями к поставщику и формирует «короткий список» потенциальных поставщиков для приглашения

к участию в конкурсе. При этом сложность и продолжительность процедур отбора в разных организациях могут варьироваться в зависимости от требований для организаций различных типов и форм собственности в соответствии с действующим законодательством.

При формировании короткого списка следует также руководствоваться простым правилом «доверяй, но проверяй» и помимо перечня возможностей, анонсируемых потенциальным поставщиком, оценить его юридический статус, деловую репутацию и состояние финансово-хозяйственной деятельности используя внешние источники информации об организациях, в первую очередь веб-сервисы для оперативной проверки контрагентов «Спарк-Интерфакс», Право.ру (данные картотеки арбитражных дел Высшего Арбитражного Суда РФ), Фокус-контур, а также данные по Единому федеральному реестру сведений о банкротстве, данные с сайтов ФНС, ФССП, ФССФМ, Росстата, Сообщениям о предстоящем изменении регистрационных данных ИП в ЕГРИП и другие.

На что следует обращать внимание при выборе короткого списка помимо перечня услуг/работ/продуктов, предлагаемых потенциальным поставщиком:

- Статус юридического лица — действующая или на одной из стадий ликвидации или реорганизации.
- Возраст организации — чем выше возраст организации, тем ниже вероятность того, что поставщик окажется неблагонадежным.
- Размер уставного капитала — чем выше уставной капитал, тем ниже вероятность того, что поставщик является неблагонадежным. С учётом имеющихся данных из открытых источников считается, что среднестатистические мошенники имеют возможность легко пожертвовать суммой до 100 000 рублей.

- Соотношение между численностью работников и выручкой (очень малая численность в сочетании с большой выручкой за последний год не всегда свидетельствует об высоком профессионализме).
- Количество организаций, где первый руководитель потенциального поставщика числится в качестве первого лица (больше трёх — уже фактор риска).
- Число организаций, зарегистрированных по тому же адресу.
- Наличие информации о нахождении поставщика на одной из стадий банкротства. Это весьма существенный фактор, поскольку средства, перечисленные поставщику-банкроту, могут пойти на погашение исков кредиторов, а не на закупку необходимого заказчику оборудования.
- Наличие арбитражных дел и исполнительных производств в отношении организации поставщика.
- Наличие в реестре недобросовестных поставщиков
- Наличие задолженности по налогам
- Наличие отрицательных чистых активов, низкие показатели финансово-хозяйственной деятельности или отсутствие данных, предоставляемых в Росстат за последние 2 года

Эти же показатели могут использоваться для как для ранжирования поставщиков — контрагентов на этапе аккредитации и отбора, так и для регулярного мониторинга изменения их состояния в ходе исполнения договорных обязательств.

По решению организатора конкурса для потенциальных поставщиков, попавших в короткий список, могут проводиться открытые и закрытые конкурсы. Для выбора поставщика программных продуктов в области бизнес-приложений и поставщика услуг по его внедрению, а также поставщиков типа 2 предпочтительным является проведение закрытого конкурса.

При выборе поставщика услуг важно обращать внимание на логику развития интегратора на рынке. Кто-то растёт, садясь на одного большого заказчика и благополучно осваивая его бюджет, кто-то пользуется административными ресурсами, а кто-то настойчиво вырабатывает свою экспертизу. И внешне у компаний результат один и тот же — преуспевающий интегратор. Однако в первом и втором случаях, то, как интегратор будет чувствовать себя в следующем году, зависит от немногочисленных и достаточно случайных факторов. Устойчивость такого интегратора может оказаться ниже, чем среднерыночная.

Дмитрий Назипов

В зависимости от принятого в организации Положения о закупках торги могут быть одноэтапными и двухэтапными. Для поставщиков ИТ-оборудования, расходных материалов, ПО общего назначения, других простых работ и услуг, не предусматривающих взаимосвязано закупку ПО и услуг по их внедрению, может проводиться одноэтапный конкурс.

Если говорить о внедрении сложных бизнес-приложений, конкурс, как правило, является двухэтапным, с предварительным квалификационным отбором.

1. Первый этап — выбор поставщиков программного обеспечения и выбор ПО по конкретным функциональным и архитектурным требованиям и общим и техническим требованиям.

2. Второй этап — выбор поставщика услуг по внедрению.

Для большинства случаев выбора поставщика типа 2 рекомендуются двухэтапные конкурсы. На основании результатов анализа полученной от поставщиков информации о соответствии функциональным и техническим и общим требованиям, а также по результатам референс-визитов Рабочая группа готовит рекомендации для Конкурсной комиссии, ко-

торая принимает окончательное решение о выборе победителя.

