Предисловие

Описание

*Восьмое издание книги «Современный системный анализ и проектирование*» охватывает концепции, навыки, методы, методы, инструменты и перспективы, необходимые системным аналитикам для успешной разработки информационных систем. Основной целевой аудиторией являются студенты старших курсов, обучающиеся по учебным программам по информационным системам управления (MIS) или компьютерным информационным системам; вторичная целевая аудитория – специалисты MIS по программам MBA и MS. Несмотря на то, что эта книга не написана специально для младших колледжей и рынков профессионального развития, она также может быть использована этими программами.

У нас есть более чем 55-летний опыт преподавания системного анализа и проектирования, и мы использовали этот опыт для создания этого новейшего издания *«Современный системный анализ и дизайн*». Мы предоставляем четкое представление о концепциях, навыках и технологиях, которые необходимы студентам, чтобы стать эффективными системными аналитиками, которые работают с другими над созданием информационных систем для бизнеса. Мы используем модель жизненного цикла разработки систем (SDLC) в качестве организующего инструмента на протяжении всей книги, чтобы предоставить студентам сильную концептуальную и систематическую основу. SDLC в этом издании имеет пять фаз и круговую конструкцию.

С этим текстом мы предполагаем, что студенты прошли вводный курс по компьютерным системам и имеют опыт разработки программ по крайней мере на одном языке программирования. Мы рассматриваем основные системные принципы для тех студентов, которые не знакомы с материалом, на котором основаны методы разработки систем. Мы также предполагаем, что студенты имеют солидный опыт в области компьютерной грамотности и общее понимание основных элементов бизнеса, включая основные термины, связанные с функциями производства, маркетинга, финансов и бухгалтерского учета.

Новое в 8 издании

В восьмом издании появились следующие функции:

* *Новый материал.* Чтобы идти в ногу с изменяющейся средой разработки систем, глава 12 подверглась полному и тщательному пересмотру. В то время как облачные вычисления представлены в главе 2, они широко освещаются в пересмотренной главе 12. Сервис-ориентированная архитектура была вновь введена в книгу в версии главы 12. К числу других новых материалов относятся дополнения к двум приложениям к главе 7. Добавления, посвященные диаграммам деятельности и нотации управления бизнес-процессами, теперь содержат дополнительный текст и рисунки. На протяжении всей книги рисунки, таблицы и связанный с ними контент были обновлены и обновлены.
* *Обновленный контент.* На протяжении всей книги содержание каждой главы обновлялось там, где это уместно. В главе 2 мы расширили охват нескольких разделов. Примеры обновлений в других главах включают пересмотр информации о рынке труда в области информационных услуг (ИС)/информационных технологий в главе 1. Другой пример – глава 13, где мы обновили и расширили раздел о безопасности информационных систем. Все скриншоты взяты из актуальных версий ведущих программных продуктов. Мы также приложили особые усилия для обновления наших списков литературы, очистив устаревшие материалы и включив в них текущие ссылки.
* *Удаленный материал.*  В наших усилиях по поддержанию актуальности книги и ее рационализации освещение некоторых вещей было исключено из этого издания. Глава 1 больше не включает в себя быструю разработку приложений. Глава 12 больше не охватывает хранилища данных или витрины данных. В главу 13 больше не включен раздел, посвященный электронным системам поддержки служебной деятельности.
* *Организация.*  Мы сохранили организацию книги, впервые представленной в шестом издании. У нас есть 14 глав и 6 приложений. Первое приложение следует за главой 1. За главой 7 следуют четыре приложения, в том числе новое приложение, посвященное моделированию бизнес-процессов. Шестое приложение следует за главой 8. Эта оптимизированная организация хорошо работала в шестом и седьмом изданиях, поэтому мы решили продолжить ее.
* *Подход к изложению объектно-ориентированного материала.*  Мы сохраняем наш подход к объектно-ориентированной ориентации (ОО) из прошлого издания. Краткие приложения, связанные с объектно-ориентированным подходом, продолжают появляться сразу после соответствующих глав. Приложения ОО выглядят следующим образом: В главе 3 содержится специальный раздел ОО, посвященный управлению проектами ИБ. В главе 7 теперь есть три приложения по объектно-ориентированным мероприятиям: одно посвящено вариантам использования; один на диаграммах последовательности; и один о диаграммах деятельности. (Четвертое приложение к главе 7 посвящено нотации управления бизнес-процессами, которая не является частью UML, хотя и управляется группой управления объектами (OMG).) В главе 8 есть специальный раздел, посвященный объектно-ориентированному проектированию баз данных. Обоснование этой организации такое же, как и в прошлом: четко разделить структурированный и объектно-ориентированный подходы, чтобы преподаватели, не преподающие ОО, могли его обойти. С другой стороны, преподаватели, которые хотят познакомить своих учеников с объектной ориентацией, теперь могут делать это с минимальными усилиями, затрачиваемыми на поиск соответствующего ОО-материала.
* *Обновлены иллюстрации технологии.*  Снимки экрана были обновлены по всему тексту, чтобы показать примеры использования последних версий программ и сред разработки в Интернете (включая последние версии .NET, Visio и Microsoft Office) и дизайнов пользовательского интерфейса. Многие ссылки на веб-сайты предоставляются студентам, чтобы быть в курсе технологических тенденций, которые влияют на анализ и проектирование информационных систем.

Темы *современного системного анализа и проектирования*

1. Разработка систем прочно укоренилась в организационном контексте. Успешному системному аналитику требуется широкое понимание организаций, организационной культуры и организационных операций.
2. Разработка систем – это практическая область. Освещение текущей практики, а также принятых концепций и принципов имеет важное значение в учебнике.
3. Системная разработка — это профессия. Стандарты практики, чувство непрерывного личностного развития, этика, а также уважение и сотрудничество с работой других являются общими темами в учебнике.
4. Развитие систем значительно изменилось с бурным ростом баз данных, архитектур систем, управляемых данными, быстрой разработкой, Интернетом и гибкими методологиями. Разработка систем и управление базами данных могут и должны преподаваться в строго скоординированной манере. Текст совместим с текстом баз данных Хоффера, Рамеша и Топи, Modern Database Management, Eleventh Edition, также опубликованным Pearson. Надлежащая увязка этих двух учебников является стратегической возможностью для удовлетворения потребностей академической области ИС.
5. Успех в системном анализе и проектировании требует не только навыков работы с методологиями и методами, но и навыков управления проектами для управления временем, ресурсами и рисками. Таким образом, анализ и проектирование систем обучения требует глубокого понимания процесса, а также методов и результатов профессии.

Учитывая эти темы, в этом учебнике подчеркивается следующее:

1. Перспективы бизнеса, а не технологии
2. Роль, обязанности и образ мышления системного аналитика, а также менеджера системных проектов, а не программиста или бизнес-менеджера
3. Методы и принципы разработки систем, а не конкретные инструменты или связанные с инструментами навыки в этой области

**Отличительные особенности**

Ниже приведены некоторые отличительные особенности *современного системного анализа и проектирования*:

1. Эта книга организована параллельно с текстом баз данных Хоффера, Рамеша и Топи «*Современное управление базами* данных, двенадцатое издание» (2016 г.), который будет способствовать согласованности структур, определений, методов, примеров и обозначений для лучшей поддержки системного анализа и проектирования, а также курсов по базам данных, использующих оба текста. Несмотря на стратегическую совместимость между этим текстом и *«Современным управлением базами данных*», каждая из этих книг предназначена для того, чтобы оставаться лидером рынка.
2. Обоснование разработки систем в типичной архитектуре для систем в современных организациях, включая управление базами данных и веб-системы.
3. Четкая связь всех аспектов описания и моделирования систем — процесса, принятия решений и моделирования данных — во всеобъемлющий и совместимый набор подходов к системному анализу и проектированию. Такой широкий охват необходим для того, чтобы студенты понимали расширенные возможности многих методологий и инструментов разработки систем, которые автоматически генерируют большой процент кода из проектных спецификаций.
4. Широкий охват устных и письменных коммуникативных навыков, включая системную документацию, управление проектами, управление командами и различные стратегии разработки и приобретения систем (например, жизненный цикл, прототипирование, объектная ориентация, совместная разработка приложений [JAD], системная реинжиниринг и гибкие методологии).
5. Рассмотрение стандартов методологий системного анализа и платформ, на которых проектируются системы.
6. Обсуждение разработки и внедрения систем в контексте управления изменениями, стратегий преобразования и организационных факторов принятия систем.
7. Пристальное внимание к человеческому фактору при проектировании систем, которое подчеркивает удобство использования как в ситуациях с символьным, так и с графическим пользовательским интерфейсом.
8. Проиллюстрированы продукты визуальной разработки и выделены текущие технологические ограничения.
9. В текст включена отдельная глава, посвященная обслуживанию систем. Учитывая тип работы, на которую многие выпускники впервые соглашаются, и большую установленную базу систем, эта глава охватывает важную и часто игнорируемую тему в текстах по системному анализу и проектированию.

Педагогические особенности

Педагогические особенности *современного системного анализа и проектирования* усиливают и применяют ключевое содержание книги.

**три иллюстративных вымышленных случая**

The text features three fictional cases, described below.

*Мебель Пайн-Вэлли (PVF):* В дополнение к демонстрации веб-сайта электронных покупок для бизнеса, для иллюстрации ключевых моментов используется несколько других мероприятий по разработке систем от PVF. PVF представлен в главе 3 и повторяется на протяжении всей книги. По мере того, как представляются ключевые концепции жизненного цикла разработки систем, они применяются и иллюстрируются этим описательным случаем. Например, в главе 5 мы рассмотрим, как PVF планирует проект разработки системы отслеживания клиентов. Значок поля указывает на расположение сегментов дела.

*Hoosier Burger (HB):* Этот второй иллюстративный случай представлен в главе 7 и пересматривается на протяжении всей книги. HB — вымышленный ресторан быстрого питания в Блумингтоне, штат Индиана. Мы используем этот случай, чтобы проиллюстрировать, как аналитики будут разрабатывать и внедрять автоматизированную систему заказа еды. Значок поля указывает на расположение сегментов дела.

*Petrie Electronics:* Эта вымышленная розничная компания по производству электроники используется в качестве расширенного проекта в конце 12 из 14 глав, начиная с главы 2. Этот случай, предназначенный для воплощения концепций глав в жизнь, иллюстрирует, как компания инициирует, планирует, моделирует, проектирует и внедряет систему лояльности клиентов. Вопросы для обсуждения включены для развития критического мышления и участия класса. Предлагаемые решения вопросов для обсуждения приведены в Руководстве для инструктора.

**Материал в конце главы**

Мы разработали обширный выбор материалов в конце главы, которые предназначены для различных стилей обучения и преподавания.

1. *Краткое содержание главы.* Обзор основных тем главы и предварительный просмотр связи текущей главы с будущими.
2. *Ключевые термины.* Разработанный как функция самопроверки, учащиеся сопоставляют каждый ключевой термин в главе с определением.
3. *Контрольные вопросы.* Проверьте понимание учащимися ключевых понятий.
4. *Задачи и упражнения.* Проверяйте аналитические способности учащихся и требуйте от них применения ключевых понятий.
5. *Полевые учения.* Дайте студентам возможность изучить практику системного анализа и проектирования в организациях.
6. *Определения терминов маржи.* Каждый ключевой термин и его определение появляются в марке. Глоссарии терминов и аббревиатур приводятся в конце книги.
7. *Ссылки.* Список литературы расположен в конце каждой главы. Общее количество ссылок в этом тексте составляет более 100 книг, журналов и веб-сайтов, которые могут предоставить студентам и преподавателям дополнительное освещение тем.

Использование этого текста

Как указывалось ранее, эта книга предназначена для основных курсов по системному анализу и проектированию. Он может быть использован в односеместровом курсе по системному анализу и проектированию или в течение двух четвертей (сначала в системном анализе, а затем в курсе системного проектирования). Поскольку текст этой книги аналогичен «*Современному управлению* базами данных», главы из этой книги и из *«Современного управления базами данных*» можно использовать в различных последовательностях, подходящих для вашей учебной программы. Книга будет принята, как правило, в бизнес-школах или факультетах, а не в программах по информатике. Прикладные программы информатики или компьютерных технологий также могут принять книгу.

Типичный преподаватель, который найдет эту книгу наиболее интересной, — это тот, кто

* имеет практическую, а не техническую или теоретическую направленность;
* имеет представление о базах данных и системах, использующих базы данных; и
* использует практические проекты и упражнения в своих курсах.

В частности, эта книга особенно привлечет академические программы, которые пытаются лучше связать свои курсы по системному анализу и проектированию, а также по базам данных как часть всестороннего понимания разработки систем.

План книги, как правило, следует жизненному циклу разработки систем, что позволяет логически развивать темы; однако в нем подчеркивается, что также используются различные подходы (например, моделирование и итеративная разработка), поэтому то, что кажется логической прогрессией, часто является более циклическим процессом. В первой части представлен обзор разработки систем и предварительный просмотр оставшейся части книги. Первая часть также знакомит студентов со многими источниками программного обеспечения, которые они могут использовать для создания своих систем и управления проектами. Остальные четыре части обеспечивают подробное освещение пяти этапов жизненного цикла разработки генетических систем, перемежая их соответствующим образом альтернативами SDLC. Некоторые главы могут быть пропущены в зависимости от ориентации преподавателя или опыта студентов. Например, главу 3 (Управление проектом информационных систем) можно пропустить или быстро просмотреть, если студенты прошли курс по управлению проектами. Глава 4 (Идентификация и выбор проектов разработки систем) может быть пропущена, если преподаватель хочет сделать акцент на разработке систем после определения проектов или если для курса доступно менее 15 недель. Главы 8 (Структурирование системных требований к данным) и 9 (Проектирование баз данных) могут быть пропущены или быстро отсканированы (в качестве напоминания), если студенты уже подробно изучили эти темы в предыдущем курсе по базам данных или структурам данных. Разделы об объектной ориентации в главах 3, 7 и 8 можно пропустить, если преподаватели хотят избежать объектно-ориентированных тем. Наконец, главу 14 (Поддержка/сопровождение информационных систем) можно пропустить, если эти темы выходят за рамки вашего курса.

Поскольку материал представлен в рамках проекта разработки систем, не рекомендуется пытаться использовать главы не по порядку, за некоторыми исключениями: глава 9 (Проектирование баз данных) может преподаваться после глав 10 (Проектирование форм и отчетов) и 11 (Проектирование поверхностей и диалогов), но главы 10 и 11 должны преподаваться последовательно.

Благодарности

Авторы были благословлены значительной помощью со стороны многих людей по всем аспектам подготовки этого текста и дополнений к нему. Мы, конечно, несем ответственность за то, что в конечном итоге появляется между обложками, но идеи, исправления, вклады и подталкивание других значительно улучшили нашу рукопись. На протяжении многих лет десятки людей просматривали различные издания этого учебника. Их вклад стимулировал нас, часто побуждая нас включать новые темы и инновационную педагогику. Мы высоко ценим усилия многих преподавателей и практикующих системных аналитиков, которые ознакомились с этим текстом.

Мы выражаем особую благодарность Джереми Александеру (Jeremy Alexander), который сыграл важную роль в концептуализации и написании функции PVF WebStore, которая появляется в главах с 4 по 14. Добавление этой функции помогло сделать эти главы более современными и инновационными. Мы также хотели бы поблагодарить Джеффа Дженкинса (Jeff Jenkins) из Университета имени Бригама Янга (Brigham Young) за помощь в создании снимков экрана Visual Basic в текущем выпуске.

Мы также хотели бы поблагодарить Атиша Синху (Atish Sinha) из Университета Висконсин-Милуоки (University of Wisconsin-Milwaukee) за написание оригинальной версии некоторых работ по объектно-ориентированному анализу и проектированию. Доктор Синха, который преподает эту тему в течение нескольких лет как студентам, так и студентам MBA, выполнил сложное задание с творческим подходом и сотрудничеством.

Мы также в долгу перед нашими студентами бакалавриата и MBA, которые дали нам много полезных комментариев при работе с черновиками этого текста, и мы благодарим Фреда Макфаддена (Университет Колорадо, Колорадо-Спрингс), Мэри Прескотт (Университет Южной Флориды), Рамеша Венкатарамана (Университет Индианы) и Хейкки Топи (Университет Бентли) за их помощь в координации этого текста с сопутствующей книгой. *Современное управление базами данных*, также от Pearson Education.

Наконец, нам посчастливилось работать с большим количеством творческих и проницательных людей в Pearson, которые внесли большой вклад в разработку, форматирование и производство этого текста. На нас произвело большое впечатление их отношение к этому тексту и к рынку образования в области ИС. В число этих людей входят: Николь Сэм (редактор по закупкам), Нирадж Бхалла (старший редактор-спонсор), Оливия Виньоне (помощник редактора), Илен Кан (менеджер проекта). Мы также хотели бы поблагодарить Джорджа Джейкобса и команду Integra Software Services, Inc.

На написание этого текста ушли тысячи часов времени у авторов и у всех людей, перечисленных ранее. Хотя наши имена будут явно связаны с этой книгой, мы знаем, что большая заслуга принадлежит людям и организациям, перечисленным здесь, за любой успех, которого она может достичь. Важно, чтобы читатель признал всех людей и организации, которые были привержены подготовке и выпуску этой книги.

*Joseph S. Valacich,* Tucson, Arizona

*Joey F. George,* Ames, Iowa

Компания Pearson выражает благодарность и признательность Петтеру Дессне (Университет Бороса), Саймону Ву (Университет Макао), Атари Альвасель (Университет короля Сауда) и Лауре Лаппалайнен (Университет Ваасы) за вклад в издание Global Edition, а также Бернду Шенку (Университет Лихтенштейна), Свену Рему (WHU – Школа менеджмента им. Отто Байсхайма), Дженнифер Уилби (Университет Халла), Фабиану Нг (Политехнический институт Нги Энн) и Жоао Баптисте (Университет Уорика) за рецензирование Global Edition.

# Часть 1 Основы разработки систем

## Обзор

Вы начинаете путешествие, которое позволит вам опираться на каждый аспект вашего образования и опыта. Стать системным аналитиком — это не цель; это путь к богатой и разнообразной карьере, которая позволит вам тренироваться и продолжать развивать широкий спектр талантов. Мы надеемся, что эта вводная часть текста поможет вам открыть глаза на возможности области системного анализа и проектирования, а также на увлекательный характер системной работы.

Глава 1 покажет вам, что такое системный анализ и проектирование и как они развивались за последние несколько десятилетий. По мере того, как предприятия и системы становятся все более изощренными и сложными, все большее внимание уделяется скорости в системном анализе и проектировании. Разработка систем начиналась как искусство, но большинство бизнесменов вскоре поняли, что это не является разумным долгосрочным решением для разработки систем для поддержки бизнес-процессов. Разработка систем стала более структурированной и больше похожей на проектирование, а менеджеры подчеркнули важность планирования, управления проектами и документации. Сейчас мы наблюдаем реакцию на эксцессы во всех трех этих областях, и акцент сместился на гибкую разработку. Эволюция системного анализа и проектирования, а также нынешнее внимание к гибкости объясняются в главе 1. Однако также важно помнить, что системный анализ и проектирование существуют в многогранном организационном контексте, в котором участвуют другие члены организации и внешние стороны. Понимание разработки систем требует понимания не только каждой техники, инструмента и метода, но и того, как эти элементы дополняют и поддерживают друг друга в организационной среде.

Читая эту книгу, вы также обнаружите, что область системного анализа и проектирования постоянно адаптируется к новым ситуациям благодаря твердой приверженности постоянному совершенствованию. Наша цель в этой книге состоит в том, чтобы предоставить вам мозаику навыков, необходимых для эффективной работы в любой среде, где вы можете оказаться, вооружившись знаниями, чтобы определить лучшие практики для этой ситуации и эффективно аргументировать их.

В главе 2 представлено введение во многие источники, из которых можно получить программное обеспечение и программные компоненты. В те времена, когда системный анализ и проектирование были искусством, все системы писались с нуля штатными экспертами. У бизнеса не было выбора. Теперь нет никаких оправданий для внутренней разработки, поэтому становится крайне важным, чтобы системные аналитики понимали индустрию программного обеспечения и множество различных источников программного обеспечения. Глава 2 представляет собой начальную карту ландшафта индустрии программного обеспечения и объясняет большинство из множества вариантов, доступных системным аналитикам.

В главе 3 рассматривается фундаментальная характеристика жизни системного аналитика: работа в рамках проектов с ограниченными ресурсами. Вся работа, связанная с системами, требует внимания к срокам, работы в рамках бюджетов и координации работы разных людей. Сама природа жизненного цикла разработки систем (SDLC) подразумевает системный подход к проекту, который представляет собой группу взаимосвязанных действий, ведущих к конечному результату. Проекты должны быть спланированы, начаты, выполнены и завершены. Запланированная работа проекта должна быть представлена таким образом, чтобы все заинтересованные стороны могли ее рассмотреть и понять. В вашей работе в качестве системного аналитика вам придется работать в рамках графика и других планов проекта, и поэтому важно понимать процесс управления, контролирующий вашу работу.

Наконец, часть I знакомит с делом Petrie Electronics. Случай Петри помогает продемонстрировать, как то, что вы узнаете в каждой главе, может вписаться в практическую организационную ситуацию. Рассмотрение дела начинается после главы 2; остальные главы книги по главу 13 имеют соответствующую часть дела. Первый раздел знакомит с компанией и ее существующими информационными системами. Это введение дает представление о Petrie, которое поможет вам более полно понять компанию, когда мы рассмотрим требования и дизайн новых систем в последующих разделах.