После принятия решения о выборе победителя может быть подписан протокол о намерениях (хотя это не обязательно и практикуется не всегда). Протокол о намерениях необходим для фиксации того факта, что стороны достигли определённых договорённостей и начинают двигаться в сторону заключения контракта. Это предконтрактная форма формализации партнёрства, дающая сторонам право далее тратить ресурсы на работу с тем партнёром, которого они выбрали, для детализации всех контрактных условий.

Подписав протокол о намерениях, стороны могут планировать свою производственную деятельность, формировать программу работ, заказывать и получать проектно-сметную документацию и т. д.

Необходимо отметить, что конкурсные процедуры сами по себе не являются панацеей: «практика тендеров даёт частичное решение проблемы — уровень поставщиков и связанные с этим риски часто не учитываются или учитываются не полностью. Без соответствующих мировой практике процедур оценки и отбора поставщиков мы обречены на нерациональные затраты при выполнении контрактов — и чем крупнее контракт, тем выше эти затраты».

Выбор поставщика не на конкурсной основе

Конкурсы — это важный элемент процесса выбора, однако опыт российских ИТ-директоров свидетельствует, что в рамках конкурсных процедур невозможно провести детальную оценку поставщиков. В результате многие ИТ-директора применяют конкурс только как техническую процедуру, а весь выбор поставщика ведут заранее.

Более того, именно в предварительной работе в потенциальным победителем тендера они видят залог будущего эффективного партнёрства. То есть, по сути, при таком подходе

Табл. 2.5.2. Построение взаимоотношений с поставщиками различных типов.

Характеристики	Взаимоотношения с поставщиком типа 1	Взаимоотношения с поставщиком типа 2
Риски	Небольшие	Высокие
Выстраивание партнёрских отношений	Не нужно	Необходимо
Конкурс	Открытый	Закрытый
Этапы конкурса	Одноэтапный	Двухэтапный
Подходы к выбору	Выбор из короткого списка по цене	Использование матричной системы оценки по большому количеству критериев

победитель тендера должен быть понятен СЮ ещё до его начала, а сама процедура тендера может помочь детально согласовать ожидания обоих партнёров и повлиять лишь только на финансовую сторону взаимоотношений. Как мы уже писали выше, успешное партнёрство строится на глубинном понимании обоими сторонами приоритетов, правил и принципов работы друг друга. Добиться выгодных условий от поставщика можно только в том случае, если СЮ хорошо понимает его бизнес. Аналогично и от поставщика требуется хорошо понимать задачи, стоящие перед заказчиком, тенденции в его отрасли и т. д. Для СЮ проблема понимания охватывает целый спектр вопросов:

- Какой реальный интерес у поставщика к работе с организацией? Начиная с какой суммы контракта, работа с вами, как с потенциальным партнёром, становится интересной для данного ИТ-поставщика?
- Кто и за что отвечает внутри организации-поставщика? Кто у поставщика принимает ключевые решения, и кто из этих людей участвует в переговорах?

- Каковы ожидания поставщика от работы с нами? Насколько точно мы донесли поставщику наши требования, и как он нас понял?
- Каково в общих чертах ценообразование у поставщика и каков уровень цен на рынке?
- Готов ли ИТ-поставщик к регулярному пересмотру требований и условий, связанных с изменяющимися рыночными условиями, нормативно-правовыми актами и/или бизнеса заказчика?

Проблеме понимания всегда сопутствуют проблема доверия, которую можно резюмировать такой фразой: «никогда не доверяй»

Выбирая поставщика, я прежде всего выбираю партнёра. Я ни разу не начинал тендер, не понимая, кто его выиграет. Безусловно, тендер нужен, чтобы довести до правильного уровня ожидания партнёра, прежде всего по деньгам. Но партнёра я всегда выбираю заранее. Выбор правильного партнёра - это важнейшее умение СЮ, и его нельзя доверять случаю.

Павел Пестряков

те тем, кого вы не понимаете». Здесь вопросы другие:

- Что реально сделала команда ИТ-поставщика по сравнению с декларируемыми результатами? Какими реально знаниями и экспертизой обладают сотрудники заявленной команды? Кто из сотрудников принимал участие в проектах, которые декларируются как референтные?
- Насколько можно полагаться на поставщика в случае возникновения непредвиденных проблем?

Последнее — очень важный аспект при построении взаимоотношений, которому надо уделить особое внимание.

Проблемы в ходе выполнения ИТ-проектов возникают регулярно и для их преодоления необходимы дополнительные ресурсы, как людские, так и временные. И вопрос «насколько поставщик сможет и будет готов предоставить необходимые ресурсы» является ключевым. Острота проблемы ресурсов может зависеть от многих факторов, на которые СТО повлиять не сможет, например:

- наличия большого количества других проектов у потенциальных поставщиков, в то время как вы пытаетесь привлечь их для решения ваших задач;
- недоступности высококвалифицированных аналитиков и архитекторов, в то время как простаивают менее квалифицированные специалисты;
- наличия поставщиков/специалистов нужной квалификации в регионах, особенно если необходимо обеспечить единый стандарт услуг разных городах страны.

Для максимальной нейтрализации приведённых выше рисков можно дать такой совет — взаимодействовать лучше с теми организациями, с руководителями, с которыми у СТО наилучшие отношения, где больше всего ваше влияние на принятие решения внутри поставщика. Ещё лучше, если это взаимодействие строится на уровне акционера или собственника организации. В этом случае вы имеете реальные шансы быстро закрыть появившуюся проблему дополнительными ресурсами.

Градации поставщиков по уровню рисков

При проведении оценки поставщиков можно использовать механизм градации поставщиков по уровню рисков. Для этого вводится чёткая градация поставщиков — уровня А (или «золотые» — надёжные, безупречно отрабатывающие контрактные условия), уровня В (или «серебряные» — относительно надёжные), уровня С (или «бронзовые» — иногда допускающие сбои). Закупочные цены для поставщиков каждого уровня — разные, цены для «С» и «А» могут различаться на 20% и более, но, покупая дороже, организации на самом деле экономят на рисках. Понятно, что работать с поставщиками, которые определены как ненадёжные (уровень ниже «С»), организации соглашаются только тогда, когда они сами не заботятся о качестве получаемых услуг и производимой продукции.

Градации поставщиков по уровню рейтинга

При проведении оценки и последующего автоматизированного мониторинга изменения состояния поставщиков в крупных организациях достаточно удобно пользоваться следующей шкалой ранжирования, в которой 100 баллов присваивается «идеальной компании» и все поставщики делятся на 6 групп:

А — поставщики, по которым негативные факторы практически отсутствуют (их примерно 10 % от общего числа)

В – поставщики, по которым выявляются отдельные негативные факторы имеющие достаточно высокий рейтинг (согласно статистике, их примерно 40 %)

С – Поставщики с рейтингом ниже среднего и более существенным набором негативных факторов, чем в группе «В» (от 35 до 40%)

Д – поставщики с низким рейтингом, по которым выявлен значительный объем негативных факторов, касающихся деловой репутации, состояния ФХДЮ исполнения договорных обязательств (их примерно 10-15 %)

Е – организации в процессе ликвидации или реорганизации, по которым необходим постоянный мониторинг (примерно 0.2%)

Ф – организации, ликвидированные, в соответствии с законодательством РФ (таких может быть до 0.2% от общего числа)

Ещё один важный совет — тестирование поставщиков услуг на небольших задачах. Ведь не попробовав работать с поставщиком, невозможно глубоко понять его работу, систему управления, возможные ресурсы, его готовность терпеть и инвестировать в партнёрские отношения, а значит, практически невозможно правильно выбрать партнёра.

Специфика российских интеграторов — ими, как правило, до сих пор управляют владельцы. А значит, лучше (если есть возможность) выходить на владельцев компаний и договариваться именно с ними, а не с менеджерами, которые могут поменяться. Серьёзные решения, например, быстро увеличить ресурсы на проекте, зачастую могут принимать только владельцы компаний-интеграторов.

Сергей Кирюшин

4. Определение условий и заключение договоров

Прежде всего, СІО необходимо чётко оговорить с юристами организации и финансовыми службами зоны ответственности при подготовке договоров. Все договора, которые относятся к деятельности ИТ (на любые виды ИТ-услуг и поставку ИТ-оборудования и материалов), должны готовиться в рамках ИТ-подразделения.

Если говорить о договорах с поставщиками типа 2, то практика, когда договора, относящиеся в ИТ готовятся бизнес-подразделениями не может быть признана успешной. Специалисты бизнес-подразделений не могут учесть все нюансы и тонкости хода проекта и взаимодействия с подрядчиком, поскольку не являются в этом профессионалами. Таким образом, именно СІО должен отвечать за подготовку договоров, касающихся ИТ. И СІО должен либо подписывать договора (в зоне своей ответственности), либо, если договор подписывается его руководителем, СІО должен согласовать договор. При этом надо чётко понимать, что подписание договора, равно как и других документов по нему, влечёт за собой финансовые обязательства.