## Глава 1 Окружение разработки систем

### Вступление

Анализ и проектирование информационных систем — это сложная, сложная и стимулирующая организация, которую команда специалистов по бизнесу и системам использует для разработки и обслуживания компьютерных информационных систем. Несмотря на то, что достижения в области информационных технологий постоянно открывают новые возможности, анализ и проектирование информационных систем осуществляется с организационной точки зрения. Организация может состоять из целого предприятия, определенных отделов или отдельных рабочих групп. Организации могут реагировать на проблемы и возможности и предвидеть их за счет инновационного использования информационных технологий. Таким образом, анализ и проектирование информационных систем представляют собой процесс совершенствования организации. Системы строятся и перестраиваются для получения организационных выгод. Выгоды возникают в результате создания добавленной стоимости в процессе создания, производства и поддержки продуктов и услуг организации. Таким образом, анализ и проектирование информационных систем основаны на вашем понимании целей, структуры и процессов организации, а также на ваших знаниях о том, как использовать информационные технологии в своих интересах.

В нынешней бизнес-среде Интернет, особенно Всемирная паутина, прочно интегрирован в способ ведения бизнеса организации. Хотя вы, вероятно, наиболее знакомы с маркетингом, осуществляемым в Интернете и на сайтах розничной торговли в Интернете, таких как eBay или Amazon.com, подавляющее большинство бизнес-использования Интернета — это приложения для бизнеса. Эти приложения охватывают весь спектр всего, что делают предприятия, включая передачу заказов и платежей поставщикам, выполнение заказов и сбор платежей от клиентов, поддержание деловых отношений и создание электронных торговых площадок, где предприятия могут делать покупки в Интернете для получения лучших предложений на ресурсы, необходимые им для сборки своих продуктов и услуг. Несмотря на то, что в наши дни Интернет, кажется, пронизывает бизнес, важно помнить, что многие ключевые аспекты бизнеса — предложение продукта или услуги для продажи, сбор оплаты, оплата труда сотрудников, поддержание отношений с поставщиками и клиентами — не изменились. Понимание бизнеса и того, как он функционирует, по-прежнему является ключом к успешной разработке систем, даже в быстро меняющейся, технологичной среде, в которой сегодня находятся организации.

Карьера в области информационных технологий (ИТ) предоставляет вам прекрасную возможность оказать значительное и заметное влияние на бизнес. Спрос на квалифицированных специалистов в области информационных технологий растет. По данным Бюро статистики труда США, в период с 2010 по 2020 год профессиональная ИТ-рабочая сила вырастет более чем на 22 процента (Thibodeau, 2012). Самый быстрый рост пришелся на разработчиков программного обеспечения (32%) и администраторов баз данных (31%). Один из конкретных аспектов индустрии информационных технологий, облачные вычисления, создали почти 14 миллионов технологий и связанных с ними рабочих мест в период с 2012 по 2015 год (McDougall, 2012). Начиная с этого года, годовой доход от облачных вычислений составит более 1,1 триллиона долларов США. И рост будет глобальным: количество рабочих мест в сфере облачных вычислений в Бразилии увеличится на 186 процентов, количество рабочих мест в Китае и Индии почти удвоится, а количество рабочих мест, связанных с облачными технологиями, увеличится на 66 процентов в Соединенных Штатах. (Подробнее об облачных вычислениях см. в главе 2.) С проблемами и возможностями, связанными с быстрым развитием технологий, трудно представить себе более захватывающий выбор карьеры, чем информационные технологии, а системный анализ и проектирование являются большой частью ИТ-ландшафта. Кроме того, анализ и проектирование информационных систем даст вам возможность понять организации на глубине и широте, на достижение которых в других профессиях может уйти еще много лет.

Важным (но не единственным) результатом системного анализа и проектирования является **прикладное программное обеспечение, программное обеспечение**, предназначенное для поддержки конкретной организационной функции или процесса, такого как управление запасами, расчет заработной платы или анализ рынка. В дополнение к прикладному программному обеспечению, общая информационная система включает в себя аппаратное и системное программное обеспечение, на котором работает прикладное программное обеспечение, документацию и учебные материалы, конкретные рабочие роли, связанные с общей системой, элементы управления и людей, которые используют программное обеспечение, а также их методы работы. Несмотря на то, что мы рассмотрим все эти различные аспекты всей системы, мы сделаем упор на разработку программного обеспечения для приложений — вашу основную ответственность как системного аналитика.

В первые годы вычислительной техники анализ и проектирование считались искусством. Теперь, когда потребность в системах и программном обеспечении стала настолько велика, люди в промышленности и научных кругах разработали методы работы, которые делают анализ и разработку дисциплинированным процессом. Наша цель состоит в том, чтобы помочь вам развить знания и навыки, необходимые для понимания и следования таким процессам разработки программного обеспечения. Центральное место в процессах разработки программного обеспечения (и в этой книге) занимают различные методологии, методы и инструменты, которые были разработаны, протестированы и широко используются на протяжении многих лет, чтобы помочь таким людям, как вы, в системном анализе и проектировании.

Методологии — это комплексные, многоступенчатые подходы к разработке систем, которые будут направлять вашу работу и влиять на качество вашего конечного продукта — информационной системы. Методология, принятая организацией, будет соответствовать ее общему стилю управления (например, ориентация организации на управление на основе консенсуса будет влиять на выбор методологии разработки систем). Большинство методологий включают в себя несколько методов разработки.

Методы — это конкретные процессы, которым вы, как аналитик, будете следовать, чтобы убедиться, что ваша работа хорошо продумана, завершена и понятна другим членам вашей проектной группы. Методы обеспечивают поддержку широкого круга задач, включая проведение тщательных собеседований для определения того, что должна делать ваша система, планирование и управление действиями в проекте разработки систем, построение диаграммы логики системы и разработку отчетов, которые будет генерировать ваша система.

Инструменты, как правило, представляют собой компьютерные программы, которые упрощают использование и извлекают выгоду из методов, а также добросовестно следуют рекомендациям общей методологии разработки. Чтобы быть эффективными, методы и инструменты должны соответствовать методологии разработки систем организации. Методы и инструменты должны облегчать разработчикам систем выполнение шагов, предусмотренных методологией. Эти три элемента — методологии, методы и инструменты — работают вместе, чтобы сформировать организационный подход к системному анализу и проектированию (см. рис. 1-1).

Хотя многие люди в организациях отвечают за системный анализ и проектирование, в большинстве организаций **системный** аналитик несет основную ответственность. Когда вы начнете свою карьеру в системной разработке, вы, скорее всего, начнете как системный аналитик или как программист с некоторыми обязанностями по системному анализу. Основная роль системного аналитика заключается в изучении проблем и потребностей организации, чтобы определить, как люди, методы и информационные технологии могут быть наилучшим образом объединены для достижения улучшений в организации. Системный аналитик помогает пользователям системы и другим бизнес-менеджерам определить свои требования к новым или улучшенным информационным услугам. Таким образом, системный аналитик является агентом изменений и инноваций.

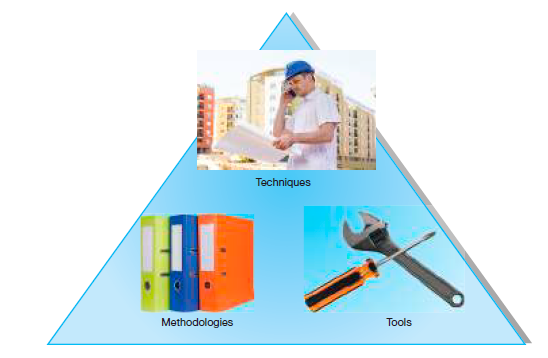


Рисунок 1.1 Организационный подход к системному анализу и проектированию основан на методологиях, методах и инструментах. Источники: вверху: Mitarart/Fotolia; слева: Лев/Фотолия; справа: PaulPaladin/Fotolia

### Современный подход к системному анализу и проектированию

Анализ и проектирование компьютерных информационных систем началось в 1950-х годах. С тех пор среда разработки сильно изменилась, обусловленная организационными потребностями, а также быстрыми изменениями в технологических возможностях компьютеров. В 1950-х годах разработка была сосредоточена на процессах, выполняемых программным обеспечением. Поскольку вычислительная мощность была критически важным ресурсом, главной целью стала эффективность процессов. Компьютеры были большими, дорогими и не очень надежными. Акцент был сделан на автоматизации существующих процессов, таких как закупки или оплата, часто в рамках отдельных отделов. Все приложения должны были быть разработаны на машинном языке или языке ассемблера, и они должны были быть разработаны с нуля, потому что не было индустрии программного обеспечения. Поскольку компьютеры были очень дорогими, компьютерная память также была на высоте, поэтому разработчики систем сохранили как можно больше памяти для хранения данных.

Первые процедурные языки программирования третьего поколения стали доступны только в начале 1960-х годов. Компьютеры по-прежнему были большими и дорогими, но в 1960-х годах произошли важные прорывы в технологиях, которые позволили разработать меньшие, более быстрые и менее дорогие компьютеры — мини-компьютеры — и начало индустрии программного обеспечения. Большинство организаций по-прежнему разрабатывают свои приложения с нуля, используя свой собственный штат разработчиков. Разработка систем была скорее искусством, чем наукой. Однако этот взгляд на разработку систем начал меняться в 1970-х годах, когда организации начали осознавать, насколько дорого обходится разработка специализированных информационных систем для каждого приложения. Разработка систем стала более дисциплинированной, поскольку многие люди работали над тем, чтобы сделать ее более похожей на инженерию. Ранние системы управления базами данных, использующие иерархические и сетевые модели, помогли дисциплинировать хранение и извлечение данных. Разработка систем управления базами данных помогла сместить акцент при разработке систем с процессов на данные.

1980-е годы были отмечены крупными прорывами в вычислительной технике в организациях, поскольку микрокомпьютеры стали ключевыми организационными инструментами. Индустрия программного обеспечения значительно расширилась, поскольку все больше и больше людей начали писать готовое программное обеспечение для микрокомпьютеров. Разработчики стали писать все больше и больше приложений на языках четвертого поколения, которые, в отличие от процедурных языков, инструктировали компьютер о том, что делать, а не как это делать. Инструменты автоматизированной разработки программного обеспечения (CASE) были разработаны, чтобы сделать работу разработчиков систем более простой и последовательной. По мере того, как вычислительные машины становились все меньше, быстрее и дешевле, а операционные системы для компьютеров переходили от интерфейсов с подсказками к интерфейсам на основе окон и значков, организации переходили к приложениям с большим количеством графики. Организации разрабатывали меньше программного обеспечения собственными силами и покупали относительно больше у поставщиков программного обеспечения. Работа разработчика систем прошла путь от строителя до интегратора.

Среда разработки систем конца 1990-х годов была ориентирована на системную интеграцию. Разработчики использовали визуальные среды программирования, такие как PowerBuilder или Visual Basic, для разработки пользовательских интерфейсов для систем, работающих на клиент-серверных платформах. База данных, которая может быть реляционной или объектно-ориентированной и которая может быть разработана с использованием программного обеспечения таких фирм, как Oracle, Microsoft или Ingres, находилась на сервере. Во многих случаях логика приложения находилась на одном сервере. В качестве альтернативы организация может решить приобрести всю свою корпоративную систему у таких компаний, как SAP AG или Oracle. Корпоративные системы — это большие, сложные системы, состоящие из ряда независимых системных модулей. Разработчики собирают системы, выбирая и внедряя конкретные модули. Начиная с середины 1990-х годов, все больше и больше усилий по разработке систем было сосредоточено на Интернете, особенно в Интернете.

В настоящее время основное внимание по-прежнему уделяется разработке систем для Интернета, а также для внутренних сетей и внешних сетей фирм. Как и в случае с традиционными системами, интернет-разработчики теперь полагаются на компьютерные инструменты для ускорения и упрощения разработки веб-систем. Многие инструменты CASE непосредственно поддерживают разработку веб-приложений. Все чаще реализация систем предполагает трехуровневое проектирование, с базой данных на одном сервере, приложением на втором сервере, а клиентской логикой, расположенной на пользовательских машинах. Еще одним важным событием является переход к беспроводным компонентам системы. Беспроводные устройства могут получать доступ к веб-приложениям практически из любого места. Наконец, продолжается тенденция к сборке систем из программ и компонентов, купленных с полки. Во многих случаях организации не разрабатывают приложение собственными силами. Они даже не запускают приложение собственными силами, предпочитая вместо этого использовать приложение по отдельности, получая к нему доступ через облако.

### Разработка информационных систем и жизненный цикл разработки систем

Большинство организаций считают полезным использовать стандартный набор шагов, называемый **методологией** разработки систем, для разработки и поддержки своих информационных систем. Как и многие процессы, разработка информационных систем часто следует жизненному циклу. Например, коммерческий продукт следует жизненному циклу в том смысле, что он создается, тестируется и выводится на рынок. Его продажи увеличиваются, достигают пика и падают. Наконец, продукт удаляется с рынка и заменяется чем-то другим. Жизненный цикл разработки систем **(SDLC)** является общей методологией разработки систем во многих организациях; он имеет несколько этапов, которые отмечают прогресс системного анализа и проектирования. Каждый автор учебника и организация, занимающаяся разработкой информационных систем, использует несколько иную модель жизненного цикла, включающую от 3 до почти 20 идентифицируемых фаз.

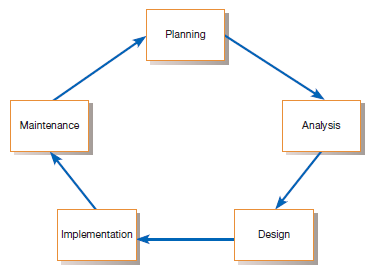


Рисунок 1.2 Жизненный цикл разработки систем

Жизненный цикл можно рассматривать как циклический процесс, в котором окончание срока полезного использования одной системы приводит к началу другого проекта, который разработает новую версию или полностью заменит существующую систему (см[. рис. 1-2).](#bookmark0)  На первый взгляд жизненный цикл кажется последовательно упорядоченным набором фаз, но это не так. Конкретные шаги и их последовательность должны быть адаптированы по мере необходимости для проекта в соответствии с подходами к управлению. Например, на любом этапе SDLC проект может вернуться к более ранней фазе, если это необходимо. Аналогичным образом, если коммерческий продукт не показывает хороших результатов сразу после его внедрения, он может быть временно удален с рынка и улучшен перед повторным введением. В SDLC также можно выполнять некоторые действия на одном этапе параллельно с некоторыми действиями другого этапа. Иногда жизненный цикл является итеративным; то есть фазы повторяются по мере необходимости до тех пор, пока не будет найдена приемлемая система. Некоторые люди считают, что жизненный цикл представляет собой спираль, в которой мы постоянно проходим через фазы на разных уровнях детализации (см. [рис. 1-3).](#bookmark0) Жизненный цикл разработки систем, используемый в организации, представляет собой упорядоченный набор действий, проводимых и планируемых для каждого проекта разработки. Навыки, требуемые от системного аналитика, применимы ко всем моделям жизненного цикла. Программное обеспечение является наиболее очевидным конечным продуктом жизненного цикла; Другие важные результаты включают документацию о системе и о том, как она была разработана, а также обучение пользователей.

Каждая средняя и крупная корпорация и каждый производитель программного обеспечения на заказ будут иметь свой собственный конкретный жизненный цикл или методологию разработки систем (см. [рис. 1-4).](#bookmark0)  Даже если конкретная методология не похожа на цикл, а на [рисунке 1-4](#bookmark0)  - нет, вы, вероятно, обнаружите, что многие шаги SDLC выполняются и используются методы и инструменты SDLC. Изучение системного анализа и проектирования на основе подхода жизненного цикла.

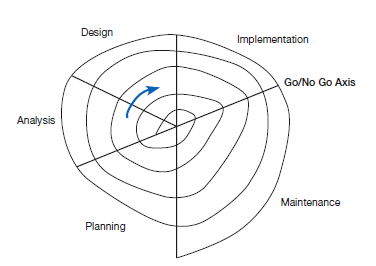


Рисунок 1.3 Модель развития



**Рисунок 1.4** Отправление Министерства юстиции США Жизненный цикл разработки

Когда вы начнете свою первую работу, вы, скорее всего, потратите несколько недель или месяцев на изучение SDLC вашей организации и связанных с ней методологий, методов и инструментов. Чтобы сделать эту книгу как можно более общей, мы следуем довольно общей модели жизненного цикла, как более подробно описано на рисунке 1-5. Обратите внимание, что наша модель является круговой. Мы используем этот SDLC в качестве одного из примеров методологии, но, что более важно, как способ упорядочить темы системного анализа и проектирования. Таким образом, то, что вы узнаете из этой книги, вы можете применить практически к любому жизненному циклу, которому вы могли бы следовать. По мере того, как мы описываем этот SDLC на протяжении всей книги, вы увидите, что каждый этап имеет определенные результаты и результаты, которые передают важную информацию другим этапам. В конце каждого этапа проект разработки систем достигает важной вехи, и по мере получения результатов они часто проверяются сторонами, не входящими в проектную группу. В оставшейся части этого раздела мы приводим краткий обзор каждого этапа SDLC. В конце раздела мы резюмируем это обсуждение в таблице, в которой перечислены основные результаты или результаты каждого этапа SDLC.

Первым этапом в SDLC является **планирование**. На этом этапе кто-то определяет потребность в новой или улучшенной системе. В более крупных организациях это признание может быть частью процесса корпоративного и системного планирования. Изучаются информационные потребности организации в целом, и активно определяются проекты, направленные на удовлетворение этих потребностей. Потребности организации в информационных системах могут быть обусловлены просьбами о решении проблем, связанных с текущими процедурами, желанием выполнить дополнительные задачи или осознанием того, что информационные технологии могут быть использованы для извлечения выгоды из существующих возможностей. Затем эти потребности могут быть расставлены по приоритетам и преобразованы в план для отдела информационных систем, включая график разработки новых основных систем. В небольших организациях (а также в крупных) на определение того, какие системы следует разрабатывать, могут влиять специальные запросы пользователей, представляемые по мере возникновения потребности в новых или усовершенствованных системах, а также формализованный процесс планирования информирования. В любом случае при определении и отборе проекта организация определяет, следует ли выделять ресурсы на развитие или совершенствование каждой рассматриваемой информационной системы. Результатом процесса идентификации и отбора проектов является определение того, какие проекты по разработке систем должны быть предприняты организацией, по крайней мере, с точки зрения первоначального исследования.

На этапе планирования также проводятся два дополнительных важных мероприятия: формальное, но все же предварительное исследование имеющейся проблемы или возможности системы и изложение причин, по которым система должна или не должна разрабатываться организацией. Важнейшим шагом на данном этапе является определение сферы применения предлагаемой системы. Руководитель проекта и первоначальная группа системных аналитиков также разрабатывают конкретный план для предлагаемого проекта, которому команда будет следовать, используя оставшиеся шаги SDLC. Этот базовый план проекта настраивает стандартизированный SDLC и определяет время и ресурсы, необходимые для его выполнения. Формальное определение проекта основано на вероятности того, что отдел информационных систем организации сможет разработать систему, которая решит проблему или воспользуется возможностью и определит, перевешивают ли затраты на разработку системы выгоды, которые она может обеспечить. Окончательная презентация экономического обоснования для продолжения последующих этапов проекта обычно делается руководителем проекта и другими членами команды кому-либо из руководства или специальному комитету по управлению, который решает, какие проекты будет осуществлять организация.

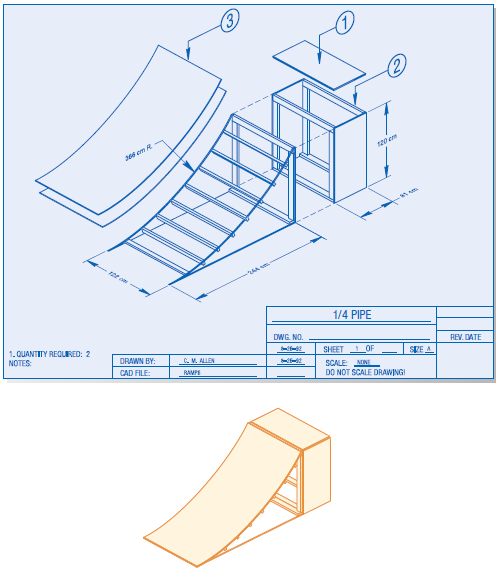
Вторым этапом в SDLC является **анализ**. На этом этапе аналитик тщательно изучает текущие процедуры организации и информационные системы, используемые для выполнения организационных задач. Анализ состоит из двух подфаз. Во-первых, это определение требований. На этом подэтапе аналитики работают с пользователями, чтобы определить, чего пользователи хотят от предлагаемой системы. Процесс определения требований обычно включает в себя тщательное изучение любых существующих систем, ручных и компьютеризированных, которые могут быть заменены или усовершенствованы в рамках проекта. Во второй части анализа аналитики изучают требования и структурируют их в соответствии с их взаимосвязями и устраняют любые избыточности. Результатом этапа анализа является описание (но не детальный проект) альтернативного решения, рекомендованного группой анализа. После того, как рекомендация будет принята теми, у кого есть полномочия на финансирование, аналитики могут начать планировать приобретение любого аппаратного и системного программного обеспечения, необходимого для создания или эксплуатации системы, как это было предложено.