Переговорный процесс

Заключению договора предшествует переговорный процесс.

В случае договора с поставщиком типа 1 он, фактически сводится к минимуму. Но в случае построения взаимоотношений с поставщиком типа 2 переговорный процесс становится важнейшим элементом построения долгосрочных взаимоотношений. В этом случае качество договорного процесса — очень важный момент. В современном мире времени постоянно не хватает, надо двигать-

К сожалению, без ИТ-поставщиков обойтись нельзя. И в очередной раз, начиная проект, ловишь себя на мысли: «ну вот, снова все сначала...». Идёт новая волна проектов, приходится работать с поставщиками и снова убеждаешься, что они ничего кроме как двигать коробки не умеют... Но деваться некуда, а значит:

- надо выращивать у себя экспертизу по их продуктам, надо понимать, куда и как они развиваются, есть примеры, когда очень удачные продукты не внедряются только потому, что российский рынок неинтересен для вендора;
- их надо терпеливо выращивать, надо воспитывать их команду, надо договариваться, например, если у поставщика появляется внятный эккаунт-менеджер, его ни в коем случае не надо менять;
- надо понимать, в каком состоянии они находятся и каковы их возможности, как строится работа внутри их организации, как работает их система и т.д.;
- их надо знать в лицо, не только лично, но и на деловом уровне — кто за что отвечает, понимать изменения в их структуре;
- надо знать менеджеров поставщика вплоть до верхнего руководства, включая штаб-квартиры международных вендоров, потому что если у тебя нет контакта с вице-президентом международного вендора, скажем, по региону EMEA, то многие вопросы просто не решаются;
- их надо вовремя заменять, потому что со временем поставщики, как правило, портятся, причём иногда бывают скоропортящиеся, а иногда и более долгоживущие.

Владимир Филиппов

ся все быстрее, в результате из поля зрения выпадают важные вопросы формализации достигнутых договорённостей. Но чем качественней будет построен договорной процесс, тем легче будет жить в дальнейшем.

Для того, чтобы обеспечить получение ожидаемых параметров договора, необходимо иметь чёткий план ведения переговоров, а также основанный на цифрах перечень аргументов по пунктам договора и приложений к нему. До проведения переговоров следует определиться с желаемыми условиями, на достижение которых направлены переговоры, рамками стоимости договора и критичных требований к приобретаемым услугам и продуктам. Тщательная подготовка к переговорам с поставщиками и грамотное составление договоров позволяют реализовать проекты и организовывать поставки с ми-

нимальными для организации рисками, избегать неприятных ситуаций и результативно отстаивать свои права в суде. В частности, для успешного ведения переговоров по контракту нужно заранее определиться со следующими вопросами:

- Каковы возможные цели поставщика в переговорном процессе?
- Какие пункты проекта договора вызвали дополнительные вопросы со стороны юридических и финансовых служб (в частности, порядок формирования стоимости закупок, размеры скидок, возможные штрафные санкции, порядок оплаты, другие возможные условия)?
- Насколько полно прописаны требования в техническом задании, как приложении к договору? Что конкретно предполагается сделать? Каковы риски вследствие непол-

ноты и недостаточной конкретности технического задания?

- Что может помешать выполнению договора? Какие необозначенные ранее риски несёт организация по данному договору?
- Кто и как будет контролировать объем и сроки выполнения работ?
- Как будет оцениваться качество выполненных работ и представляемых результатов?
- Если контракт не предусматривает фиксированной стоимости, как будут контролироваться и подтверждаться трудозатраты?
- Как и кем будут согласовываться возможные изменения и дополнения к договору?
- Как будут приниматься результаты работ?
- Какие другие ограничения и условия должны быть внесены в договор и приложения к нему?

Данный перечень можно взять за основу для подготовки перечня вопросов для подготовки и проведения переговоров с поставщиком, добавив к нему свои специфические вопросы.

Структура переговоров должна быть понятна обеим сторонам — в процессе переговоров должны укрепляться доверие и открытость. Если установлены доверительные отношения, добиться согласования юридических и стоимостных характеристик контракта гораздо проще. И наоборот — если на этапе согласования юридических и стоимостных характеристик контракта возникают споры, то стоит задуматься — а правильный ли поставщик выбран?