Третьим этапом в SDLC является **проектирование**. Во время проектирования аналитики преобразуют описание рекомендуемого альтернативного решения в логические, а затем физические спецификации системы. Аналитики должны проектировать все аспекты системы, от экранов ввода и вывода до отчетов, баз данных и компьютерных процессов. Затем аналитики должны предоставить физические характеристики системы, которую они разработали, либо в виде модели, либо в виде подробной документации, чтобы направлять тех, кто будет строить новую систему. Та часть процесса проектирования, которая не зависит от какой-либо конкретной аппаратной или программной платформы, называется **логическим проектированием**. Теоретически система может быть реализована на любом аппаратном и системном программном обеспечении. Идея состоит в том, чтобы убедиться, что система функционирует так, как задумано. Логическое проектирование концентрируется на бизнес-аспектах системы и, как правило, ориентировано на высокий уровень специфики.

После того, как общий высокоуровневый дизайн системы разработан, аналитики начинают превращать логические спецификации в физические. Этот процесс называется **физическим проектированием**. В рамках физического проектирования аналитики проектируют различные части системы для выполнения физических операций, необходимых для облегчения сбора, обработки и вывода данных. Это можно сделать разными способами: от создания рабочей модели системы, которая будет реализована, до написания подробных спецификаций, описывающих все различные части системы и то, как они должны быть построены. Во многих случаях рабочая модель становится основой для реальной системы, которая будет использоваться. Во время физического проектирования команда аналитиков должна определить многие физические детали, необходимые для создания окончательной системы, от языка программирования, на котором будет написана система, до системы баз данных, в которой будут храниться данные, до аппаратной платформы, на которой будет работать система. Часто выбор языка, базы данных и платформы уже решается организацией или клиентом, и на этом этапе эти информационные технологии должны быть приняты во внимание при физическом проектировании системы. Конечным продуктом этапа проектирования являются спецификации физической системы в форме, готовой к передаче программистам и другим сборщикам систем для строительства. На рисунке 1-6 показаны различия между логическим и физическим проектированием.

Четвертым этапом SDLC является **реализация**. Спецификации физической системы, будь то в виде подробной модели или в виде подробных письменных спецификаций, передаются программистам в качестве первой части этапа реализации. Во время внедрения аналитики превращают спецификации системы в рабочую систему, которая тестируется, а затем вводится в эксплуатацию. Реализация включает в себя кодирование, тестирование и установку. Во время кодирования программисты пишут программы, из которых состоит система. Иногда код генерируется той же системой, которая использовалась для построения детальной модели системы. Во время тестирования программисты и аналитики тестируют отдельные программы и всю систему, чтобы найти и исправить ошибки. Во время установки новая система становится частью повседневной деятельности организации. Прикладное программное обеспечение устанавливается или загружается на существующее или новое оборудование, а пользователи знакомятся с новой системой и обучаются. Тестирование и монтаж должны планироваться уже на этапе инициирования и планирования проекта; как тестирование, так и установка требуют тщательного анализа, чтобы разработать правильный подход.

Мероприятия по внедрению также включают первоначальную поддержку пользователей, такую как доработка документации, программы обучения и постоянная помощь пользователям. Обратите внимание, что документация и программы обучения дорабатываются во время внедрения; Документация создается на протяжении всего жизненного цикла, а обучение (и обучение) происходит с самого начала проекта. Внедрение может продолжаться до тех пор, пока существует система, потому что постоянная поддержка пользователей также является частью реализации. Однако, несмотря на все усилия аналитиков, менеджеров и программистов, установка не всегда является простым процессом. Многие хорошо спроектированные системы потерпели неудачу из-за того, что процесс установки был неправильным. Даже хорошо спроектированная система может потерпеть неудачу, если реализация не будет хорошо организована. Поскольку проектная группа обычно управляет реализацией, мы подчеркиваем вопросы реализации на протяжении всей этой книги.



**FIgure 1-6**

Дизайн разницы ставок и физический дизайн

(a) Схема Ask (логическое проектирование)

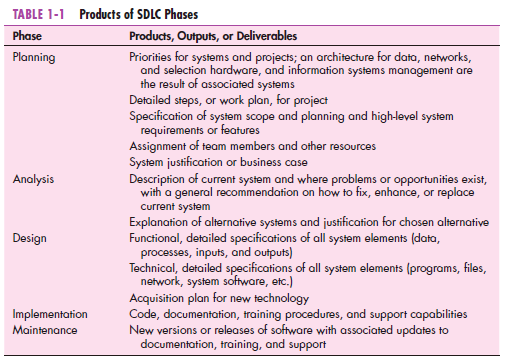
(b) Рампа для скейтборда (физический дизайн)

Пятым и последним этапом SDLC является **техническое обслуживание**. Когда система (включая ее обучение, документацию и поддержку) работает в организации, пользователи иногда сталкиваются с проблемами в том, как она работает, и часто думают о лучших способах выполнения ее функций. Кроме того, потребности организации в отношении системы меняются со временем. При обслуживании программисты вносят изменения, которые запрашивают пользователи, и модифицируют систему в соответствии с меняющимися условиями бизнеса. Эти изменения необходимы для поддержания работоспособности и полезности системы. В каком-то смысле техническое обслуживание — это не отдельная фаза, а повторение других фаз жизненного цикла, необходимых для изучения и внедрения необходимых изменений. Можно думать о техническом обслуживании как о наложении на жизненный цикл, а не как об отдельном этапе. Количество времени и усилий, затрачиваемых на техническое обслуживание, во многом зависит от производительности предыдущих этапов жизненного цикла. Однако неизбежно наступает момент, когда информационная система перестает работать так, как хотелось бы, когда затраты на обслуживание становятся непомерно высокими или когда потребности организации существенно меняются. Такие проблемы указывают на то, что пора приступать к проектированию замены системы, тем самым завершая цикл и запуская жизненный цикл заново. Часто различие между капитальным обслуживанием и новой разработкой не ясно, что является еще одной причиной, по которой техническое обслуживание часто напоминает сам жизненный цикл.

SDLC представляет собой тесно связанный набор фаз, продукты которых питают деятельность на последующих этапах. [В таблице 1-1](#bookmark0) кратко изложены результаты или продукты каждого этапа на основе текстовых описаний. В главах, посвященных этапам SDLC, будут подробно описаны продукты каждого этапа, а также способы разработки продуктов.

На протяжении всего SDLC сам проект разработки систем должен быть тщательно спланирован и управляем. Чем крупнее системный проект, тем больше потребность в управлении проектами. За последние десятилетия было разработано несколько методов управления проектами, и многие из них стали более полезными благодаря автоматизации. В главе 3 содержится более подробное описание методов планирования и управления проектами. Далее мы обсудим некоторые критические замечания в адрес SDLC и представим альтернативы, разработанные для устранения этой критики. Однако сначала мы познакомим вас со специализированным SDLC, который фокусируется на безопасности во время разработки.

Таблица 1.1 Результаты SDLC фазы



### Жизненный цикл разработки специализированных систем

Несмотря на то, что базовый SDLC дает обзор процесса разработки систем, концепция SDLC также может быть применена к очень специфическим аспектам процесса. Как упоминалось ранее, фаза обслуживания может быть описана в терминах SDLC. Другим примером специализированного SDLC является Microsoft Security Development Lifecycle (SDL) (подробнее см. <http://www.microsoft.com/security/sdl/default.aspx> ). Жизненный цикл разработки безопасности показан на рисунке 1-7. Во-первых, обратите внимание на то, что пять основных фаз жизненного цикла разработки (выделены зеленым цветом) не совсем совпадают с пятью фазами SDLC, которые мы будем использовать в этой книге. Три из пяти фаз почти идентичны фазам в нашем SDLC. Microsoft SDL начинается с «требований», что похоже на «анализ»; затем следует этап проектирования, за которым следует реализация. Наш жизненный цикл начинается с планирования и заканчивается техническим обслуживанием. Оба эти этапа характерны для разработки систем в организационном контексте, где закупаемые информационные системы используются внутри организации. Необходимо тщательное планирование, чтобы определить, какие системы будут разработаны для организации. Каждая разработанная система является инвестицией, и если организация инвестирует в отдельную систему, она не может инвестировать в какую-то альтернативную систему или во что-то еще, например, в новый магазин. В конце концов, инвестиционные средства ограничены. Техническое обслуживание также характерно для организационного контекста. После того, как системы перейдут в общее пользование, организация должна получить как можно большую отдачу от этих инвестиций, поэтому важно, чтобы системы работали как можно дольше. Таким компаниям, как Microsoft, которые разрабатывают системы для использования другими, не нужно беспокоиться о внутреннем планировании и обслуживании в жизненных циклах своих продуктов. Вместо этого, поскольку они имеют ограниченный контроль над системами после того, как они были проданы, они озабочены продажей зрелого и надежного продукта. Таким образом, они имеют два этапа после реализации: проверка и выпуск. Проверка включает в себя обеспечение качества продуктов перед их выпуском. Релиз включает в себя все действия, связанные с тем, чтобы сделать продукт доступным для общего использования. Затем обратите внимание на две части SDL, которые предшествуют и следуют за основными фазами разработки: обучение (синим цветом) и реакция (оранжевым). Две вещи делают этот конкретный SDLC специализированным для вопросов безопасности: два уникальных этапа, которые начинают и заканчивают жизненный цикл (обучение и реагирование), и конкретные действия по обеспечению безопасности, связанные с каждым этапом жизненного цикла разработки.

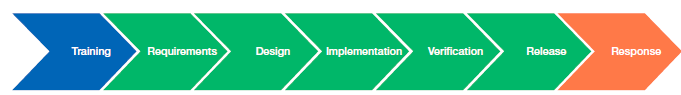


Рисунок 1.7 Жизненный цикл разработки безопасности в Microsoft

Обучение в Microsoft SDL относится к обучению, которое каждый член команды разработчиков получает по основам безопасности и тенденциям в области безопасности. Идея, лежащая в основе обучения — на самом деле, идея специализированного жизненного цикла разработки безопасности — заключается в том, чтобы безопасность стала частью процесса разработки с самого начала, а не внезапно появилась в конце SDLC. Обучая членов команды безопасности и тому, как ее можно решить на протяжении всего жизненного цикла, меры безопасности могут быть встроены в систему на протяжении всей ее разработки. Ответ в конце SDL относится к плану реагирования, разработанному на этапе выпуска. Если есть угроза безопасности для конкретного продукта, то выполняется ранее разработанный план реагирования. Действия, связанные с безопасностью, которые выполняются на протяжении всего жизненного цикла разработки, различаются в зависимости от этапа. Перечисление и объяснение каждого вида деятельности выходит за рамки этой главы, но мы можем привести несколько примеров. Одним из специализированных действий, выполняемых на этапе требований, является отдельный анализ требований, связанных как с безопасностью, так и с конфиденциальностью. Во время проектирования разработчики могут моделировать угрозы для системы и учитывать, как эти угрозы различаются в зависимости от различных вариантов проектирования. Во время реализации члены проектной группы могут проводить статический анализ исходного кода, выискивая угрозы безопасности. Во время проверки они могут проводить динамические анализы. На этапе выпуска члены команды разрабатывают план реагирования на инциденты, упомянутый ранее, и проводят окончательную проверку безопасности. Придерживаясь специализированного жизненного цикла, посвященного безопасности, члены проектной группы могут обеспечить не только решение проблемы безопасности, но и ее планомерное и систематическое решение.

### Сердце процесса разработки систем

SDLC предоставляет удобный способ осмысления процессов, связанных с разработкой систем и организацией этой книги. Различные фазы четко определены, их отношения друг с другом четко определены, а последовательность фаз от одной к другой, от начала до конца, имеет убедительную логику. Во многих отношениях, однако, SDLC является фикцией. Хотя почти все проекты разработки систем придерживаются определенного типа жизненного цикла, точное местоположение действий и конкретная последовательность шагов могут сильно различаться от одного проекта к другому. Современная практика объединяет действия, традиционно считающиеся относящимися к анализу, проектированию и реализации, в единый процесс. Вместо того, чтобы системные требования вырабатывались при анализе, системные спецификации создавались при проектировании, а кодирование и тестирование выполнялись в начале реализации, нынешняя практика объединяет все эти действия в единый процесс анализа-проектирования-кода-тестирования [(рис. 1-8).](#bookmark0) Эти действия лежат в основе разработки систем, как мы предлагаем на [рисунке 1-9.](#bookmark0) Такая комбинация действий типична для современных практик в гибких методологиях. Хорошо известным примером одной из гибких методологий является экстремальное программирование, хотя есть и другие варианты. Мы познакомим вас с гибкими методологиями и экстремальным программированием, но сначала важно, чтобы вы узнали о проблемах традиционного SDLC. Об этих проблемах вы прочтете далее. Затем вы прочитаете об инструментах CASE, гибких методологиях и экстремальном программировании.



Рисунок 1.8 Цикл Анализ-проектирование-код-тест

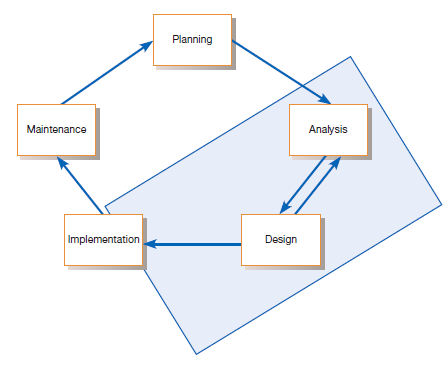


Рисунок 1.9 Сердце разработки систем

#### Традиционный водопадный SDLC процесс

Существует несколько критических замечаний в отношении традиционного подхода к разработке систем, основанного на жизненном цикле; Один из них связан с тем, как организован жизненный цикл. Чтобы лучше понять эти критические замечания, лучше всего увидеть форму, в которой жизненный цикл традиционно был тернист, так называемый водопад [(рис. 1-10).](#bookmark0)  Обратите внимание на то, как поток проекта начинается на этапе планирования и оттуда бежит «вниз» к каждой последующей фазе точно так же, как ручей, стекающий со скалы. Хотя первоначальный разработчик модели водопада, У. В. Ройс, призывал к обратной связи между фазами водопада, эта обратная связь была проигнорирована при реализации (Martin, 1999). Стало слишком заманчиво игнорировать необходимость обратной связи и рассматривать каждый этап как завершенный сам по себе, к которому никогда не возвращаться после завершения.

Традиционно один этап заканчивался, а другой начинался после достижения вехи. Веха обычно принимала форму некоторого результата или заранее определенного результата фазы. Например, конечным результатом проектирования является набор подробных спецификаций физического проектирования. После того, как веха была достигнута и начался новый этап, стало трудно возвращаться назад. Несмотря на то, что бизнес-условия продолжали меняться в процессе разработки, и на аналитиков оказывалось давление со стороны пользователей и других лиц, чтобы они изменили дизайн в соответствии с изменяющимися условиями, аналитикам было необходимо заморозить дизайн в определенный момент и двигаться вперед. Огромное количество усилий и времени, необходимых для реализации конкретного проекта, означало, что внесение изменений в систему после ее разработки было бы очень дорогим. Таким образом, традиционный жизненный цикл водопада обладал свойством привязывать пользователей к требованиям, которые были определены ранее, даже если эти требования могли измениться.

Еще одна критика традиционного водопадного SDLC заключается в том, что роль системных пользователей или клиентов была узко определена (Kay, 2002). Роли пользователей часто передавались на этапы определения или анализа требований проекта, где предполагалось, что все требования могут быть определены заранее. Такое предположение в сочетании с ограниченным участием пользователей усилило тенденцию каскадной модели фиксировать требования слишком рано, даже после изменения условий ведения бизнеса.

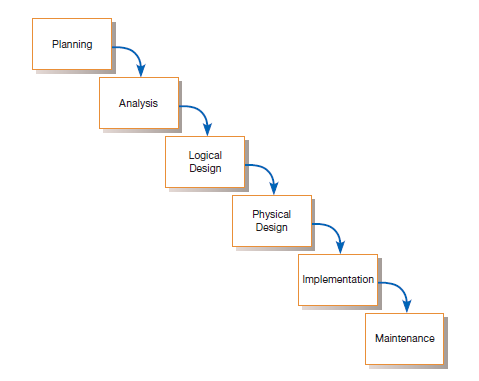


Рисунок 1.10 Традиционный водопадный SDLC процесс

Кроме того, в соответствии с традиционным каскадным подходом туманным и неосязаемым процессам, таким как анализ и проектирование, даются жесткие и быстрые сроки завершения, и успех в подавляющем большинстве случаев измеряется тем, соблюдены ли эти сроки. Сосредоточение внимания на сроках выполнения этапов, а не на получении и интерпретации обратной связи в процессе разработки, приводит к тому, что слишком мало внимания уделяется хорошему анализу и проектированию. Ориентация на сроки приводит к системам, которые не соответствуют потребностям пользователей и требуют обширного обслуживания, что неоправданно увеличивает затраты на разработку. Поиск и устранение проблемы с программным обеспечением после поставки системы часто обходится гораздо дороже, чем ее поиск и устранение во время анализа и проектирования (Griss, 2003). Результатом сосредоточения внимания на сроках, а не на передовой практике, являются ненужные доработки и усилия по обслуживанию, которые являются дорогостоящими. По некоторым оценкам, затраты на техническое обслуживание составляют от 40 до 70 процентов затрат на разработку систем (Dorfman and Thayer, 1997). Учитывая эти проблемы, люди, работающие в области разработки систем, начали искать лучшие способы проведения системного анализа и проектирования.

### Различные подходы к совершенствованию развития

В рамках постоянных усилий по совершенствованию процесса системного анализа и проектирования было разработано несколько различных подходов. Более подробно мы опишем наиболее важные подходы в следующих главах. Попытки сделать разработку систем не искусством, а наукой, обычно называют системной инженерией или *программно-аппаратной инженерией*. Как видно из названий, для разработки систем были применены строгие инженерные методы. Одним из проявлений инженерного подхода являются CASE-инструменты, о которых вы прочтете далее.

#### CASE средства

Другие усилия по совершенствованию процесса разработки систем позволили воспользоваться преимуществами, предлагаемыми самой вычислительной технологией. Результатом стало создание и довольно широкое применение **средств автоматизированной программной инженерии (CASE).** Инструменты CASE были разработаны для внутреннего использования и для продажи несколькими ведущими фирмами, но наиболее известной является серия инструментов Rational от IBM.

Инструменты CASE используются для поддержки широкого спектра мероприятий SDLC. Инструменты CASE могут быть использованы для помощи на нескольких этапах SDLC: идентификация и выбор проекта, инициация и планирование проекта, анализ, проектирование, внедрение и обслуживание. Интегрированная и стандартная база данных, называемая *репозиторием*, является распространенным методом обеспечения интеграции продуктов и инструментов и является ключевым фактором, позволяющим CASE легче управлять более крупными и сложными проектами и беспрепятственно интегрировать данные между различными инструментами и продуктами. Идея центрального хранилища информации о проекте не нова — ручная форма такого репозитория называется *словарем* проекта или *рабочей тетрадью*.

Ниже перечислены основные типы CASE-инструментов:

* Инструменты построения диаграмм позволяют графически представлять системные процессы, данные и структуры управления.
* Компьютерные дисплеи и генераторы отчетов помогают создавать прототипы того, как системы «выглядят и ощущаются». Генераторы дисплеев (или форм) и отчетов облегчают системному аналитику определение требований к данным и взаимосвязей.
* Инструменты анализа автоматически проверяют наличие неполных, несогласованных или неправильных спецификаций в диаграммах, формах и отчетах.
* Центральный репозиторий обеспечивает интегрированное хранение спецификаций, диаграмм, отчетов и информации об управлении проектами.
* Генераторы документации производят техническую и пользовательскую документацию в стандартных форматах.
* Генераторы кода позволяют автоматически генерировать код определения программ и баз данных непосредственно из проектной документации, диаграмм, форм и отчетов.

Таблица 1.2 Примеры использования CASE средств вместе с SDLC

|  |  |  |
| --- | --- | --- |
| SDLC фаза | Ключевые активности | Использование CASE инструментов |
| Идентификация и отбор проекта | Отображение и структурирование высокоуровневой организационной информации | Инструменты построения диаграмм и матриц для создания и структурирования информации |
| Инициация и планирование проекта | Разработка объема и технико-экономического обоснования проекта | Генераторы репозиториев и документации для разработки планов проектов |
| Анализ | Определение и структурирование системных требований | Построение диаграмм для создания моделей процессов, логики и данных |
| Логическое и физическое проектирование | Создание новых проектов систем | Формировать и сообщать генераторы к прототипам; Генераторы анализа и документации для определения спецификации |
| Реализация | Преобразование проектов в информационную систему | Генераторы и анализ кода, генераторы форм и отчетов для разработки системы; генераторы документации для разработки системы; Генераторы документации для разработки системной и пользовательской документации |
| Содержание | Развивайте информационную систему | Все инструменты используются (повторяют жизненный цикл) |

CASE помогает программистам и аналитикам выполнять свою работу эффективнее и результативнее, автоматизируя рутинные задачи. Однако многие организации, использующие инструменты CASE, не используют их для поддержки всех этапов SDLC. Некоторые организации могут широко использовать функции построения диаграмм, но не генераторы кода. [В таблице 1-2](#bookmark0) показано, как CASE обычно используется на каждом этапе SDLC. Существует множество причин, по которым организации предпочитают использовать CASE частично или не использовать его вообще. Эти причины варьируются от отсутствия видения применения CASE ко всем аспектам SDLC до веры в то, что технология CASE не сможет удовлетворить уникальные потребности организации в разработке систем. В некоторых организациях CASE был чрезвычайно успешным, в то время как в других он не был успешным.