Заключение контракта. После проведения переговоров стороны приходят к заключению контракта. Независимо от уровня взаимоотношений с поставщиком и степени влияния личностных и субъективных факторов, контракт с ним должен быть максимально жёстким и в максимальной степени фиксировать все нюансы взаимоотношений. Какой бы

Работать с интеграторами проще, чем с вендорами, поскольку можно терпеливо строить взаимоотношения, строить их команду, глубоко заходить в их структуру и т.д. Я всегда подбираю тех, кто может сначала потерпеть. В классическом сценарии - приходит «коммивояжер» выкладывает кучу продуктов — я обычно действую так: давайте найдём небольшую понятную задачу, за которую не будет денег (собственно, чего ради?), вы её сделаете, пока потерпите, но покажете свою экспертизу и ресурсы, мы их увидим и тогда, возможно выберем вас для большого проекта. К сожалению, так практически ни с кем из интеграторов работать не получается, их принципы работы и структура не предусматривают это. Очень мало тех, кто готов вложиться и потерпеть, конечно, с риском потерять деньги, но зато, как показывает опыт, те, кто так терпеливо и с пониманием входил в нашу деятельность, оставались надолго.

Владимир Филиппов

уровень доверия и отношения с поставщиком ни сложился, в жизни все происходит по-разному, и не всегда все при дальнейшей работе складывается хорошо. Поэтому договора должны быть составлены максимально требовательно по отношению к поставщикам.

В случае возникновения спорных моментов при обсуждении договоров на стадии подписания рекомендуется подписывать протоколы разногласий, которые позволяют свести спорные условия договора к «общему знаменателю», устраивающему все стороны.

Важные моменты, на которые надо обратить внимание при составлении контракта:

- не абстрактное качество, а конкретные SLA или сформулированные иным образом параметры качества и предоставления продуктов и услуг;
- определение механизмов мониторинга и контроля исполнения договорных обяза-

Табл. 2.5.3. Пример матрицы оценки поставщиков

№	Критерий	Вес критерия/ Максимальный балл	Поставщик 1	Поставщик 2
1.	Инвестиции в технологии	3		
2.	Инвестиции в технологии	2		
3.	Опыт работы и внедрения проектов в России	1		
4.	...			
Характеристики продукта				
5.	Функциональные возможности системы	10		
6.	Технологические возможности	3		
7.	Возможности интеграции в среду компании	4		
8.	Потенциал развития	3		
9.	Информационная безопасность	2		
10.	Надёжность системы	2		
11.	...			
Экономические показатели				
12.	Общая стоимость решения	9		
13.	Стоимость миграции	3		
14.	Эффект от внедрения	20		
15.	Риски проекта	5		
16.	...			
Гарантии и ответственность				
17.	Гарантии поставщика	4		
18.	Ответственность поставщика	4		

19.	...			
15.	Риски проекта	5		
	ИТОГО:	100		

тельств (определение механизмов мониторинга и контроля исполнения договорных обязательств может рассматриваться как отдельный раздел в общем процессе управления поставщиками, однако, эти задачи являются неотъемлемой частью работ по подготовке договора с поставщиком и приложений к нему);

- штрафные санкции в случае нарушения сроков и качества предоставления продуктов и услуг (хотя взыскание с подрядчиков штрафов или пени за просрочку или некачественное выполнение работ в российской практике — явление весьма редкое, и кроме того, надо принять во внимание, что тут играет роль российское бухгалтерское законодательство — такой возврат средств очень неоднозначно воспринимается бухгалтерией);
- механизмы управления изменениями и новыми требованиями;
- действия при возникновении проблем с командой поставщика, что, к сожалению, случается часто;
- условия пролонгации и закрытия договора.

Для договоров, которые предполагают выполнение проектных работ, обязательным приложением должен быть детально проработанный Устав проекта. В ряде случаев поставщики ИТ-услуг предлагают включать его в состав оплачиваемого подготовительного этапа работ, однако лучше включить Устав в виде приложения к договору и прописать в нем механизмы и инструменты мониторинга и контроля исполнения договорных обязательств. Когда заканчивается срок действия договора с существующим поставщиком у СІО есть три

альтернативы: пролонгировать договор, пересмотреть условия существующего договора или выбрать нового поставщика.

Для пересмотра договора с ИТ-поставщиком нужно иметь достаточно вескую аргументацию в виде оформленного в письменном виде перечня вопросов по цене, качеству и срокам поставки ИТ оборудования, материалов или предоставления услуг. Прежде чем пересматривать договор, необходимо провести оценку рисков, определить ключевые изменения, которые СІО хочет увидеть в вашем новом договоре, и заручиться поддержкой юристов организации. Нужно иметь в виду, что в случае пересмотра договора у СІО меньше рычагов воздействия на поставщика, чем при проведении конкурса.