### Agile методологии

За эти годы было разработано множество подходов к системному анализу и проектированию. В феврале 2001 года многие из сторонников этих альтернативных подходов встретились в штате Юта и достигли консенсуса по нескольким основополагающим принципам, содержащимся в их различных подходах. Этот консенсус превратился в документ, который они назвали «Манифест Agile» (таблица 1-3). Согласно Фаулеру (2003), гибкие методологии разделяют три ключевых принципа: (1) сосредоточение внимания на адаптивных, а не прогностических методологиях, (2) сосредоточение внимания на людях, а не на ролях, и (3) сосредоточение внимания на самоадаптивных процессах.

Группа Agile Methodologies утверждает, что методы разработки программного обеспечения, адаптированные из инженерии, как правило, не соответствуют реальной разработке программного обеспечения (Fowler, 2003). В инженерных дисциплинах, таких как гражданское строительство, требования, как правило, хорошо понятны. После того, как творческая и сложная работа по проектированию завершена, строительство становится очень предсказуемым. Кроме того, на строительство может приходиться до 90 процентов от общего объема проектных усилий. С другой стороны, для программного обеспечения требования редко бывают хорошо понятными, и они постоянно меняются в течение жизненного цикла проекта. На строительство может приходиться всего 15 процентов от общего объема проектных работ, а на проектирование - до 50 процентов. Применение методов, которые хорошо работают для предсказуемых, стабильных проектов, таких как строительство мостов, как правило, не работает хорошо для гибких, тяжелых проектов, таких как написание программного обеспечения, говорят сторонники гибкой методологии. Необходимы методологии, которые охватывают изменения и способны справиться с отсутствием предсказуемости. Одним из механизмов борьбы с отсутствием предсказуемости, который разделяют все гибкие методологии, является итеративная разработка (Martin, 1999). Итеративная разработка фокусируется на частом производстве рабочих версий системы, которые имеют подмножество общего числа требуемых функций. Итеративная разработка обеспечивает обратную связь как для клиентов, так и для разработчиков.

Agile-методологии сосредоточены на людях, а не на ролях, которые люди выполняют (Fowler, 2003). Роли, которые выполняют люди, системный аналитик, тестировщик или менеджер, не так важны, как люди, которые выполняют эти роли. Фаулер утверждает, что применение инженерных принципов к разработке систем привело к взгляду на людей как на взаимозаменяемые единицы, а не на людей как на талантливых личностей, каждый из которых привносит что-то уникальное в команду разработчиков.

Таблица 1.3 Манифест Agile

**Манифест гибкой разработки программного обеспечения**

*Семнадцать анархистов согласны с этим:*

* Мы открываем лучшие способы разработки программного обеспечения, делая это и помогая другим делать это. Благодаря этой работе мы пришли к оценке:
* *Люди и взаимодействия* над процессами и инструментами.
* *Рабочее программное обеспечение* над исчерпывающей документацией.
* *Сотрудничество* с клиентами вместо переговоров по контракту.
* *Реагирование на изменения,* а не следование плану.

То есть, в то время как мы ценим предметы справа, мы больше ценим предметы слева. Мы следуем следующим принципам:

1. Нашим главным приоритетом является удовлетворение потребностей клиентов за счет своевременной и непрерывной поставки ценного программного обеспечения.
2. Приветствуйте изменение требований, даже на поздних этапах разработки. Гибкие процессы используют изменения для получения конкурентного преимущества клиента.
3. Поставляйте работающее программное обеспечение часто, от пары недель до нескольких месяцев, отдавая предпочтение более коротким срокам.
4. Бизнесмены и разработчики ежедневно работают вместе на протяжении всего проекта.
5. Стройте проекты вокруг мотивированных людей. Обеспечьте им среду и поддержку, в которых они нуждаются, и доверьте им выполнение работы.
6. Наиболее эффективным и действенным методом передачи информации команде разработчиков и внутри нее является личное общение.
7. Работающее программное обеспечение является основным мерилом прогресса.
8. Постоянное внимание к техническому совершенству и хорошему дизайну повышает гибкость.
9. Гибкие процессы способствуют устойчивому развитию. Спонсоры, разработчики и пользователи должны иметь возможность поддерживать постоянный темп на неопределенный срок.
10. Простота — искусство максимизировать объем невыполненной работы — имеет важное значение.
11. Лучшие архитектуры, требования и проекты возникают в результате самоорганизующихся команд.
12. Через равные промежутки времени команда размышляет о том, как стать более эффективной, а затем соответствующим образом настраивает и корректирует свое поведение.

—Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern,

k, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

[*ww.agileAlliance.org*)](http://www.agileAlliance.org/)

Гибкие методологии способствуют самоадаптирующемуся процессу разработки программного обеспечения. По мере разработки программного обеспечения процесс, используемый для его разработки, должен уточняться и совершенствоваться. Команды разработчиков могут сделать это с помощью процесса проверки, часто связанного с завершением итераций. Подразумевается, что по мере адаптации процессов нельзя ожидать найти единую монолитную методологию в рамках данной корпорации или предприятия. Вместо этого можно найти множество вариантов методологии, каждый из которых отражает конкретные таланты и опыт команды, использующей ее.

Гибкие методологии подходят не для каждого проекта. Фаулер (2003) рекомендует гибкий или адаптивный процесс, если ваш проект включает в себя

* + непредсказуемые или динамические требования,
  + ответственные и мотивированные разработчики, а также
  + клиенты, которые понимают процесс и будут участвовать.

Таблица 1.4 Критических факторов, которые отличают гибкий и традиционный подходы к разработке систем

|  |  |  |
| --- | --- | --- |
| Фактор | Гибкие методы | Традиционные методы |
| Размер | Хорошо сочетается с мелкими продуктами и опирается на неявные знания  его масштабируемость. | Методы развивались для работы с большими продуктами и командами. Трудно адаптировать к небольшим проектам. |
| Критичность | Не протестировано на критически важных для безопасности продуктах. Возможные трудности с простой конструкцией и отсутствием документации | Методы эволюционировали для работы с высококритическими продуктами. Трудно адаптировать продукты, которые не являются критичными. |
| Динамизм | Простая конструкция и непрерывный рефакторинг отлично подходят для высокодинамичных сред, но являются источником потенциально дорогостоящей доработки для высокостабильных сред. | Детальные планы и Big Design Up Front, отлично подходят для высокостабильной среды, но являются источником дорогостоящей доработки для высокодинамичных сред |
| Персонал | Требует постоянного присутствия критической массы дефицитных экспертов. Рискованно использовать неподвижных людей. | Нуждается в критической массе дефицитных экспертов во время определения проекта, но может работать с меньшим количеством позже в проекте, если среда не очень динамична. |
| Культура | Процветает в культуре, где люди чувствуют себя комфортно и наделены полномочиями, имея много степеней свободы (процветая в хаосе). | Процветает в культуре, где люди чувствуют себя комфортно и наделены полномочиями, поскольку их роли определяются четкими практиками и процедурами (процветание по порядку). |

Более инженерно-ориентированный, предсказуемый процесс может потребоваться, если команда разработчиков превышает 100 человек или если проект работает по контракту с фиксированной ценой или фиксированным объемом. На самом деле, вопрос о том, организован ли проект разработки систем с точки зрения Agile или более традиционных методологий, зависит от многих различных соображений. Если проект считается высокорискованным и очень сложным и имеет команду разработчиков, состоящую из сотен человек, то будут применяться более традиционные методы. Менее рискованные, меньшие по размеру и более простые усилия по разработке больше подходят для гибких методов. К числу других определяющих факторов относятся организационная практика и стандарты, а также степень, в которой различные части системы будут переданы на подряд другим в целях развития. Очевидно, что чем больше доля системы, которая будет передана на аутсорсинг, тем более подробными должны быть проектные спецификации, чтобы субподрядчики могли понять, что нужно. Несмотря на то, что они не пришли к единому мнению, ключевые различия между этими подходами к развитию перечислены в [таблице 1-4,](#bookmark0) которая основана на работе Бема и Тернера (2004). Эти различия могут быть использованы для определения того, какой подход к разработке лучше всего подходит для конкретного проекта.

Многие различные индивидуальные методологии входят в состав гибких методологий. Фаулер (2003) перечисляет семейство методологий Crystal, адаптивную разработку программного обеспечения, Scrum, разработку на основе функций и другие как гибкие методологии. Однако, возможно, самой известной из этих методологий является экстремальное программирование, обсуждаемое далее.

#### Экстремальное программирование

eXtreme Programming — это подход к разработке программного обеспечения, разработанный Беком и Андресом (2004). Он отличается короткими циклами, инкрементальным подходом к планированию, ориентацией на автоматизированные тесты, написанные программистами и клиентами для мониторинга процесса разработки, и опорой на эволюционный подход к разработке, который длится на протяжении всего срока службы системы. Ключевыми акцентами eXtreme Programming являются использование команд программистов из двух человек, описанных ниже, и наличие клиента на месте в процессе разработки. Соответствующими частями экстремального программирования, относящимися к спецификациям проектирования, являются: (1) то, как планирование, анализ, проектирование и строительство объединяются в одну фазу деятельности; и (2) его уникальный способ сбора и представления системных требований и проектных спецификаций. При экстремальном программировании все фазы жизненного цикла объединяются в ряд действий, основанных на основных процессах кодирования, тестирования, прослушивания и проектирования.

При таком подходе кодирование и тестирование являются тесно связанными частями одного и того же процесса. Программисты, которые пишут код, также разрабатывают тесты. Акцент делается на тестировании тех вещей, которые могут сломаться или пойти не так, а не на тестировании всего. Код тестируется очень скоро после написания. Общая философия, лежащая в основе экстремального программирования, заключается в том, что код будет интегрирован в систему, для которой он разрабатывается, и протестирован в течение нескольких часов после его написания. Если все тесты проходят успешно, то разработка продолжается. В противном случае код переделывается до тех пор, пока тесты не будут успешными.

Еще одна часть экстремального программирования, которая делает процесс кодирования и тестирования более плавным, — это практика парного программирования. Все кодирование и тестирование выполняются двумя людьми, работающими вместе над написанием кода и разработкой тестов. Бек говорит, что парное программирование — это не один человек, печатающий, а другой смотрящий; скорее, два программиста работают вместе над проблемой, которую они пытаются решить, обмениваясь информацией и пониманием и делясь навыками. По сравнению с традиционными методами кодирования, преимущества парного программирования включают в себя: (1) большее (и лучшее) общение между разработчиками, (2) более высокий уровень производительности, (3) более качественный код и (4) усиление других практик в экстремальном программировании, таких как дисциплина кодирования и тестирования (Beck & Andres, 2004). Несмотря на то, что процесс экстремального программирования имеет свои преимущества, как и любой другой подход к разработке систем, он подходит не всем и применим не к каждому проекту.

### Объектно-ориентированный анализ и проектирование

Нет никаких сомнений в том, что **объектно-ориентированный анализ и проектирование (OOAD)** становится все более популярным (мы подробно остановимся на этом подходе позже на протяжении всей книги). OOAD часто называют третьим подходом к разработке систем, после процессно-ориентированного и ориентированного на данные подходов. Объектно-ориентированный подход объединяет данные и процессы (называемые *методами*) в отдельные сущности, называемые **объектами**. Объекты, как правило, соответствуют реальным вещам, с которыми имеет дело информационная система, таким как клиенты, поставщики, контракты и договоры аренды. Объединение данных и процессов в одном месте признает тот факт, что существует ограниченное количество операций для любой заданной структуры данных, и объектно-ориентированный подход имеет смысл, даже если разработка типичных систем сохраняет данные и процессы независимыми друг от друга. Цель OOAD состоит в том, чтобы сделать элементы систем более многоразовыми, тем самым повышая качество системы и производительность системного анализа и проектирования.

Еще одна ключевая идея, лежащая в основе объектной ориентации, — **наследование**. Объекты организуются в классы объектов, которые представляют собой группы объектов, имеющих общие структурные и поведенческие характеристики. Наследование позволяет создавать новые классы, которые разделяют некоторые характеристики существующих классов. Например, из класса объектов, называемого «person», можно использовать наследование для определения другого класса объектов, называемого «клиент». Объекты класса «клиент» будут иметь определенные общие характеристики с объектами класса «человек»: они оба будут иметь имена, адреса, номера телефонов и так далее. Поскольку «человек» является более общим классом, а «клиент» более конкретным, каждый клиент является личностью, но не каждый человек является клиентом.

Как и следовало ожидать, требуется язык компьютерного программирования, который может создавать и манипулировать объектами и классами объектов для создания объектно-ориентированных информационных систем. Было создано несколько объектно-ориентированных языков программирования (например, C++, Eiffel и Java). Фактически, объектно-ориентированные языки были разработаны первыми, а затем последовали методы объектно-ориентированного анализа и проектирования. Поскольку OOAD все еще является относительно новым, существует мало консенсуса или стандартизации среди многих доступных методов OOAD. В целом, основной задачей объектно-ориентированного анализа является идентификация объектов и определение их структуры, поведения и взаимосвязей. Основными задачами объектно-ориентированного проектирования являются моделирование деталей поведения объектов и связи с другими объектами, чтобы системные требования были выполнены, а также пересмотр и переопределение объектов, чтобы лучше использовать преимущества наследования и других преимуществ ориентации на объекты.

Объектно-ориентированный подход к разработке систем разделяет итеративный подход к разработке гибких методологий. Некоторые говорят, что нынешнее внимание к гибкости в разработке систем является не чем иным, как массовым принятием объектно-ориентированных подходов, которые прорастали в течение многих лет, поэтому это сходство не должно вызывать удивления (Fowler, 2003). Одной из наиболее популярных реализаций итеративного подхода к объектно-ориентированной разработке является **Rational Unified Process (RUP**), основанный на итеративном, инкрементальном подходе к разработке систем. RUP состоит из четырех этапов: создание, разработка, построение и переход (см. [рис. 1-11).](#bookmark0)

На начальном этапе аналитики определяют объем, определяют осуществимость проекта, понимают требования пользователей и готовят план разработки программного обеспечения. На этапе разработки аналитики детализируют требования пользователей и разрабатывают базовую архитектуру. Анализ и проектирование составляют основную часть этапа разработки. На этапе создания программное обеспечение фактически кодируется, тестируется и документируется. На переходном этапе система развертывается, а пользователи обучаются и поддерживаются. Как видно из [рисунка 1-11](#bookmark0), этап строительства, как правило, является самым длительным и ресурсоемким. Этап разработки также длительный, но менее ресурсоемкий. Переходный этап является ресурсоемким, но коротким. Начальный этап является коротким и наименее ресурсоемким. Площади прямоугольников на [рисунке 1-11](#bookmark0) дают оценку общих ресурсов, выделенных на каждый этап.

Каждая фаза может быть дополнительно разделена на итерации. Программное обеспечение разрабатывается постепенно в виде серии итераций. Начальная фаза, как правило, влечет за собой одну итерацию. На этом этапе определяется масштаб и осуществимость проекта. Этап разработки может состоять из одной или двух итераций и, как правило, считается наиболее важным из четырех этапов (Kruchten, 2000). Этап разработки в основном связан с системным анализом и проектированием, хотя задействованы и другие виды деятельности. В конце этапа проработки должна была быть разработана архитектура проекта. Архитектура включает в себя видение продукта, исполняемую демонстрацию критически важных частей, подробный глоссарий и предварительное руководство пользователя, подробный план строительства и пересмотренную смету запланированных расходов.

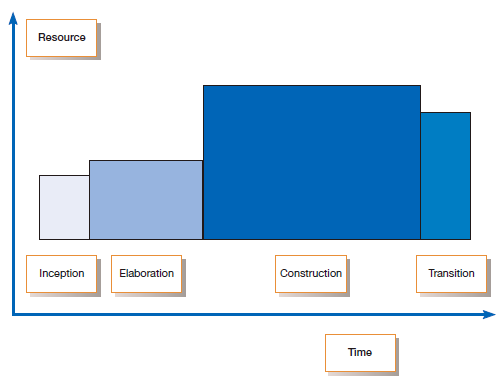


Рисунок 1.11 Фазы разработки, основанной на OOAD

Хотя этап строительства в основном включает в себя кодирование, которое выполняется в нескольких итерациях, пересмотренные требования пользователей могут потребовать анализа и проектирования. Компоненты разрабатываются или приобретаются и используются в коде. По мере завершения работы над каждым исполняемым файлом он тестируется и интегрируется. В конце этапа строительства выпускается бета-версия проекта, которая должна иметь эксплуатационные возможности. Переходный этап включает в себя исправление проблем, бета-тестирование, обучение пользователей и конверсию продукта. Переходный этап завершается, когда цели проекта соответствуют критериям приемлемости. После того, как критерии приемки будут выполнены, продукт может быть выпущен для распространения.

### Наш подход к разработке систем

Большая часть критики SDLC была основана на злоупотреблениях жизненным циклом, как реальным, так и воображаемым. Одна из критических замечаний, основанная на реальности, заключается в том, что опора на подход жизненного цикла вынуждает нематериальные и динамические процессы, такие как анализ и проектирование, переходить во временные фазы, которые были обречены на провал (Martin, 1999). Разработка программного обеспечения не похожа на строительство моста, и одни и те же типы инженерных процессов не всегда могут быть применены (Fowler, 2003), даже несмотря на то, что рассмотрение разработки программного обеспечения как науки, а не искусства, несомненно, привело к значительным улучшениям в процессе и конечных продуктах. Еще одно критическое замечание, основанное на фактах, заключается в том, что зависимость от жизненного цикла привела к огромному количеству процессов и документов, большая часть которых, по-видимому, существует сама по себе. Слишком много процессов и документации замедляют разработку, отсюда и оптимизация, лежащая в основе гибких методологий, и предостережение разработчиков Agile о том, что исходного кода достаточно документации. Критика SDLC, которая больше основана на художественной литературе, заключается в том, что все версии SDLC похожи на водопад, без обратной связи между шагами. Еще одна ложная критика заключается в том, что подход жизненного цикла обязательно ограничивает участие пользователей на самых ранних стадиях процесса. Тем не менее, гибкие методологии и, в частности, экстремальное программирование пропагандируют последовательность анализ-дизайн-код-тестирование, которая представляет собой цикл (рис. 1-8), и пользователи могут быть вовлечены и участвуют в каждом этапе этого цикла; таким образом, циклы сами по себе не обязательно ограничивают участие пользователя.

Независимо от того, была ли критика основана на фактах или нет, это правда, что традиционный каскадный подход SDLC имеет проблемы, и мы приветствуем изменения, происходящие в сообществе разработчиков систем. Эти изменения позволяют устранять проблемы с традиционными подходами, и, без сомнения, результатом является более качественное программное обеспечение, созданное более профессионально и быстрее.

Однако, несмотря на критику подхода к системному анализу и проектированию, основанного на жизненном цикле, взгляд на системный анализ и проектирование, происходящие в цикле, по-прежнему широко распространен и, как мы думаем, также верен. Существует множество типов циклов, от водопадения до цикла анализа-проектирования-кода-тестирования, и все они отражают итеративный характер разработки систем. Каскадный подход, возможно, теряет свою актуальность, но цикл на рисунке 1-8 набирает популярность, и цикл анализ – проектирование – код – тестирование встраивается в более крупный организационный цикл. Хотя мы обычно используем термины «системный анализ» и «*проектирование» и «разработка систем»* взаимозаменяемо, возможно, лучше думать о *системном анализе и проектировании* как о цикле на рисунке 1-8, а о разработке систем как о более крупном цикле на рисунке 1-2. Цикл «анализ-проектирование-код-тестирование» в значительной степени игнорирует организационное планирование, которое предшествует ему, и последующую организационную установку и обслуживание систем, но все они являются важными аспектами более крупных усилий по разработке систем. И для нас лучший, самый ясный способ думать об обоих усилиях — это с точки зрения циклов.

Поэтому в этой книге вы увидите рисунок 1-2 в начале почти каждой главы. В этой книге мы будем использовать наш SDLC в качестве организующего принципа, с действиями и процессами, организованными в соответствии с тем, подпадают ли они под категорию планирования, анализа, проектирования, реализации или обслуживания. В какой-то степени мы будем искусственно разделять действия и процессы, чтобы каждый из них можно было индивидуально изучать и понимать. Как только отдельные компоненты четко поняты, легче увидеть, как они сочетаются с другими компонентами, и в конечном итоге становится легко увидеть целое. Точно так же, как мы можем искусственно разделять деятельность и процессы, мы также можем создавать искусственные границы между фазами SDLC. Наше установление границ никогда не должно интерпретироваться как жесткое разделение. На практике, как мы видели на примере методологий Agile и во введении в OOAD, фазы и части фаз могут быть объединены для скорости, понимания и эффективности. Наша цель состоит в том, чтобы представить части в логической манере, чтобы вы могли понять все части и как собрать их наилучшим образом для целей разработки ваших систем. Тем не менее, общая структура цикла, итерации остается неизменной. Думайте о цикле как об организующем и руководящем принципе.