Пролонгация договора с ИТ-поставщиком услуг подразумевает автоматическое продление договора с минимальными изменениями

Есть ситуации, когда тендеры экономически оправданны и эффективны.

Это простые и типовые продукты и услуги. Я убеждён, что в сложных проектах реальная эффективность заложена в партнёрских отношениях. Именно это позволяет интегратору правильно развивать свои ресурсы, а СІО — максимально эффективно решать поставленные задачи. И необходимость по тем или иным причинам использовать тендерные процедуры сильно мешает партнёрству. Но, тем не менее, партнёрские отношения значительно важнее тендера и возможности с помощью этого механизма некоторого снижения цены.

Андрей Кельманзон

условий контракта. Однако, не стоит использовать пролонгацию в течение длительного периода — более двух-трёх лет. При более длительных сроках лучшей альтернативой становится пересмотр договора.

Если ни пролонгация, ни повторный пересмотр договора с конкретным ИТ- поставщиком не является приемлемым, или Вы хотите «вежливо отказаться от услуг поставщика» — наилучшей альтернативой может стать повторный конкурс.

5. Мониторинг и контроль исполнения договорных обязательств и управление изменениями

Собственно, порядок и инструменты мониторинга и контроля исполнения договорных обязательств, инструменты управления изменениями (новыми требованиями) имеют значение в основном для поставщиков типа 2. Порядок и инструменты мониторинга и контроля должны быть определены в приложениях к договору, а для проектов — в Уставе проекта. Нет необходимости создавать регламент мониторинга исполнения по каждому договору, достаточно прописать ограниченный пере-

чень регулярно проверяемых пунктов, общих для всех договоров. В ряде случаев также рекомендуется иметь реестры открытых вопросов, куда заносятся все проблемные вопросы по работе поставщиков, а также реестры рисков и проводить анализ актуальности рисков не реже одного раза в месяц.

Поскольку мало какой проект или выполнение работ по договорам долгосрочных поставок обходятся без изменений, лучше иметь прописанный регламент внесения изменений и простой реестр запросов на изменения. До определённого уровня можно, конечно, держать все открытые вопросы и запросы на изменения в памяти, но лучше не рисковать, а внести данные в реестр и проверять статус исполнения в соответствии с простыми регламентами, пусть даже в виде листа напоминаний о регулярных действиях по мониторингу.

Практика выполнения долгосрочных договоров напрямую указывает на необходимость мониторинга не только исполнения самого договора, но и на мониторинг изменения со-

Вопрос, который задают себе все СЮ: «Как найти баланс, сколько заплатить денег поставщику, чтобы, с одной стороны, не переплачивать, а с другой — чтобы он не потерял интерес к твоему проекту, потому что денег в нем мало?» Цена, которую подрядчик озвучивает заказчику, как правило, складывается из трёх составляющих: себестоимости работ, наценки поставщика и финансовой страховки рисков. Резерв снижения цены, очевидно, лежит прежде всего в снижении финансовой страховки рисков, которые в ряде случаев могут достигать 50% от стоимости контракта. В ходе переговоров с поставщиком СЮ может повлиять на эту сумму, показав, как он работает с рисками, или взяв на себя часть из них.

В случае, когда риски сыграли, основным приоритетом поставщика станет стремление минимизировать свои затраты. Это приведёт к попытке выполнить работы менее квалифицированными специалистами, упростить технологию решения задач и т.д. То есть сыграть за счёт качества. При этом важно понимать, что если СЮ «выжмет» поставщика «досуха», то возникновение любой мало-мальски сложной ситуации ведёт к большой проблеме. В этом случае у поставщика пропадают стимулы — нет ни желания работать с компанией, ни дополнительных ресурсов и т.д.

Поскольку сложные проекты, как правило, требуют дополнительных непредусмотренных ресурсов, в бюджете проекта необходимо оставлять некоторый резерв на случай непредвиденных ситуаций, изменения проекта (здесь средний показатель — около 5% от бюджета проекта, хотя, конечно, это сильно зависит от специфики проекта, его рискованности, инновационности и т.д.) или по-другому управлять этими рисками.

стояния самого поставщика по ряду ключевых показателей, которые рассматривались при по этапу «Оценка и выбор поставщика-победителя». Такой мониторинг поставщиков организация может проводить либо самостоятельно, регулярно собирая и анализируя данные из открытых источников в соответствии с собственными критериями оценки, либо воспользоваться сервисами специализированных агентств, предоставляющих такую информацию на коммерческой основе.