### Резюме

Эта глава познакомила вас с анализом и проектированием информационных систем, сложным организационным процессом, посредством которого разрабатываются и поддерживаются компьютерные информационные системы. Вы читали о том, как системный анализ и проектирование в организациях изменились за последние несколько десятилетий. Вы также узнали о базовой структуре, которая направляет системный анализ и проектирование, — жизненном цикле разработки систем (SDLC) с пятью основными этапами: планирование, анализ, проектирование, внедрение и обслуживание. Жизненный цикл SDLC подвергся критике, о которой вы читали, и были разработаны другие фреймворки для решения проблем жизненного цикла. Эти подходы включают в себя инструменты автоматизированной разработки программного обеспечения (CASE) и гибкие методологии, наиболее известной из которых является экстремальное программирование. Вы также кратко познакомились с объектно-ориентированным анализом и проектированием, подходом, который становится все более популярным. Все эти подходы разделяют основную идею итерации, что проявляется в жизненном цикле разработки систем и цикле анализа — проектирования — кода — тестирования гибких методологий.

## Глава 2 Истоки программного обеспечения

### Введение

Как вы узнали из главы 1, не так давно было время, когда не существовало системных аналитиков и символьных языков программирования. Тем не менее, люди по-прежнему писали и программировали приложения для компьютеров. В главе 1 вы прочтете об изменениях, произошедших за последние 60 с лишним лет. Несмотря на то, что у современного системного аналитика есть десятки языков программирования и инструментов разработки для работы, вы можете легко утверждать, что разработка систем сейчас еще сложнее, чем 60 лет назад. Тогда, как и совсем недавно, за вас были решены определенные вопросы: если вы хотели написать прикладное программное обеспечение, вы делали это самостоятельно, и вы писали программное обеспечение с нуля. Сегодня существует множество различных источников программного обеспечения, и многие из вас, читающих эту книгу, в конечном итоге будут работать в фирмах, производящих программное обеспечение, а не в отделе информационных систем корпорации. Но для тех из вас, кто продолжает работать в отделе корпоративных информационных систем, основное внимание больше не уделяется исключительно внутренней разработке. Вместо этого основное внимание будет сосредоточено на том, где получить множество частей и компонентов, которые вы объедините в систему приложений, которую вас попросили создать. Вы и ваши коллеги по-прежнему будете писать код, в основном для того, чтобы все различные части работали вместе, но все больше и больше вашего прикладного программного обеспечения будет написано кем-то другим. Несмотря на то, что вы не будете писать код, вы все равно будете использовать базовую структуру и процессы жизненного цикла системного анализа и проектирования для создания прикладных систем, необходимых вашей организации. Организационный процесс разработки систем остается в центре внимания остальной части книги, но сначала вам нужно узнать больше о том, откуда берется программное обеспечение в современной среде разработки.

В этой главе вы узнаете о различных источниках программного обеспечения для организаций. Первым рассматриваемым источником является аутсорсинг, при котором все или часть информационных систем организации, их разработка и обслуживание передаются другой организации. Затем вы прочитаете о шести различных источниках программного обеспечения: (1) фирмы, предоставляющие услуги в области информационных технологий, (2) поставщики пакетного программного обеспечения, (3) поставщики программного обеспечения для корпоративных решений, (4) облачные вычисления, (5) программное обеспечение с открытым исходным кодом и (6) сама организация, когда она разрабатывает программное обеспечение собственными силами. Вы узнаете о критериях оценки программного обеспечения из этих разных источников. Глава завершается обсуждением повторного использования и его влияния на разработку программного обеспечения.

### Приобретение систем

Несмотря на то, что всегда будут некоторые споры о том, когда и где была разработана первая административная информационная система, существует общее согласие в том, что первая такая система в Соединенном Королевстве была разработана в J. Lyons & Sons. В Соединенных Штатах первой административной информационной системой была система расчета заработной платы General Electric (GE), которая была разработана в 1954 году (Computer History Museum, 2003). В то время и в течение многих лет после этого получение информационной системы означало только одно: собственную разработку. Индустрия программного обеспечения появилась только через десять лет после того, как была внедрена система расчета заработной платы GE.

С тех пор, как была построена система расчета заработной платы GE, внутренняя разработка стала пропорционально меньшей частью всей работы по разработке систем, которая происходит в организациях и для них. Внутренние отделы корпоративных информационных систем теперь тратят все меньше и меньше своего времени и усилий на разработку систем с нуля. Компании по-прежнему тратят относительно мало времени и денег на традиционную разработку и обслуживание программного обеспечения. Вместо этого они инвестируют в пакетное программное обеспечение, программное обеспечение с открытым исходным кодом и аутсорсинговые услуги. Сегодня у организаций есть много вариантов при поиске информационной системы. Мы начнем с обсуждения аутсорсинга разработки и эксплуатации, а затем перейдем к презентации об источниках программного обеспечения.

#### Аутсорсинг

Если одна организация разрабатывает или запускает компьютерное приложение для другой организации, такая практика называется **аутсорсингом**. Аутсорсинг включает в себя целый спектр рабочих механизмов. С одной стороны, когда фирма разрабатывает и запускает ваше приложение на своих компьютерах — все, что вы делаете, это вводите и принимаете вывод. Типичным примером такой договоренности является компания, которая запускает приложения для расчета заработной платы для клиентов, чтобы клиентам не приходилось разрабатывать независимую внутреннюю систему расчета заработной платы. Вместо этого они просто предоставляют компании информацию о заработной плате сотрудников, и за определенную плату компания возвращает заполненные зарплаты, отчеты по заработной плате, а также налоговые и другие отчеты для сотрудников. Для многих организаций расчет заработной платы является очень рентабельной операцией, если передать ее на аутсорсинг таким образом. Другим примером аутсорсинга может быть то, что вы наняли компанию для запуска ваших приложений на вашем сайте на ваших компьютерах. В некоторых случаях организация, использующая такую схему, распускает некоторые или все свои подразделения информационных систем (ИС) и увольняет всех своих сотрудников ИБ. Часто компания, привлеченная для управления вычислительной техникой организации, повторно нанимает многих из первоначальных сотрудников подразделения ИБ организации. Аутсорсинг – это большой бизнес. Некоторые организации передают на аутсорсинг разработку информационных технологий (ИТ) для многих своих ИТ-функций, что обходится в миллиарды долларов. Большинство организаций передают на аутсорсинг по крайней мере некоторые аспекты своей деятельности в области информационных систем. Одним из примеров масштабов аутсорсинга является Shell Oil. В 2008 году Shell подписала контракты на аутсорсинг с EDS, T-Systems и AT&T на сумму 3,2 миллиарда долларов США. Кроме того, Shell подписала соглашения о поддержке приложений с IBM, Logica, Wipro и Accenture. В 2011 году Shell передала на аутсорсинг все свои услуги по управлению персоналом и приложениями для расчета заработной платы на основе SAP компании Accenture. Более 90 000 сотрудников Shell в 60 странах мира используют эти системы. Accenture предоставляет эти услуги через аутсорсинговые центры в Индии и на Филиппинах. Отдельные поставщики аутсорсинга, такие как EDS, IBM и Accenture, обычно подписывают крупные контракты на свои услуги. Эти поставщики имеют несколько контрактов на аутсорсинг со многими различными фирмами по всему миру. Зачем организации передавать на аутсорсинг свои операции с информационными системами? Как мы видели на примере расчета заработной платы, аутсорсинг может быть экономически эффективным. Если компания специализируется на расчете заработной платы для других компаний, она может использовать экономию за счет масштаба, которую она достигает от запуска одного стабильного компьютерного приложения для многих организаций, по очень низким ценам. Аутсорсинг также дает фирмам возможность перепрыгнуть через свое нынешнее положение в информационных системах и передать разработку и эксплуатацию внешнему персоналу, обладающему знаниями и навыками, которых нет внутри компании (Ketler and Willems, 1999). Другие причины аутсорсинга включают в себя:

* + высвобождение внутренних ресурсов,
  + повышение доходного потенциала организации,
  + сокращение времени выхода на рынок,
  + повышение эффективности процессов, а также
  + аутсорсинг непрофильных видов деятельности.

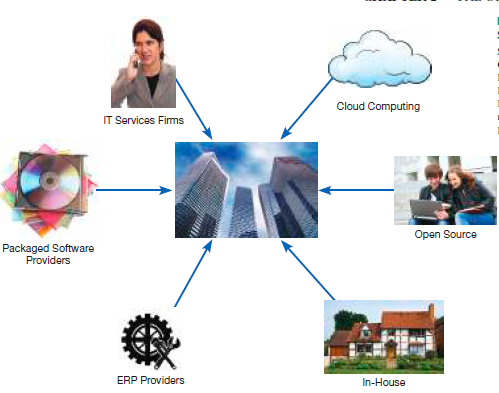
Организация может перейти на аутсорсинг и распустить все свое подразделение обработки информации по политическим причинам, например, для преодоления операционных проблем, с которыми организация сталкивается в своем подразделении информационных систем. Например, город Гранд-Рапидс, штат Мичиган, нанял стороннюю фирму для управления своим вычислительным центром 40 лет назад, чтобы лучше управлять сотрудниками своего вычислительного центра. Профсоюзные контракты и ограничения на государственную службу, действовавшие в то время, затрудняли увольнение людей, поэтому город привлек организацию по управлению объектами для выполнения своих вычислительных операций, и в то же время она смогла избавиться от проблемных сотрудников. Как упоминалось ранее, еще одна причина полного аутсорсинга заключается в том, что руководство организации может чувствовать, что ее основная миссия не связана с управлением подразделением информационных систем и что оно может достичь более эффективных вычислений, передав все свои операции более опытной, ориентированной на компьютер компании. В конце 1980-х годов Kodak решила, что она не занимается компьютерными приложениями, и передала управление своими мэйнфреймами IBM, а управление своими персональными компьютерами - Businessland (Applegate and Montealagre, 1991).

Хотя вы, скорее всего, слышали об аутсорсинге с точки зрения ИТ-рабочих мест со всего мира, отправляющихся в Индию, оказывается, что глобальный рынок аутсорсинга намного сложнее. Согласно отчету ATKearney (2014) за 2014 год, Индия является страной номер один по аутсорсингу, в то время как Китай отстает, а Малайзия занимает третье место. Несмотря на большие потрясения на рынке аутсорсинга в целом на протяжении многих лет, первые три рейтинга не изменились с момента первого отчета ATKearney об аутсорсинге в 2003 году. Не все из 10 лучших аутсорсинговых стран 2014 года расположены в Азии. Хотя шесть из них находятся в Азии, два — в Латинской Америке (Мексика и Бразилия), один — в Европе (Болгария) и один — в Африке (Египет). Даже Соединенные Штаты являются страной аутсорсинга, занимающей 14-е место в списке ATKearney. Фактически, индийские аутсорсинговые фирмы, такие как Wipro, Infosys и Tata Consulting, имеют офисы в Соединенных Штатах. По мере того, как индийские фирмы стали настолько успешными в офшоринге, а валюты колебались, для фирм стало дороже заключать контракты с индийскими компаниями, поэтому многие фирмы начали искать в другом месте. Многие американские фирмы обратились к так называемому ниаршорингу, или заключению контрактов с компаниями в странах Латинской Америки. Многие из этих стран находятся не более чем в одном часовом поясе, и они сохраняют некоторые преимущества стоимости рабочей силы, которые разрушаются в Индии (King, 2008a). Мексика все чаще рассматривается как дополнение к Индии и другим оффшорным локациям и занимает четвертое место в списке ATKearney 2014. Кроме того, фирмы все чаще распределяют свою аутсорсинговую работу между поставщиками в нескольких странах одновременно.

Аналитики должны знать об аутсорсинге как об альтернативе. При разработке альтернативных стратегий развития системы для системы, как аналитик, вы должны консультироваться с организациями в вашем регионе, которые предоставляют услуги аутсорсинга. Вполне возможно, что по крайней мере одна такая организация уже разработала и запускает приложение, очень близкое к тому, о чем просят ваши пользователи. Возможно, аутсорсинг системы замены должен быть одной из ваших альтернатив. Знание ваших системных требований, прежде чем рассматривать аутсорсинг, означает, что вы можете тщательно оценить, насколько хорошо поставщики аутсорсинговых услуг могут удовлетворить ваши потребности. Однако, если вы решите не рассматривать аутсорсинг, вам необходимо определить, следует ли приобретать некоторые программные компоненты вашей системы замены, а не создавать их собственными силами.

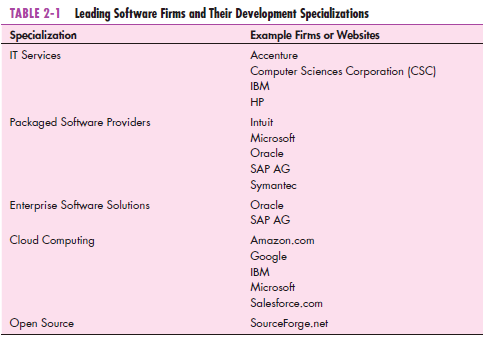
#### Источники программ

Мы можем сгруппировать источники программного обеспечения в шесть основных категорий: фирмы, предоставляющие услуги в области информационных технологий, производители пакетного программного обеспечения, корпоративные решения, поставщики облачных вычислений, программное обеспечение с открытым исходным кодом и собственные разработчики (рис. 2-1). Эти различные источники представляют собой точки в континууме вариантов, с множеством гибридных комбинаций на этом пути.



**Figure 2-1** Sources of application software

Если компания нуждается в **информационной системе**, но не имеет опыта или персонала для разработки системы собственными силами, а подходящая готовая система недоступна, компания, скорее всего, проконсультируется с фирмой, предоставляющей услуги в области информационных технологий. Фирмы, предоставляющие ИТ-услуги, помогают компаниям разрабатывать пользовательские информационные системы для внутреннего использования, или они разрабатывают, размещают и запускают приложения для клиентов, или предоставляют другие услуги. Обратите внимание на таблицу 2-1, что многие ведущие компании-разработчики программного обеспечения в мире специализируются на услугах, которые включают разработку пользовательских систем. В этих фирмах работают люди, обладающие опытом разработки информационных систем. Их консультанты также могут иметь опыт в данной области бизнеса. Например, консультанты, которые работают с банками, разбираются в финансовых учреждениях, а также в информационных системах. Консультанты используют многие из тех же методологий, методов и инструментов, которые компании используют для разработки систем собственными силами.



Вы можете удивиться, увидев, что IBM указана в списке ведущих мировых производителей программного обеспечения; Некоторые люди до сих пор думают об этом в первую очередь как об аппаратной компании. Тем не менее, IBM уже много лет отходит от зависимости от разработки аппаратного обеспечения. Приобретение в 2002 году компанией IBM подразделения ИТ-консалтинга PricewaterhouseCoopers укрепило ее переход к сфере услуг и консалтинга. IBM также хорошо известна своей разработкой программного обеспечения для веб-серверов и промежуточного программного обеспечения. Другие ведущие фирмы, предоставляющие ИТ-услуги, включают традиционные консалтинговые фирмы, такие как Computer Sciences Corp., Accenture и HP (Hewlett-Packard). HP, еще одна компания, ранее занимавшаяся аппаратным обеспечением, перешла на фирму, предоставляющую ИТ-услуги. В 2008 году HP купила EDS, продолжив переход к сервисно-ориентированной компании.

**Производители пакетного программного обеспечения** Рост индустрии программного обеспечения был феноменальным с момента ее основания в середине 1960-х годов. Некоторые из крупнейших компьютерных компаний в мире — это компании, которые производят исключительно программное обеспечение. Хорошим примером является Microsoft, вероятно, самая известная компания-разработчик программного обеспечения в мире. Почти 87 процентов доходов Microsoft приходится на продажи программного обеспечения, в основном для операционных систем Windows и программного обеспечения для повышения производительности Microsoft Office Suite. Также перечисленная в таблице 2-1, Oracle является исключительно компанией-разработчиком программного обеспечения, известной в первую очередь своим программным обеспечением для баз данных, но Oracle также производит корпоративные системы. Еще одна компания в списке, SAP, также является компанией, ориентированной на программное обеспечение, которая разрабатывает системные решения для всего предприятия. Подробнее об Oracle и SAP вы узнаете в ближайшее время, в разделе о корпоративных системах.

Компании-разработчики программного обеспечения разрабатывают то, что иногда называют *предварительно упакованными* или *готовыми системами*. Популярными примерами такого программного обеспечения являются Microsoft Word (рис. 2-2) и Quicken, QuickPay и QuickBooks от Intuit. Индустрия разработки пакетного программного обеспечения обслуживает многие сегменты рынка. Их программные предложения варьируются от общих широких пакетов, таких как инструменты повышения производительности, до очень узких нишевых пакетов, таких как программное обеспечение для управления детским садом. Компании-разработчики программного обеспечения разрабатывают программное обеспечение для работы на различных компьютерных платформах, от микрокомпьютеров до больших мэйнфреймов. Компании варьируются по размеру от нескольких человек до тысяч сотрудников.



**Рисунок 2-2**

Документ, созданный в Microsoft Word

(*Источник:* корпорация Майкрософт.)

Компании-разработчики программного обеспечения консультируются с пользователями системы после завершения первоначального проектирования программного обеспечения и создания ранней версии системы. Затем системы тестируются в реальных организациях, чтобы определить, есть ли какие-либо проблемы или можно ли внести какие-либо улучшения. До тех пор, пока тестирование не будет завершено, система не предлагается к продаже населению.

Некоторые готовые программные системы не могут быть модифицированы для удовлетворения конкретных, индивидуальных потребностей конкретной организации. Такие прикладные системы иногда называют *системами «под ключ*». Производитель системы «под ключ» внесет изменения в программное обеспечение только тогда, когда значительное количество пользователей попросит внести конкретное изменение. Тем не менее, другое готовое прикладное программное обеспечение может быть изменено или расширено производителем или пользователем, чтобы оно более точно соответствовало потребностям организации. Несмотря на то, что многие организации выполняют схожие функции, нет двух организаций, которые делают одно и то же совершенно одинаково. Система «под ключ» может быть достаточно хороша для определенного уровня производительности, но она никогда не будет идеально соответствовать тому, как данная организация ведет бизнес. Разумная оценка заключается в том, что готовое программное обеспечение может в лучшем случае удовлетворить 70 процентов потребностей организации. Таким образом, даже в лучшем случае 30 процентов программного комплекса не соответствует спецификациям организации.

Как упоминалось в главе 1, многие фирмы выбрали комплексные программные решения, называемые *корпоративными решениями* или **системами планирования ресурсов предприятия (ERP),** для поддержки своих операций и бизнес-процессов. Эти программные решения ERP состоят из ряда интегрированных модулей. Каждый модуль поддерживает индивидуальную традиционную бизнес-функцию, такую как бухгалтерский учет, распределение, производство или человеческие ресурсы. Разница между модулями и традиционными подходами заключается в том, что модули интегрированы, чтобы сосредоточиться на бизнес-процессах, а не на функциональных областях бизнеса. Например, ряд модулей будет поддерживать весь процесс ввода заказа, от получения заказа до корректировки запасов, доставки, выставления счетов и послепродажного обслуживания. Традиционный подход будет использовать различные системы в разных функциональных областях бизнеса, такие как биллинговая система в бухгалтерском учете и система инвентаризации на складе. Используя корпоративные программные решения, фирма может интегрировать все части бизнес-процесса в единую информационную систему. Все аспекты одной транзакции происходят в рамках единой информационной системы, а не в виде ряда разрозненных, отдельных систем, ориентированных на функциональные области бизнеса.

Преимущества подхода корпоративных решений включают единое хранилище данных для всех аспектов бизнес-процесса и гибкость модулей. Единое хранилище обеспечивает более согласованные и точные данные, а также меньшую потребность в обслуживании. Модули являются гибкими, поскольку дополнительные модули могут быть добавлены по мере необходимости после того, как базовая система будет установлена. Добавленные модули сразу же интегрируются в существующую систему. Однако у программного обеспечения для корпоративных решений есть недостатки. Системы очень сложные, поэтому внедрение может занять много времени. Организации, как правило, не имеют необходимого опыта для внедрения систем, поэтому им приходится полагаться на консультантов или сотрудников поставщика программного обеспечения, что может быть очень дорого. В некоторых случаях организациям необходимо изменить методы ведения бизнеса, чтобы извлечь выгоду из перехода на корпоративные решения.



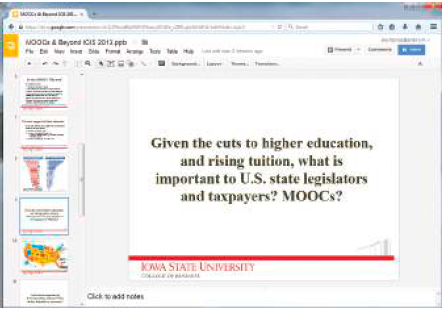
**Figure 2-3**

SAP Business ByDesign, продукт, предназначенный для компаний среднего размера

Несколько крупных поставщиков предоставляют программное обеспечение для корпоративных решений. Наиболее известным, вероятно, является SAP AG, немецкая фирма, упомянутая ранее, известная своим флагманским продуктом R/3. SAP расшифровывается как «Системы, приложения и продукты в обработке данных». SAP AG была основана в 1972 году, но большая часть ее роста произошла с 1992 года. С 2010 года SAP является одним из крупнейших поставщиков программного обеспечения в мире. Другим крупным поставщиком корпоративных решений является Oracle Corp., американская фирма, возможно, более известная своим программным обеспечением для баз данных. Oracle захватила большую долю рынка ERP за счет собственных финансовых систем и приобретения других поставщиков ERP. В конце 2004 года Oracle приобрела PeopleSoft, Inc., американскую фирму, основанную в 1987 году. PeopleSoft начинала с корпоративных решений, ориентированных на управление человеческими ресурсами, и расширилась, чтобы охватить финансы, управление материалами, дистрибуцию и производство до того, как Oracle приобрела их. Незадолго до того, как компания была куплена Oracle, PeopleSoft укрепила свою корпоративную мощь в 2003 году, приобретя другого поставщика ERP, JD Edwards. Вместе SAP и Oracle контролируют около 36 процентов рынка ERP (Compare Business Products, 2014). Поскольку более высокий сегмент рынка стал насыщен ERP-системами, большинство поставщиков ERP смотрят на средний и малый бизнес для роста. Например, предложение SAP для среднего и малого бизнеса называется SAP Business ByDesign (рис. 2-3).

**Облачные вычисления** Другой способ получения приложений для организаций — арендовать их или лицензировать у сторонних поставщиков, которые запускают приложения на удаленных сайтах. Пользователи имеют доступ к приложениям через Интернет или через виртуальные частные сети. Поставщик приложений покупает, устанавливает, обслуживает и обновляет приложения. Пользователи платят по факту использования или лицензируют программное обеспечение, как правило, ежемесячно. Хотя эта практика была известна под разными названиями на протяжении многих лет, сегодня она называется **облачными вычислениями**. Облачные вычисления относятся к предоставлению приложений через Интернет, где клиентам не нужно вкладывать средства в аппаратные и программные ресурсы, необходимые для запуска и обслуживания приложений. Возможно, вы видели, как Интернет называют облаком в других контекстах, что происходит из-за того, как Интернет изображен на диаграммах компьютерных сетей. Хорошо известным примером облачных вычислений является Google Apps, где пользователи могут обмениваться документами, электронными таблицами и презентациями и создавать их (рис. 2-4). Другим известным примером является Salesforce.com, которая предоставляет программное обеспечение для управления взаимоотношениями с клиентами в Интернете. Облачные вычисления охватывают многие области технологий, включая программное обеспечение как услугу (часто называемое *SaaS*), которое включает в себя Salesforce.com, и оборудование как услугу, которое включает Amazon Web Services и позволяет компаниям заказывать серверные мощности и хранилища по требованию.

Microsoft и IDC прогнозировали, что к 2015 году облачные вычисления создадут 14 миллионов новых рабочих мест, а общий мировой рынок облачных вычислений достигнет 1,1 триллиона долларов США в этом году (McDougall, 2012). Компании, которые, скорее всего, получат немедленную прибыль, — это те, которые могут быстро настроить свои продуктовые линейки в соответствии с потребностями облачных вычислений. К ним относятся такие известные имена, как IBM, которая построила несколько центров облачных вычислений по всему миру; Microsoft, которая в 2008 году объявила о своей платформе Azure для поддержки разработки и эксплуатации бизнес-приложений и потребительских сервисов на своих собственных серверах; и Amazon.com, которая предоставляет клиентам хранилище и емкость со своих собственных серверов.



**Рисунок 2-4**

Презентация, отредактированная в Google Apps Перепечатано с разрешения Джоуи Ф. Джорджа.

Как показывают эти прогнозы роста, выбор пути облачных вычислений имеет свои преимущества. Три основные причины выбора облачных вычислений, каждая из которых приводит к преимуществам для компании: (1) высвобождение внутреннего ИТ-персонала, (2) получение доступа к приложениям быстрее, чем при внутренней разработке, и (3) достижение более дешевого доступа к приложениям корпоративного качества. Особенно привлекательной является возможность получить доступ к большим и сложным системам без необходимости проходить дорогостоящий и трудоемкий процесс внедрения самих систем собственными силами. Получение ваших вычислений через облако также облегчает уход от неудовлетворительного системного решения. Другие причины включают экономическую эффективность, скорость выхода на рынок и лучшую производительность (Moyle & Kelley, 2011).

Однако у ИТ-менеджеров есть некоторые опасения по поводу облачных вычислений. Основная проблема связана с безопасностью. Опасения по поводу безопасности основаны на хранении данных компании на машинах, которыми никто не владеет и к которым другие могут получить доступ. Фактически, двумя основными причинами отказа от использования облачных сервисов являются опасения по поводу несанкционированного доступа к конфиденциальной информации и несанкционированного доступа к информации о клиентах (Moyle & Kelley, 2011). Еще одна проблема - надежность. Некоторые предупреждают, что облако на самом деле представляет собой сеть сетей, и поэтому оно уязвимо для неожиданных рисков из-за своей сложности (kfc, 2012). Еще одной проблемой является соблюдение правительственных постановлений, таких как Закон Сарбейнса-Оксли. Эксперты рекомендуют трехэтапный процесс безопасной миграции в облако (Moyle & Kelley, 2011). Во-первых, привлеките экспертов компании по безопасности на ранних этапах процесса миграции, чтобы можно было выбрать поставщика, который понимает требования компании к безопасности и нормативным требованиям. Во-вторых, установите реалистичные требования к безопасности. Убедитесь, что требования четко прописаны в процессе торгов. В-третьих, проведите честную оценку рисков. Определите, какие данные будут перенесены, и обратите внимание на то, как они будут управляться поставщиком облачных услуг. После того, как миграция произошла, важно, чтобы компании продолжали контролировать свои данные и системы и активно работали с поставщиком для обеспечения безопасности.

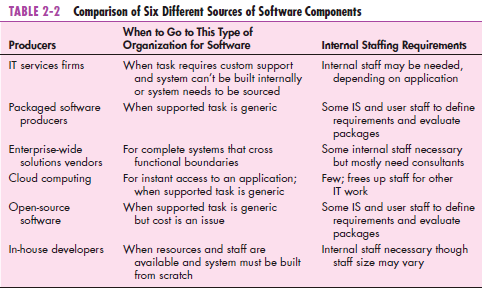
**Программное обеспечение с открытым исходным кодом** Программное обеспечение с открытым исходным кодом отличается от других типов программного обеспечения, о которых вы читали до сих пор. Программное обеспечение с открытым исходным кодом отличается тем, что оно находится в свободном доступе, не только конечный продукт, но и сам исходный код. Он отличается еще и тем, что разрабатывается сообществом заинтересованных людей, а не сотрудниками конкретной компании. Программное обеспечение с открытым исходным кодом выполняет те же функции, что и коммерческое программное обеспечение, такое как операционные системы, электронная почта, системы баз данных, веб-браузеры и т. д. Некоторые из самых известных и популярных названий программного обеспечения с открытым исходным кодом: Linux, операционная система; mySQL, система баз данных; и Firefox, веб-браузер. Открытый исходный код также применим к программным компонентам и объектам. Открытый исходный код разрабатывается и поддерживается сообществами людей, и иногда эти сообщества могут быть очень большими. Разработчики часто используют общие веб-ресурсы, такие как SourceForge.net, для организации своей деятельности. По состоянию на январь 2105 года SourceForge.net размещал 430 000 проектов и имел более 3,7 миллиона зарегистрированных пользователей. Нет никаких сомнений в том, что движение за открытый исходный код не имело бы того успеха, которым оно пользуется, без наличия Интернета для обеспечения доступа и организации деятельности в области развития. Если программное обеспечение бесплатное, вы можете задаться вопросом, как кто-то зарабатывает деньги, разрабатывая программное обеспечение с открытым исходным кодом. Компании и частные лица могут зарабатывать деньги с открытым исходным кодом двумя основными способами: (1) предоставляя техническое обслуживание и другие услуги или (2) предоставляя одну версию программного обеспечения бесплатно и продавая более полнофункциональную версию. Некоторые решения с открытым исходным кодом оказывают большее влияние на индустрию программного обеспечения, чем другие. Linux, например, был очень успешен на рынке серверного программного обеспечения, где, по оценкам, он занимает до 36 процентов доли рынка (W3Techs, 2015). На рынке настольных операционных систем Linux занимает около 1% рынка. Другие программные продукты с открытым исходным кодом, такие как mySQL, также были успешными, и доля открытого исходного кода в индустрии программного обеспечения, похоже, обречена на дальнейший рост.

**Собственная разработка** Мы говорили о нескольких различных типах внешних организаций, которые служат источниками программного обеспечения, но внутренняя разработка остается вариантом. Внутренняя разработка становится все меньшей частью всей работы по разработке систем, которая происходит в организациях и для них. Как вы читали ранее в этой главе, внутренние отделы корпоративных информационных систем теперь тратят все меньше и меньше своего времени и усилий на разработку систем с нуля. Внутренняя разработка может привести к большей нагрузке на обслуживание, чем другие методы разработки, такие как пакетные приложения. Исследование, проведенное Бэнкером, Дэвисом и Слотером, показало, что использование генератора кода в качестве основы для внутренней разработки было связано с увеличением времени обслуживания, тогда как использование пакетных приложений было связано с уменьшением усилий по обслуживанию. Конечно, внутренняя разработка не обязательно должна повлечь за собой разработку всего программного обеспечения, которое будет составлять всю систему. Распространены гибридные решения, включающие в себя некоторые приобретенные и некоторые собственные программные компоненты. Если вы решите приобрести программное обеспечение из внешних источников, этот выбор будет сделан в конце этапа анализа. Выбор между пакетом и внешним поставщиком будет определяться вашими потребностями, а не тем, что поставщик должен продать. Как мы обсудим, результаты вашего аналитического исследования определят тип продукта, который вы хотите купить, и сделают работу с внешним поставщиком намного проще, продуктивнее и выгоднее. В таблице 2-2 сравниваются шесть различных источников программного обеспечения, обсуждаемых в этом разделе.

#### Выбор готового программного обеспечения

После того, как вы решили приобрести готовое программное обеспечение, а не писать часть или все программное обеспечение для своей новой системы, как вы решаете, что покупать? Есть несколько критериев, которые следует учитывать, и особые критерии могут возникать при каждой потенциальной покупке программного обеспечения. Для каждого критерия следует провести четкое сравнение между пакетом программного обеспечения и процессом разработки того же приложения собственными силами. К наиболее распространенным критериям относятся следующие:

* Стоимость
* Функциональность
* Поддержка вендоров
* Жизнеспособность вендора
* Гибкость
* Документация
* Время ответа
* Простота установки



Эти критерии представлены в произвольном порядке. Относительная важность критериев будет варьироваться от проекта к проекту и от организации к организации. Если бы вам пришлось выбрать два критерия, которые всегда были бы одними из самых важных, этими двумя, вероятно, были бы жизнеспособность поставщика и поддержка поставщика. Вы не хотите связываться с поставщиком, который, возможно, не будет работать завтра. Точно так же вы не хотите лицензировать программное обеспечение у поставщика с репутацией плохой поддержки. То, как вы ранжируете важность остальных критериев, будет во многом зависеть от конкретной ситуации, в которой вы окажетесь.  *Стоимость* включает в себя сравнение стоимости разработки той же системы собственными силами с затратами на покупку или лицензирование пакета программного обеспечения. Вы должны включить сравнение затрат на приобретение обновлений поставщика или ежегодных лицензионных сборов с затратами, которые вы понесете для обслуживания собственного программного обеспечения. Затраты на закупку и разработку собственных средств могут быть сопоставлены на основе показателей экономической целесообразности (например, приведенная стоимость может быть рассчитана для денежного потока, связанного с каждой альтернативой).  *Функциональность* относится к задачам, которые может выполнять программное обеспечение, а также к обязательным, важным и желаемым функциям системы. Может ли программный пакет выполнять все или только некоторые задачи, необходимые вашим пользователям? Если только некоторые, может ли он выполнять необходимые основные задачи? Обратите внимание, что выполнение требований пользователей происходит в конце этапа анализа, так как вы не можете оценить упакованное программное обеспечение до тех пор, пока требования пользователей не будут собраны и структурированы. Приобретение прикладного программного обеспечения не является заменой проведения этапа системного анализа; скорее, приобретение программного обеспечения является частью одной стратегии проектирования для приобретения системы, выявленной в ходе анализа.

*Стоимость* включает в себя сравнение стоимости разработки той же системы собственными силами с затратами на покупку или лицензирование пакета программного обеспечения. Вы должны включить сравнение затрат на приобретение обновлений поставщика или ежегодных лицензионных сборов с затратами, которые вы понесете для обслуживания собственного программного обеспечения. Затраты на закупку и разработку собственных средств могут быть сопоставлены на основе показателей экономической целесообразности (например, приведенная стоимость может быть рассчитана для денежного потока, связанного с каждой альтернативой).

*Функциональность* относится к задачам, которые может выполнять программное обеспечение, а также к обязательным, важным и желаемым функциям системы. Может ли программный пакет выполнять все или только некоторые задачи, необходимые вашим пользователям? Если только некоторые, может ли он выполнять необходимые основные задачи? Обратите внимание, что выполнение требований пользователей происходит в конце этапа анализа, так как вы не можете оценить упакованное программное обеспечение до тех пор, пока требования пользователей не будут собраны и структурированы. Приобретение прикладного программного обеспечения не является заменой проведения этапа системного анализа; Скорее, приобретение программного обеспечения является частью одной стратегии проектирования для приобретения системы, выявленной в ходе анализа.

Как мы уже говорили ранее, *поддержка поставщика* относится к тому, может ли поставщик предоставлять поддержку и в каком объеме он может предоставить. Поддержка осуществляется в виде помощи в установке программного обеспечения, обучения пользователей и системного персонала работе с программным обеспечением, а также оказания помощи при возникновении проблем после установки. В последнее время многие компании-разработчики программного обеспечения значительно сократили объем бесплатной поддержки, которую они будут предоставлять клиентам, поэтому следует учитывать стоимость использования средств поддержки по телефону, на месте, факсу или компьютерной доске объявлений. С поддержкой связана жизнеспособность поставщика. Вы не хотите застревать с программным обеспечением, разработанным поставщиком, который может скоро выйти из бизнеса. Этот последний момент не следует преуменьшать. Индустрия программного обеспечения довольно динамична, и инновационное прикладное программное обеспечение создается предпринимателями, работающими из домашних офисов — классическая кустарная индустрия. Такие организации, даже с выдающимся программным обеспечением, часто не имеют ресурсов или способности управлять бизнесом, чтобы оставаться в бизнесе очень долго. Кроме того, конкурентные действия крупных фирм-разработчиков программного обеспечения могут привести к тому, что продукты более мелких фирм станут устаревшими или несовместимыми с операционными системами. Одна фирма-разработчик программного обеспечения, с которой мы разговаривали во время работы над этой книгой, изо всех сил пыталась выжить, просто пытаясь заставить свое программное обеспечение работать на любом предположительно ПК с Windows (учитывая бесконечную комбинацию видеокарт, мониторов, чипов BIOS и других компонентов). Идти в ногу с изменениями аппаратного и системного программного обеспечения может быть больше, чем может справиться небольшая фирма, и хорошее готовое прикладное программное обеспечение может быть потеряно.

*Гибкость* относится к тому, насколько легко вам или поставщику настроить программное обеспечение. Если программное обеспечение не очень гибкое, пользователям, возможно, придется адаптировать способ работы, чтобы он соответствовал программному обеспечению. Могут ли они адаптироваться таким образом? Приобретенное программное обеспечение может быть изменено несколькими способами. Иногда поставщик будет готов внести для вас индивидуальные изменения, если вы готовы заплатить за редизайн и программирование. Некоторые поставщики разрабатывают программное обеспечение для настройки. Например, программное обеспечение может включать в себя несколько различных способов обработки данных, и во время установки заказчик выбирает, какой из них инициировать. Кроме того, дисплеи и отчеты могут быть легко переработаны, если эти модули написаны на языке четвертого поколения. Отчеты, формы и дисплеи можно легко настроить с помощью процесса, при котором название вашей компании и выбранные заголовки для отчетов, дисплеев, форм, заголовков столбцов и т. д. выбираются из предоставленной вами таблицы параметров. Возможно, вы захотите использовать некоторые из этих же методов настройки для систем, разработанных собственными силами, чтобы программное обеспечение можно было легко адаптировать для различных бизнес-подразделений, линеек продуктов или отделов.

*Документация* включает в себя руководство пользователя, а также техническую документацию. Насколько понятна и актуальна документация? Какова стоимость нескольких копий, если требуется? *Время отклика* относится к тому, сколько времени требуется программному пакету для ответа на запросы пользователя в интерактивном сеансе. Другой мерой времени может быть то, сколько времени требуется программному обеспечению для завершения выполнения задания. Наконец, простота установки является мерой сложности загрузки программного обеспечения и обеспечения его работоспособности.

Конечно, критерии приобретения программного обеспечения будут варьироваться в зависимости от типа системы, которую вы приобретаете. Например, если вы думаете о лицензировании ERP-системы, вы, безусловно, примете во внимание все предыдущие критерии, но вы также захотите изучить критерии, характерные для ERP-систем. Вервиль и его коллеги (2005) изучили организации, которые приобрели ERP-системы, чтобы выяснить, какие критические факторы были для успеха. Они обнаружили 10 факторов успеха, 5 из которых связаны с процессом приобретения, а 5 - с людьми в процессе. Они обнаружили, что процесс приобретения должен быть тщательно спланирован и структурирован, и он должен быть строгим. Чтобы процесс был успешным, во время планирования нельзя упускать из виду ничего. Важно, чтобы два из пяти факторов успеха, связанных с этим процессом, были завершены до того, как были установлены контакты с поставщиками ПОР. Эти два фактора определяли все системные требования и устанавливали критерии отбора и оценки. Эти два фактора помогли организациям составить четкое описание своих потребностей и оценить предложения от поставщиков. Пятым критерием, связанным с процессом, является получение точной информации. Источники информации нуждаются в проверке и перекрестной проверке.

Остальные пять факторов успеха касались людей, вовлеченных в процесс приобретения. Первым фактором был четкий и недвусмысленный авторитет. Человек, отвечающий за процесс, должен быть объективным и сильным лидером. Во-вторых, важен был состав команды по приобретению. Команда должна была быть разнообразной, чтобы каждый член обладал определенным набором навыков, который дополнял навыки других членов команды. В-третьих, было сочтено важным подходить к отношениям с поставщиком как к партнерству, а не как к состязательным или нейтральным отношениям. Учитывая сложность и стоимость систем ПОР, члены организации-покупателя будут работать с поставщиками в течение нескольких лет, поэтому необходимы комфортные рабочие отношения. В-четвертых, будущие пользователи системы ПОР были активными участниками процесса приобретения. Наконец, пятым фактором успеха, связанным с людьми в этом процессе, была поддержка пользователей. В исследованных компаниях поддержка пользователей часто выражалась в принятии системой и даже в энтузиазме и волнении по этому поводу.

#### Проверка сведений о приобретенном программном обеспечении

Один из способов получить всю необходимую информацию о программном пакете — собрать ее у поставщика. Часть этой информации может содержаться в документации по программному обеспечению и технической маркетинговой литературе. Другая информация может быть предоставлена по запросу. Например, вы можете отправить потенциальным поставщикам анкету, задавая конкретные вопросы об их пакетах. Это может быть частью процесса запроса **предложений (RFP)** или запроса котировок (RFQ), который требуется вашей организации при совершении крупных закупок. Место не позволяет нам обсуждать здесь тему RFP и RFQ; Вы можете обратиться к текстам по закупкам и маркетингу, если вы не знакомы с такими процессами (дополнительные ссылки на RFP и RFQ можно найти в конце этой главы).

Конечно, ничто не заменит фактическое использование программного обеспечения самостоятельно и прохождение его через серию тестов на основе критериев выбора программного обеспечения. Не забудьте протестировать не только программное обеспечение, но и документацию, учебные материалы и даже средства технической поддержки. Одно из требований, которое вы можете предъявить потенциальным поставщикам программного обеспечения в рамках процесса торгов, заключается в том, что они должны установить (бесплатно или по согласованной цене) свое программное обеспечение в течение ограниченного периода времени на ваши компьютеры. Таким образом, вы можете определить, как их программное обеспечение работает в вашей среде, а не в какой-то оптимизированной среде, которую они используют для демонстрационных целей.

Одним из самых надежных и проницательных источников являются другие пользователи программного обеспечения. Продавцы обычно предоставляют список клиентов (помните, что они, естественно, расскажут вам об довольных клиентах, поэтому вам, возможно, придется исследовать срез клиентов) и людей, которые хотят, чтобы с ними связались потенциальные клиенты. И здесь ваша личная сеть контактов, развитая через профессиональные группы, друзей по колледжу, торговые ассоциации или местные бизнес-клубы, может быть ресурсом; не стесняйтесь найти контакты самостоятельно. Такие нынешние или бывшие клиенты могут предоставить глубокое представление об использовании пакета в своих организациях.

Чтобы получить ряд мнений о возможных пакетах, вы можете использовать независимые службы тестирования и абстрагирования программного обеспечения, которые периодически оценивают программное обеспечение и сопоставляют мнения пользователей. Такие опросы доступны за плату либо в виде услуг по подписке, либо по запросу (двумя популярными сервисами являются Auerbach Publishers и DataPro); время от времени в отраслевых изданиях появляются объективные обзоры. Однако часто статьи в отраслевых изданиях, даже обзоры программного обеспечения, на самом деле засеяны производителем программного обеспечения и не являются беспристрастными.