6. Приёмка и закрытие работ по договору

Формальными документами, обосновывающими завершение работ по договору в полном объёме, является акт сдачи приёмки работ, формат которого является обязательным приложением к договору, а также товарно-транспортная накладная, являющаяся обоснованием для передачи ПО/оборудования/ материалов.

Подписывая Акт приёмки сдачи работ, заказчик своей подписью подтверждает, что работы были выполнены в полном объёме и соответствуют техническому заданию. С момента подписания Акта приёма-передачи ИТ-директора все риски утраты или повреждения оборудования, некорректной работы настроенной информационной системы и т. д. переходят к заказчику. И, как правило, этап до подписания акта — это последняя возможность для ИТ-дирек-

тора решить большую часть открытых вопросов и воздействовать на поставщика для обеспечения необходимого качества работ по договору, либо поставляемого оборудования/ материалов (на основании статей 475 и 518 ГК РФ), решить разногласия в случае поставки некомплектного товара (на основании требований, предусмотренных статьёй 480 и 519 ГК РФ.)

Бывают ситуации, когда поставщик просит СТО подписать Акт приёмки-сдачи работ вместе с Гарантийным письмом поставщика о том, что последний довыполнит оставшиеся работы или допоставит оборудование. В случае, если СТО подписывает Акт, ответственность полностью ложится на него.

7. Обработка платежей

Обработка платежей, как правило, является последним этапом сквозной процедуры «от закупки до оплаты», однако зачастую она становится болезненным местом во взаимоотношениях с ИТ поставщиком. Связано это с тем, что практика российского бизнеса, допускающая частичную оплату за ещё невыполненные работы, вносит свои коррективы в стандарты, применяющиеся во многих развитых странах. Обилие схем промежуточных оплат также вносит свои коррективы в, казалось бы, простую процедуру. Когда перечень поставщиков и разнообразие договоров начинает исчисляться десятками, ИТ-директору не обойтись без реестра договоров, счетов и платежей, а также без регламента по работе с договорами и счетами для сотрудника, отвечающего за эту работу.

Связь процесса управления поставщиками с другими процессами ITIL

В соответствии с положениями ITIL рекомендуется рассматривать процесс управления поставщиками во взаимодействии с другими ITIL процессами, которые в той или иной мере уже затрагивались в предыдущих разделах. В частности:

При работе с поставщиками мы постоянно сталкиваемся с одной и той же проблемой — перетеканием кадров между поставщиками. К сожалению, по российскому законодательству очень сложно закреплять кадры на каком-то одном месте. И прописывание основной команды в уставе проекта здесь мало помогает. Для демпфирования рисков ухода команды поставщика из проекта, мы закладываем в бюджет проекта средства на самостоятельную покупку команды.

Павел Пестряко

- Управление качеством (Quality management)
- Управление уровнем услуг (Service Level Management, SLM)
- Управление финансами (Financial Management for IT Services, FIN),
- Управление проектами (Project Management)
- Управление ИТ-Активами (IT Assets Management)

Контроль состояния работы с поставщиками

Ведение учёта с поставщиками и подрядчиками — это важнейший элемент системы управления взаимоотношениями с ними. Такой учёт может вестись различными способами. Наиболее предпочтительный вариант — ведение такого учёта в рамках общей системы бюджетирования расходов на ИТ. В случае отсутствия системы бюджетирования можно использовать простые таблицы:

- реестр поставщиков и договоров;
- реестр счетов от поставщиков и оплаты.

Но в любом случае, учёт должен фиксировать:

- С какими поставщиками и вендорами заключены договора?
- Когда и на какой срок?
- Какова задача, которую решает поставщик и вендор?
- Какой объем оплаты соответствует заключённым договорам и ИТ бюджету, в какие

У меня как минимум раз в квартал возникают проблемы, связанные с уходом специалистов поставщика услуг. И каждый раз эти проблемы решаются по-разному. Я стараюсь переложить ответственность — и денежную, и организационную, и моральную — на партнёра, объяснить ему, что это не моя проблема.

Дмитрий Назипов

сроки, в какой мере эти договора отвечают корпоративным требованиям?

- Какова доля каждого поставщика в текущем объёме поставок ИТ оборудования, программного обеспечения материалов, работ и услуг?
- Какова история взаимоотношения с каждым их поставщиков (объем и качество выполненных ранее работ, поставленных материалов и оборудования, своевременность поставок, цены, задолженность поставщикам, правомерность авансов...), как они попали в портфель поставщиков ИТ?
- Какова задолженность организации перед поставщиками и вследствие чего она возникла?