Если вы сравниваете несколько пакетов программного обеспечения, вы можете присвоить баллы для каждого пакета по каждому критерию и сравнить баллы, используя количественный метод, который мы демонстрируем в главе 4 для сравнения альтернативных стратегий проектирования систем.

### Переиспользование

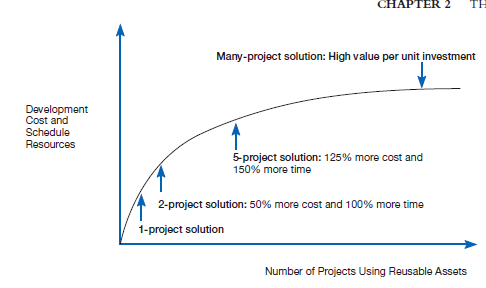
**Повторное использование** — это использование ранее написанных программных ресурсов в новых приложениях. Поскольку так много битов и фрагментов приложений являются относительно общими для всех приложений, кажется интуитивно понятным, что во многих областях может быть достигнута большая экономия, если эти общие биты и фрагменты не нужно писать заново каждый раз, когда они необходимы. Повторное использование должно повысить производительность программистов, поскольку возможность использовать существующее программное обеспечение для некоторых функций означает, что они могут выполнять больше работы за то же время. Повторное использование также должно сократить время разработки, сводя к минимуму превышение расписания. Поскольку существующие части программного обеспечения уже были протестированы, их повторное использование также должно привести к созданию более качественного программного обеспечения с более низким уровнем дефектов, что снижает затраты на обслуживание.

Хотя повторное использование может применяться ко многим различным аспектам программного обеспечения, обычно оно чаще всего применяется к двум различным технологиям разработки: объектно-ориентированной и компонентно-ориентированной разработке. Вы вкратце познакомились с объектно-ориентированной разработкой в главе 1. Например, рассмотрим класс объектов, созданный для моделирования сотрудника. Класс объекта Employee будет содержать как данные о сотрудниках, так и инструкции, необходимые для расчета заработной платы для различных типов должностей. Класс объектов можно использовать в любом приложении, имеющем дело с сотрудниками, но если необходимо внести изменения в расчет заработной платы для различных типов сотрудников, изменения должны быть внесены только в класс объектов, а не в различные приложения, которые его используют. По определению, использование класса объекта Employee в нескольких приложениях представляет собой повторное использование.

Компонентная разработка похожа на объектно-ориентированную разработку в том, что основное внимание уделяется созданию частей программного обеспечения общего назначения, которые могут использоваться взаимозаменяемо во многих различных программах. Компоненты могут быть как маленькими, как объекты, так и большими, как части программного обеспечения, которые обрабатывают отдельные бизнес-функции, такие как конвертация валюты. Идея, лежащая в основе разработки на основе компонентов, заключается в сборке приложения из множества различных компонентов на разных уровнях сложности и размера. Многие поставщики работают над разработкой библиотек компонентов, которые могут быть извлечены и собраны по мере необходимости в желаемые приложения.

Некоторые данные свидетельствуют о том, что повторное использование может быть эффективным, особенно для классов объектов. Например, одно лабораторное исследование показало, что повторное использование библиотек классов объектов приводит к повышению производительности, снижению плотности дефектов и уменьшению количества доработок (Basili et al., 1996). Для HP программа повторного использования привела к сокращению времени вывода на рынок определенных продуктов в три и более раза, с 18 месяцев до менее чем 5 месяцев (Griss, 2003). Однако для того, чтобы повторное использование работало в организационной среде, необходимо решить множество различных вопросов. Технические проблемы включают в себя отсутствие в настоящее время методологии для создания и четкого определения и маркировки повторно используемых компонентов для размещения в библиотеке, а также небольшое количество повторно используемых и надежных программных ресурсов, доступных в настоящее время. Основные организационные проблемы включают в себя отсутствие приверженности повторному использованию, а также отсутствие надлежащей подготовки и вознаграждений, необходимых для его продвижения, отсутствие организационной поддержки институционализации повторного использования и трудности в измерении экономических выгод от повторного использования. Ройс (1998) утверждает, что из-за значительных затрат на разработку многоразового компонента большинство организаций не могут экономически конкурировать с устоявшимися коммерческими организациями, которые сосредоточены на продаже компонентов в качестве основного направления своей деятельности. Успех зависит от способности эффективно использовать стоимость компонентов для большого числа пользователей и проектной базы (рис. 2-5). Существуют также ключевые правовые и договорные вопросы, касающиеся повторного использования классов объектов и компонентов, первоначально использовавшихся в других приложениях (Kim and Stohr, 1998).

Когда руководство организации решает использовать повторное использование в качестве стратегии, важно, чтобы организация согласовывала свой подход к повторному использованию со своими стратегическими бизнес-целями (Griss, 2003). Преимущества повторного использования растут по мере того, как от него извлекается больше корпоративного опыта, но также растут затраты и количество ресурсов, необходимых для того, чтобы повторное использование работало хорошо. Повторное использование программного обеспечения состоит из трех основных этапов: абстракция, хранение и реконтекстуализация (Grinter, 2001). Абстракция включает в себя проектирование многократно используемого программного обеспечения, начиная с существующих программных активов или с нуля. Хранение включает в себя предоставление программных ресурсов для использования другими пользователями. Хотя это звучит как простая проблема, хранение на самом деле может быть очень сложным. Проблема заключается не только в том, чтобы положить программные активы на полку; Проблема заключается в правильной маркировке и каталогизации ресурсов, чтобы другие могли найти те, которые они хотят использовать. После того, как актив найден, реконтекстуализация становится важной. Это включает в себя то, чтобы сделать повторно используемый актив понятным для разработчиков, которые хотят использовать его в своих системах. Программное обеспечение является сложным, и программный актив, разработанный для конкретной системы в специфических для системы обстоятельствах, может вовсе не быть тем активом, которым он кажется. То, что кажется общим активом, называемым «клиентом», на самом деле может быть чем-то совершенно другим, в зависимости от контекста, в котором он был разработан. Часто может показаться, что проще просто создать свои собственные активы, чем вкладывать время и энергию, необходимые для того, чтобы хорошо понять программное обеспечение, разработанное кем-то другим. Ключевой частью стратегии повторного использования, как упоминалось ранее, является создание вознаграждений, стимулов и организационной поддержки для повторного использования, чтобы сделать его более выгодным, чем разработка собственных активов.

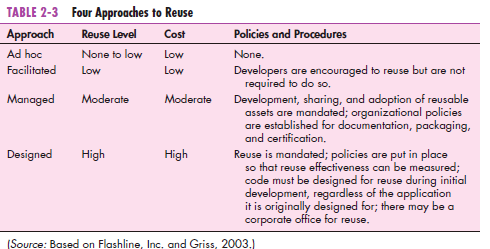


**Рисунок 2-5**

Инвестиции, необходимые для создания многоразовых компонентов

Согласно Griss (2003), организация может использовать один из четырех подходов к повторному использованию (T[able 2-3).](#bookmark0)  Специальный подход к повторному использованию на самом деле вообще не является подходом, по крайней мере, с официальной организационной точки зрения. При таком подходе индивидуумы могут свободно находить или разрабатывать повторно используемые активы самостоятельно, и существует мало организационных вознаграждений за повторное использование активов, если таковые имеются. Хранение не является проблемой, потому что люди отслеживают и распространяют свои собственные программные активы. При таком специальном, индивидуальном подходе трудно измерить какие-либо потенциальные выгоды для компании.

Другим подходом к повторному использованию является *упрощенное повторное* использование. При таком подходе разработчики не обязаны практиковать повторное использование, но их поощряют к этому. Организация предоставляет некоторые инструменты и методы, которые позволяют разрабатывать и совместно использовать повторно используемые активы, и одному или нескольким сотрудникам может быть назначена роль евангелиста для рекламы и продвижения программы. Однако очень мало делается для отслеживания качества и использования повторно используемых активов, и общие корпоративные инвестиции невелики.



Управляемое повторное использование — это более структурированный и более дорогостоящий способ управления повторным использованием программного обеспечения. При управляемом повторном использовании требуется разработка, совместное использование и внедрение повторно используемых ресурсов. Организация устанавливает процессы и политики для обеспечения практики повторного использования и измерения результатов. Организация также устанавливает политику и процедуры для обеспечения качества своих повторно используемых активов. Основное внимание уделяется выявлению существующих активов, которые потенциально могут быть повторно использованы из различных источников, в том числе из библиотек служебных активов, поставляемых с операционными системами, от компаний, которые продают активы, от сообщества с открытым исходным кодом, из внутренних репозиториев, от поиска существующего устаревшего кода и так далее.

Самым дорогим и экстенсивным подходом к повторному использованию является *разработанное повторное использование*. В дополнение к обязательному повторному использованию и измерению его эффективности, разработанный подход к повторному использованию делает дополнительный шаг, предписывающий, чтобы активы были предназначены для повторного использования, поскольку они разрабатываются для конкретных приложений. Основное внимание уделяется разработке повторно используемых активов, чем поиску существующих активов, которые могут быть кандидатами на повторное использование. Может быть создан корпоративный офис повторного использования для мониторинга и управления общей методологией. При таком подходе до 90 процентов программных активов могут быть повторно использованы в различных приложениях.

Каждый подход к повторному использованию имеет свои преимущества и недостатки. Ни один подход не является панацеей, которая решит головоломку повторного использования для всех организаций и для всех ситуаций. Успешное повторное использование требует понимания того, как повторное использование вписывается в более крупные организационные цели и стратегии, а также понимания социального и технического мира, в который должны вписываться повторно используемые активы.

### Резюме

Как системный аналитик, вы должны знать, где вы можете получить программное обеспечение, отвечающее некоторым или всем потребностям организации. Прикладное (и системное) программное обеспечение можно получить у фирм, предоставляющих услуги в области информационных технологий, поставщиков пакетного программного обеспечения, поставщиков программного обеспечения для решений в масштабах всего предприятия, поставщиков облачных вычислений и поставщиков программного обеспечения с открытым исходным кодом, а также из внутренних ресурсов разработки систем, включая повторное использование существующих программных компонентов. Вы даже можете нанять организацию для решения всех ваших потребностей в разработке систем, что называется *аутсорсингом*. Вы также должны знать критерии, которые следует использовать при выборе среди готовых программных продуктов. Эти критерии включают стоимость, функциональность, поддержку поставщиков, жизнеспособность поставщиков, гибкость, документацию, время отклика и простоту установки. Запросы предложений — это один из способов сбора дополнительных сведений о системном программном обеспечении, его производительности и затратах.

## Глава 3 Управление проектом «Информационные системы»

В главах 1 и 2 мы представили пять фаз жизненного цикла разработки систем (SDLC) и объяснили, как проект информационных систем проходит через эти пять фаз, в некоторых случаях неоднократно. В этой главе мы сосредоточимся на роли системного аналитика как руководителя проекта информационных систем. На протяжении всего SDLC менеджер проекта отвечает за инициирование, планирование, выполнение и закрытие проекта разработки систем. Управление проектами, пожалуй, самый важный аспект проекта разработки информационных систем. Эффективное управление проектами помогает гарантировать, что проекты по разработке систем соответствуют ожиданиям клиентов и выполняются в рамках бюджета и временных ограничений.

Сегодня наблюдается сдвиг в типах проектов, осуществляемых большинством фирм, что делает управление проектами гораздо более сложным и даже более важным для успеха проекта (Fuller et al., 2008; Schiff, 2014a). Например, в прошлом организации сосредотачивали большую часть своих разработок на очень больших, специально разработанных, автономных приложениях. Сегодня большая часть усилий по разработке систем в организациях сосредоточена на внедрении пакетного программного обеспечения, такого как системы планирования ресурсов предприятия (ERP) и хранилища данных. Существующие устаревшие приложения также модифицируются, чтобы транзакции между предприятиями могли беспрепятственно выполняться через Интернет. К существующим устаревшим системам добавляются новые веб-интерфейсы, с тем чтобы более широкий круг пользователей, зачастую распределенных по всему миру, мог получить доступ к корпоративной информации и системам. Кроме того, программное обеспечение, разработанное глобальными партнерами по аутсорсингу, которое должно быть интегрировано в существующий портфель приложений организации, в настоящее время является обычной практикой (Overby, 2013). Работа с поставщиками для поставки приложений, с клиентами или поставщиками для интеграции систем или с более широким и разнообразным сообществом пользователей требует высокой квалификации менеджеров проектов. Следовательно, важно, чтобы вы получили представление о процессе управления проектами; это станет критически важным навыком для вашего будущего успеха.

В этой главе мы сосредоточимся на роли системного аналитика в управлении проектами информационных систем и будем называть эту роль *менеджером проекта.* В первом разделе будет представлена предыстория для Pine Valley Furniture (PVF), производственной компании, которую мы посетим на протяжении оставшейся части книги. Затем мы предоставим вам понимание роли менеджера проекта и процесса управления проектом. Затем обсуждение переходит к методам составления отчетов о планах проектов с использованием диаграмм Ганта и сетевых диаграмм. Глава завершится обсуждением использования коммерчески доступного программного обеспечения для управления проектами, которое может быть использовано для оказания помощи в широком спектре деятельности по управлению проектами.

### Предыстория мебельной компании Pine Valley

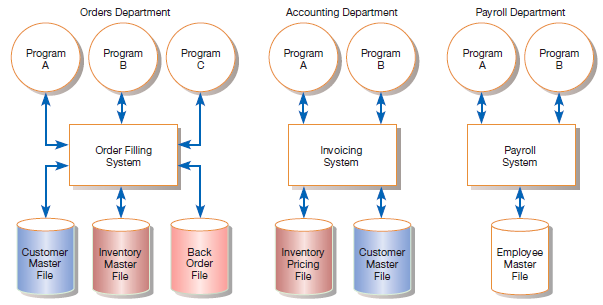
PVF производит высококачественную деревянную мебель и распространяет ее в розничных магазинах по всей территории Соединенных Штатов. Его продуктовые линейки включают столовые наборы, стереошкафы, навесные шкафы, мебель для гостиной и мебель для спальни. В начале 1980-х годов основатель PVF Алекс Шустер начал изготавливать и продавать мебель на заказ в своем гараже. Алекс управлял счетами-фактурами и отслеживал клиентов, используя папки с файлами и файловую сеть. К 1984 году бизнес расширился, и Алексу пришлось арендовать склад и нанять бухгалтера на неполный рабочий день. Продуктовая линейка PVF увеличилась в несколько раз, объем продаж удвоился, а штат сотрудников увеличился до 50 человек. К 1990 году PVF переехала в свое третье и нынешнее место. В связи с дополнительной сложностью деятельности компании, Алекс Реорга разделил компанию на следующие функциональные области:

* Производство, которое было далее разделено на три отдельные функции: изготовление, сборка и отделка.
* Сбыт
* Заказы
* Бухгалтерский учет
* Покупка

Алекс и руководители функциональных областей создали ручные информационные системы, такие как бухгалтерские книги и папки с файлами, которые какое-то время работали хорошо. В конце концов, однако, PVF выбрала и установила сетевой сервер для автоматизации выставления счетов, дебиторской задолженности и управления запасами.

Когда приложения были впервые компьютеризированы, каждое отдельное приложение имело свои собственные индивидуальные файлы данных, адаптированные к потребностям каждой функциональной области. Как это типично в таких ситуациях, приложения очень напоминали ручные системы, на которых они были основаны. На [рисунке 3-1](#bookmark0) показаны три компьютерных приложения в PVF: заполнение заказов, выставление счетов и расчет заработной платы. В конце 1990-х годов PVF сформировала целевую группу для изучения возможности перехода к подходу к базам данных. После предварительного изучения руководство решило перевести свои информационные системы на такой подход. Компания модернизировала свой сетевой сервер и внедрила централизованную систему управления базами данных. Сегодня PVF успешно развернула интегрированную базу данных в масштабах всей компании и преобразовала свои приложения для работы с базой данных. Тем не менее, PVF продолжает расти быстрыми темпами, оказывая давление на свои текущие прикладные системы.

Компьютерные приложения PVF поддерживают бизнес-процессы. Когда клиенты заказывают мебель, их заказы должны быть обработаны соответствующим образом: мебель должна быть изготовлена и отправлена нужному клиенту, а правильный счет-фактура отправлен по почте на правильный адрес. Сотрудники должны получать оплату за свою работу. Учитывая эти задачи, большинство компьютерных приложений PVF расположены в области бухгалтерского учета и финансов. Приложения включают заполнение заказов, выставление счетов, дебиторскую задолженность, управление запасами, кредиторскую задолженность, расчет заработной платы и главную книгу. В свое время каждое приложение имело свои собственные файлы данных. Например, существовали основной файл клиента, основной файл запасов, файл отложенных заказов, файл ценообразования запасов и основной файл сотрудника. Система заполнения заказов использовала данные из трех файлов: мастер клиента, мастер инвентаризации и отмену заказа. Однако сегодня все системы разрабатываются и интегрируются через общекорпоративную базу данных, в которой данные организованы вокруг сущностей или субъектов, таких как клиенты, счета-фактуры и заказы.



**РИСУНОК 3-1**

Три компьютерных приложения в PVF: заполнение заказов, выставление счетов и расчет заработной платы

PVF, как и многие фирмы, решила разработать свое прикладное программное обеспечение собственными силами; То есть, он нанял персонал и купил компьютерное оборудование и программное обеспечение, необходимое для создания прикладного программного обеспечения, соответствующего его собственным потребностям. (Другие методы, используемые для получения прикладного программного обеспечения, обсуждались в главе 2.) Хотя PVF продолжает расти быстрыми темпами, рыночные условия становятся чрезвычайно конкурентными, особенно с появлением Интернета. Давайте посмотрим, как менеджер проекта играет ключевую роль в разработке новой информационной системы для PVF.

### Управление проектом информационных систем

Управление проектами является важным аспектом развития информационных систем и критически важным навыком для системного аналитика. Основное внимание в управлении проектами уделяется обеспечению того, чтобы проекты по разработке систем соответствовали ожиданиям клиентов и выполнялись в рамках бюджета и временных ограничений.

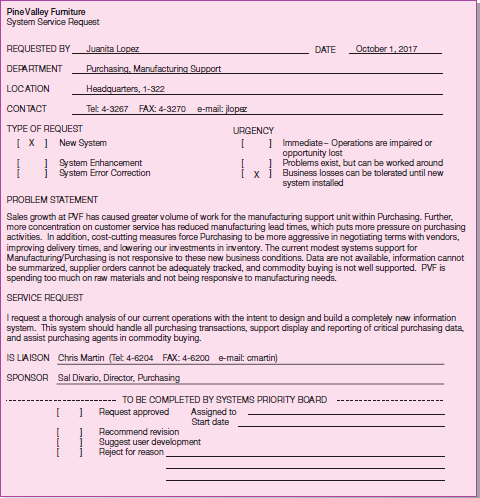
**Менеджер** проекта — это системный аналитик с разнообразным набором навыков — менеджмент, лидерство, технические навыки, управление конфликтами и отношения с клиентами, — который отвечает за инициирование, планирование, выполнение и закрытие проекта. Как руководитель проекта, ваша среда — это среда постоянных изменений и решения проблем. В некоторых организациях менеджер проекта является очень опытным системным аналитиком, в то время как в других ожидается, что эту роль возьмут на себя как младшие, так и старшие аналитики, управляя частями проекта или активно поддерживая более старшего коллегу, который берет на себя роль менеджера проекта. Понимание процесса управления проектами является критически важным навыком для вашего будущего успеха.

Создание и реализация успешных проектов требует управления ресурсами, действиями и задачами, необходимыми для завершения проекта информационных систем.  **Проект** представляет собой запланированное проведение ряда взаимосвязанных мероприятий для достижения цели, имеющей начало и конец. Первый вопрос, который вы можете задать себе: «Откуда берутся проекты?» и, рассмотрев все различные вещи, над которыми вас могут попросить работать в организации, «Как я узнаю, над какими проектами работать?» Способы, которыми каждая организация отвечает на эти вопросы, различаются.

В оставшейся части этого раздела мы опишем процесс, которому следовали Хуанита Лопес и Крис Мартин во время разработки системы выполнения закупок PVF. Хуанита работает в отделе Ордена, а Крис — системным аналитиком.

Хуанита заметила проблемы с обработкой заказов и отчетностью: рост продаж увеличил нагрузку на производственный отдел, а существующие системы больше не поддерживали должным образом отслеживание заказов. Становилось все труднее отслеживать заказы и получать нужную мебель и счета нужным клиентам. Хуанита связалась с Крисом, и вместе они разработали систему, которая устранила эти проблемы отдела Ордена.

Первым **конечным продуктом**, созданным Крисом и Хуанитой, был запрос на обслуживание системы (SSR), стандартная форма, используемая PVF для запроса работ по разработке систем. На рисунке 3-2 показан SSR для системы выполнения закупок. Форма включает в себя имя и контактную информацию лица, запрашивающего систему, изложение проблемы, а также имя и контактную информацию представителя и спонсора.