Даже если общая политика закупок в организации не обязывает проводить конкурсы по выбору поставщиков, и, если в ИТ-подразделении уже есть привязка поставщиков к статьям бюджета, для дальнейшего планирования работы по процессу управления поставщиками, СІО необходимо регулярно проводить анализ состояния взаимоотношений с существующими поставщиками, проводить оценку их возможности выполнять контрактные обязательства.

Часть 2. Управление ИТ деятельностью

Глава 2.6

Управление ИТ-рисками



Павел
Кудрявцев

Определения

В России действует ГОСТ Р 51897-2011/Руководство ИСО 73:2009 «Менеджмент риска. Термины и определения», служащий терминологической основой для целой серии стандартов, посвящённых управлению рисками. В соответствии с ним:

Риск — следствие влияния неопределённости на достижение поставленных целей.

К этому определению необходимо сделать несколько комментариев:

- **неопределённость** — это состояние, полного или частичного отсутствия информации относительно понимания или знания события, его последствий или вероятности;
- **влияние неопределённости** — это отклонение от ожидаемого результата (целей) с позитивными или негативными последствиями.

Поэтому риск часто характеризуют путём описания возможного события и его последствий или их сочетания.

Событие — возникновение или изменение специфического набора условий.

Последствие — результат события. Событие может привести к ряду последствий.

ГОСТ Р 51897-2011

Суммируя вышеприведённые определения можно сказать, что риск — это следствие вероятности возникновения события, которое окажет влияние на достижение поставленных целей.

Здесь важно учесть три обстоятельства:

1. Имеется в виду будущее событие.
2. Событие может заключаться в том, что какое-то явление не имело места.
3. Событие может быть случайным, т. е. заранее неизвестно, произойдёт оно или нет, но тем не менее, есть основания полагать, что оно может произойти.

Кроме того, что результатом события может быть одно или более последствий, первоначальные последствия могут усиливаться за счёт эффекта домино.

Последствия могут быть ранжированы от позитивных до негативных. Однако применительно к аспектам безопасности всегда рассматриваются именно негативные последствия.

Существенно, что последствие может иметь положительные и отрицательные влияния на цели.

Таким образом, ИТ-риск — это следствия влияние неопределённости, связанной с применением информационных технологий, на достижение целей.

С учётом рассмотренных определений риски характеризуются двумя величинами:

- вероятность, которая характеризует наступление события (рискового события);
- степень влияния возможных последствий.

Источник риска (risk source) — объект или деятельность, которые самостоятельно или в комбинации с другими обладают возможностью вызывать повышение риска. Источник риска может быть материальным или нематериальным (ГОСТ Р 51897-2011). В области информационной безопасности часто употребляется термин «угроза», например, ГОСТ Р ИСО/МЭК 27000-2012 определяет угрозу (threat) как «возможная причина нежелательного инцидента, который может нанести ущерб системе или организации».

Уязвимость — внутренние свойства или слабые места объекта, вызывающие его чувствительность к источнику риска, что может привести к реализации события и его последствий (ГОСТ Р 51897-2011).

Взаимосвязь между основными терминами риск-менеджмента схематически представле-

на на Рис. 2.6.1.

Уязвимость объекта может позволить источнику риска спровоцировать некоторое событие, которое может иметь одно или несколько последствий. Событие может произойти или не произойти, и даже если оно произойдёт, последствия от данного события могут возникнуть или не возникнуть, следовательно появляется неопределённость, которая может повлиять на достижение поставленных целей, т.е. возникает риск.

Оценка рисков позволяет организации учитывать, в какой степени потенциальные события могут оказать влияние на достижение её целей. Иногда эти две величины объединяют в один показатель — значимость риска.

Совокупность факторов, по сопоставлению с которыми оценивают значимость риска, называют критерием риска (ГОСТ Р 51897-2011).

Введём ещё несколько важных определений:

Идентификация риска — процесс обнаружения, распознавания и описания рисков. Идентификация включает распознавание источников риска, событий, их причин и возможных последствий (ГОСТ Р 51897-2011).

Процесс менеджмента риска (управления риском) — взаимосвязанные действия по обмену информацией, консультациям, установлению целей, области применения, идентификации, исследованию, оценке, обработке, мониторингу и анализу риска, выполняемые в соответствии с политикой, процедурами и методами менеджмента организации.