**РИСУНОК 3-2**

Системный запрос на обслуживание для покупки Система выполнения с именем и контактной информацией лица, запрашивающего систему, заявлением о проблеме, а также именем и контактной информацией представителя и спонсора

Затем эта просьба была рассмотрена Советом по системным приоритетам PVF. Поскольку все организации имеют ограниченное время и ресурсы, не все запросы могут быть одобрены. Совет директоров оценивает запросы на разработку в связи с бизнес-проблемами или возможностями, которые система решит или создаст; В нем также рассматривается вопрос о том, как предлагаемый проект вписывается в архитектуру информационных систем организации и долгосрочные планы развития. Наблюдательный совет отбирает те проекты, которые наилучшим образом соответствуют общим или организационным целям (подробнее об организационных целях мы узнаем в главе 4). В случае с запросом на систему выполнения закупок правление сочло запрос обоснованным и одобрил более подробное технико-экономическое обоснование.  **Технико-экономическое обоснование**, которое проводится менеджером проекта, включает в себя определение того, имеет ли информационная система смысл для организации с экономической и операционной точки зрения. Исследование проводится до того, как система будет построена. На рисунке 3-3 показано графическое представление шагов, выполненных во время запуска проекта системы выполнения закупок.

Таким образом, проекты по разработке систем предпринимаются по двум основным причинам: чтобы воспользоваться возможностями бизнеса и решить бизнес-проблемы. Использование возможности может означать предоставление инновационных услуг клиентам путем создания новой системы. Например, PVF может захотеть создать веб-сайт, чтобы клиенты могли легко получить доступ к его каталогу и размещать заказы в любое время. Решение бизнес-проблемы может включать в себя изменение способа обработки данных существующей системой таким образом, чтобы пользователям предоставлялась более точная и своевременная информация. Например, такая компания, как PVF, может создать защищенный паролем сайт в локальной сети, содержащий важные объявления и сведения о бюджете. Конечно, проекты не всегда инициируются по вышеупомянутым рациональным причинам (использование возможностей для бизнеса или решение бизнес-задач). Например, в некоторых случаях организации и правительство осуществляют проекты по расходованию ресурсов, достижению или заполнению бюджетов, чтобы занять людей или помочь обучить людей и развить их навыки. В этой главе мы сосредоточимся не на том, как и почему организации идентифицируют проекты, а на управлении проектами после того, как они были идентифицированы.

После того, как потенциальный проект определен, организация должна определить ресурсы, необходимые для его завершения. Это делается путем анализа объема проекта и определения вероятности успешного завершения. Получив эту информацию, организация может определить, возможно ли воспользоваться возможностью или решить конкретную проблему в рамках ограничений по времени и ресурсам. Если это считается возможным, проводится более детальный анализ проекта. Как вы увидите, способность определять размер, объем и требования к ресурсам проекта — это лишь один из многих навыков, которыми должен обладать менеджер проекта. Менеджер проекта часто рассматривается как жонглер, держащий в воздухе множество мячей, которые отражают различные аспекты развития проекта, как показано на рисунке 3-4.

Чтобы успешно организовать построение сложной информационной системы, менеджер проекта должен обладать межличностными, лидерскими и техническими навыками. В таблице 3-1 перечислены общие навыки и виды деятельности руководителя проекта. Обратите внимание, что многие навыки связаны с персоналом или общим управлением, а не просто с техническими навыками. Из таблицы 3-1 видно, что эффективный менеджер проекта не только обладает различными навыками, но и является наиболее важным человеком для успешного завершения любого проекта. Оставшаяся часть этой главы будет посвящена процессу **управления проектом**, который включает в себя четыре этапа:

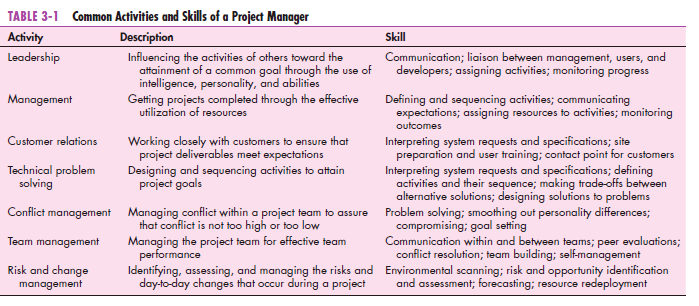
* Инициация проекта
* Планирование проекта
* Выполнение проекта
* Закрытие проекта



**РИСУНОК 3-4**

Менеджер проекта жонглирует многочисленными действиями

На каждом из этих четырех этапов необходимо выполнить несколько действий. Следование этому формальному процессу управления проектом значительно увеличивает вероятность успеха проекта .



#### Инициация проекта

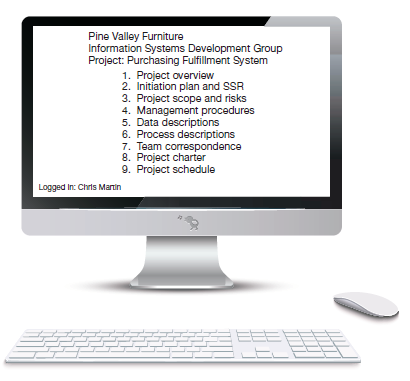
Во время **инициации** проекта руководитель проекта выполняет несколько действий для оценки размера, объема и сложности проекта и установления процедур для поддержки последующих действий. В зависимости от проекта, некоторые подготовительные мероприятия могут быть ненужными, а некоторые могут быть очень сложными. Типы действий, которые вы будете выполнять при запуске проекта, приведены на [рисунке 3-5](#bookmark0) и описаны далее.

1. *Создание команды инициации проекта.*  Эта деятельность включает в себя организацию первоначального ядра членов проектной группы для оказания помощи в выполнении мероприятий по инициированию проекта (Chau et al., 2012; «Верма», 1996; 1997). Например, во время проекта «Система выполнения заказов» в PVF Крису Мартину было поручено поддерживать отдел закупок. Политика PVF заключается в том, что все группы инициации состоят как минимум из одного представителя пользователя, в данном случае Хуаниты Лопес, и одного члена группы разработчиков информационных систем (ИС). Поэтому команда инициаторов проекта состояла из Криса и Хуаниты; Крис был менеджером проекта.
2. *Налаживание отношений с заказчиком.*  Глубокое понимание вашего клиента создает более прочные партнерские отношения и более высокий уровень доверия. В PVF руководство пыталось наладить прочные рабочие отношения между бизнес-подразделениями (например, отделом закупок) и группой разработки ИБ, назначая конкретного человека для работы в качестве связующего звена между обеими группами. Поскольку Крис был назначен в отдел закупок в течение некоторого времени, он уже знал о некоторых проблемах с существующими системами закупок. Политика PVF по назначению конкретных лиц в каждое бизнес-подразделение помогла гарантировать, что и Крису, и Хуаните будет комфортно работать вместе до начала проекта. Многие организации используют аналогичный механизм для установления отношений с клиентами.
3. *Разработка плана инициации проекта.*  На этом этапе определяются действия, необходимые для организации группы инициации, в то время как она работает над определением целей и масштабов проекта (Abdel-Hamid et al., 1999). Роль Криса заключалась в том, чтобы помочь Хуаните перевести ее бизнес-требования в письменный запрос на улучшение информационной системы. Это потребовало сбора, анализа, организации и преобразования большого количества информации. Поскольку Крис и Хуанита уже были знакомы друг с другом и их ролями в рамках проекта разработки, им нужно было определить, когда и как они будут общаться, определить результаты и этапы проекта, а также установить сроки. Их план инициации включал повестки дня нескольких встреч. Эти шаги в конечном итоге привели к созданию их формы SSR.
4. *Установление процедур управления.*  Успешные проекты требуют разработки эффективных управленческих процедур. В рамках PVF многие из этих управленческих процедур были установлены в качестве стандартных оперативных процедур Советом по системным приоритетам и группой разработчиков ИС. Например, вся работа по разработке проекта возлагается на функциональное подразделение, запросившее работу. В других организациях каждый проект может иметь уникальные процедуры, адаптированные к его потребностям. Тем не менее, как правило, при установлении процедур вы занимаетесь разработкой процедур коммуникации и отчетности в команде, назначением должностей и ролей, процедурами изменения проекта и определением того, как будет осуществляться финансирование проекта и выставление счетов. Крису и Хуаните повезло, что большинство из этих процедур уже были установлены в PVF, что позволило им перейти к другим проектным мероприятиям.
5. *Создание среды управления проектами и рабочей тетради проекта.*  Основное внимание в этом упражнении уделяется сбору и организации средств, которые будут использоваться при управлении проектом, а также созданию книги проекта. Диаграммы, диаграммы и описания систем предоставляют большую часть содержимого книги проекта. Таким образом, рабочая тетрадь проекта служит хранилищем всей корреспонденции по проекту, входных данных, выходных данных, результатов, процедур и стандартов, установленных проектной группой (Rettig, 1990; Динсмор и Кабанис-Брюин, 2006). Рабочая **тетрадь** проекта может храниться в виде электронного документа в Интернете или в большой папке с тремя кольцами. Рабочая тетрадь проекта используется всеми членами команды и полезна для аудита проекта, ориентации новых членов команды, общения с руководством и клиентами, определения будущих проектов и проведения послепроектных проверок. Создание рабочей тетради и тщательная запись всей информации о проекте являются двумя наиболее важными действиями, которые вы будете выполнять в качестве менеджера проекта.

* На рисунке 3-6 показана рабочая тетрадь проекта для системы выполнения закупок в интрасети PVF. Ведение рабочей тетради проекта в режиме онлайн имеет много преимуществ для поддержания работы проектной группы в нужном русле и эффективности. Онлайн-документы могут быть легко доступны всем членам команды. Кроме того, все всегда работают с самой актуальной информацией. Лучшая особенность использования Интернета в качестве репозитория заключается в том, что он позволяет участникам проекта и клиентам постоянно просматривать статус проекта и всю связанную с ним информацию.

1. *Разработка устава проекта.*  Уставпроекта представляет собой короткий (обычно одностраничный) высокоуровневый документ, подготовленный для клиента, в котором описывается, что будет предоставлено проектом, и излагаются многие ключевые элементы проекта. Устав проекта может различаться по количеству деталей, но часто включает в себя следующие элементы:

* Название проекта и дата авторизации
* Ф.И.О. руководителя проекта и контактная информация



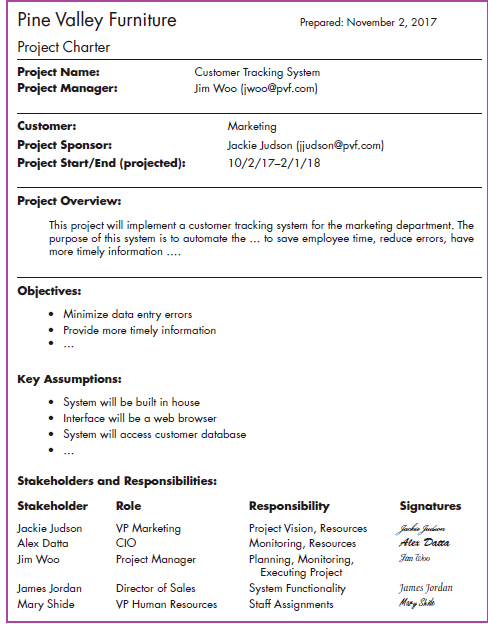
**РИСУНОК 3-6**

Рабочая тетрадь проекта «Система выполнения закупок» содержит девять ключевых элементов

* Имя клиента и контактная информация
* Предполагаемые даты начала и завершения
* Ключевые заинтересованные стороны, роль и обязанности проекта
* Цели и описание проекта
* Ключевые допущения или подход
* Раздел подписи для ключевых заинтересованных сторон

Устав проекта гарантирует, что и вы, и ваш клиент получите общее понимание проекта. Это также очень полезный инструмент общения; Он помогает объявить организации о том, что для разработки выбран тот или иной проект. Пример устава проекта показан на [рисунке 3-7.](#bookmark0)

Инициация проекта завершается после выполнения этих шести мероприятий. Прежде чем перейти к следующему этапу проекта, работа, выполненная во время инициации проекта, рассматривается на совещании, на котором присутствуют руководство, клиенты и члены проектной группы. Результатом этой встречи является решение о продолжении, изменении или выходе из проекта. В случае проекта «Система выполнения закупок» в PVF правление приняло SSR и выбрало руководящий комитет проекта для мониторинга хода выполнения проекта и предоставления рекомендаций членам команды во время последующих действий. Если масштабы проекта будут изменены, может возникнуть необходимость вернуться к мероприятиям по инициированию проекта и собрать дополнительную информацию. После того, как принято решение о продолжении проекта, на этапе планирования проекта разрабатывается гораздо более подробный план проекта.



**РИСУНОК 3-7**

Устав предлагаемого проекта по информационным системам

#### Планирование проекта

Следующим шагом в процессе управления проектом является **планирование проекта**. Исследования выявили положительную взаимосвязь между эффективным планированием проектов и положительными результатами проекта (Guinan et al., 1998; Кирш, 2000). Планирование проекта включает в себя определение четких, дискретных мероприятий и работы, необходимой для выполнения каждого мероприятия в рамках проекта. Это часто требуется, чтобы вы делали многочисленные предположения о доступности ресурсов, таких как оборудование, программное обеспечение и персонал. Гораздо легче планировать ближайшую деятельность, чем ту, которая произойдёт в будущем. На самом деле, вам часто приходится строить долгосрочные планы, которые являются более общими по объему, и краткосрочные планы, которые являются более подробными. Повторяющийся характер процесса управления проектом требует, чтобы планы постоянно контролировались на протяжении всего проекта и периодически обновлялись (обычно после каждого этапа) на основе самой последней информации. [На рисунке 3-8](#bookmark0) показан принцип, согласно которому краткосрочные планы, как правило, являются более конкретными и жесткими, чем долгосрочные планы. Например, практически невозможно строго спланировать деятельность на поздних этапах проекта, не завершив предварительно предыдущую деятельность. Кроме того, результаты мероприятий, выполненных ранее в рамках проекта, скорее всего, повлияют на последующие виды деятельности. Это означает, что очень сложно и, скорее всего, неэффективно пытаться спланировать детальные решения для действий, которые произойдут в далеком будущем.

Как и в случае с процессом инициации проекта, во время планирования проекта должны выполняться разнообразные и многочисленные мероприятия. Например, во время проекта «Система выполнения закупок» Крис и Хуанита разработали 10-страничный план. Однако планы проектов для очень больших систем могут занимать несколько сотен страниц. Типы действий, которые можно выполнять во время планирования проекта, приведены на рисунке 3-9 и описаны в следующем списке:

1. *Описание объема проекта, альтернатив и осуществимости.* Цель этого упражнения - понять содержание и сложность проекта. В рамках методологии разработки систем PVF одна из первых встреч должна быть посвящена определению объема проекта. Хотя информация об объеме проекта не была включена в SSR, разработанную Крисом и Хуанитой, было важно, чтобы оба разделяли одно и то же видение проекта, прежде чем заходить слишком далеко. В ходе этого упражнения вы должны прийти к соглашению по следующим вопросам:
   * Какую проблему или возможность решает проект?
   * Каких поддающихся количественной оценке результатов необходимо достичь?
   * Что нужно сделать?
   * Как будет измеряться успех?
   * Как мы узнаем, когда закончим?

После определения масштаба проекта вашей следующей целью является определение и документирование общих альтернативных решений для текущей бизнес-проблемы или возможности. Затем необходимо оценить осуществимость каждого альтернативного решения и выбрать, какое из них следует учитывать на последующих этапах SDLC. В некоторых случаях можно найти готовое программное обеспечение. Также важно, чтобы любые уникальные проблемы, ограничения и предположения о проекте были четко указаны.

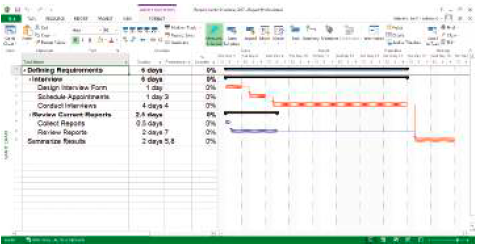


РИСУНОК 3-9

Этот проект является проектом, который является объектом

1. *Разделение проекта на управляемые задачи.* Это критически важная деятельность в процессе планирования проекта. Здесь вы должны разделить весь проект на управляемые задачи, а затем логически упорядочить их, чтобы обеспечить плавную эволюцию между задачами. Определение задач и их последовательность называется структурой разбивки **работ** (PMBOK, 2013; Институт управления проектами, 2002). Некоторые задачи могут выполняться параллельно, тогда как другие должны следовать одна за другой последовательно. Последовательность задач зависит от того, какие задачи дают результаты, необходимые для выполнения других задач, когда доступны критически важные ресурсы, ограничения, налагаемые на проект клиентом, и процесс, описанный в SDLC.

Например, предположим, что вы работаете над новым проектом разработки и вам нужно собрать системные требования, опросив пользователей новой системы и просмотрев отчеты, которые они в настоящее время используют для выполнения своей работы. Разбивка работ по этим видам деятельности представлена на диаграмме Ганта на рисунке 3-10. Диаграмма Ганта — это графическое представление проекта, которое показывает каждую задачу в виде горизонтальной полосы, длина которой пропорциональна времени ее выполнения. Различные цвета, оттенки или формы можно использовать для выделения каждого типа задач. Например, действия на критическом пути (определенном позже) могут быть выделены красным цветом, а суммарная задача может иметь специальную полосу. Обратите внимание, что черные горизонтальные полосы — строки 1, 2 и 6 на рисунке 3-10 — представляют собой суммарные задачи. Запланированное и фактическое время или ход выполнения действия можно сравнить с помощью параллельных полос разных цветов, оттенков или форм. Диаграммы Ганта (как правило) не показывают, как должны быть упорядочены задачи (приоритет), а просто показывают, когда действие должно начинаться и заканчиваться. На рисунке 3-10 продолжительность задачи показана во втором столбце по дням, «d», а необходимые предыдущие задачи отмечены в третьем столбце как предшественники. Большинство программных инструментов для управления проектами поддерживают широкий диапазон длительности задач, включая минуты, часы, дни, недели и месяцы. Как вы узнаете в последующих главах, SDLC состоит из нескольких этапов, которые вам нужно будет разбить на действия. Создание структурной декомпозиции работ требует разложения этапов на действия (суммарные задачи) и действия на конкретные задачи. [Например, на рисунке 3-10](#bookmark0) показано, что упражнение «Собеседование» состоит из трех задач: разработать форму интервью, назначить встречи и провести собеседование.



**РИСУНОК 3-10**

Диаграмма Ганта, показывающая задачи проекта, продолжительность этих задач и предшественников

Слишком подробное определение задач сделает управление проектом излишне сложным. Вы разовьете навык определения оптимального уровня хвоста для представления задач через опыт. Например, может быть очень трудно перечислить задачи, для выполнения которых требуется менее одного часа времени, в окончательной структуре разбивки работ. Кроме того, выбор задач слишком большого объема (например, продолжительностью в несколько недель) не даст вам четкого представления о статусе проекта или взаимозависимостях между задачами. Каковы характеристики «задачи»? Задача

* может быть выполнен одним человеком или четко определенной группой,
* имеет единый и идентифицируемый конечный результат (задача, однако, заключается в процессе создания конечного результата),
* Имеет известный способ или методику,
* имеет хорошо принятые шаги предшественника и преемника, и
* поддается измерению, чтобы можно было определить процент завершения.

1. *Оценка ресурсов и создание плана ресурсов.* Цель этого действия — оценить потребности в ресурсах для каждого мероприятия по проекту и использовать эти сведения для создания плана ресурсов проекта. План ресурсов помогает собрать и развернуть ресурсы наиболее эффективным способом. Например, вы не хотели бы привлекать дополнительных программистов к проекту быстрее, чем вы могли бы подготовить для них работу. Менеджеры проектов используют различные инструменты для оценки размера и стоимости проекта. Наиболее широко используемый метод называется COCOMO (**COnstructive COst MOdel**), в котором используются параметры, полученные из предыдущих проектов различной сложности (Boehm et al., 2000). COCOMO использует эти различные параметры для определения потребностей в людских ресурсах для базовых, промежуточных и очень сложных систем. Люди являются наиболее важной и дорогостоящей частью планирования ресурсов проекта. Оценка времени выполнения задач и общее качество системы в значительной степени зависят от назначения людей на задачи. Важно давать людям задания, которые позволяют им осваивать новые навыки. Не менее важно убедиться, что участники проекта не находятся «над головой» или не работают над задачей, которая не очень хорошо подходит для их навыков. Оценки ресурсов, возможно, придется пересмотреть в зависимости от навыков фактического лица (или людей), назначенных для конкретной деятельности. На рисунке 3-11 показана относительная скорость программирования в сравнении с относительным качеством программирования трех программистов. Рисунок говорит о том, что Карлу не следует поручать задачи, в которых время выполнения имеет решающее значение, и что Бренде следует назначать задачи, в которых высокое качество является наиболее важным.

Один из подходов к назначению задач заключается в назначении одного типа задач (или только нескольких типов задач) каждому работнику на время проекта. Например, можно назначить одного работника для создания всех дисплеев компьютера, а другого — для создания всех системных отчетов. Такая специализация гарантирует, что оба работника станут эффективными в своих конкретных задачах. Работнику может стать скучно, если задача слишком узкая или длительная, поэтому вы можете назначить работников на более широкий круг задач. Как бы то ни было, такой подход может привести к снижению эффективности задач. Золотой серединой было бы выполнение заданий с балансом как специализации, так и разнообразия задач. Задания зависят от размера проекта разработки и навыков проектной команды. Независимо от того, как вы назначаете задачи, убедитесь, что каждый член команды работает только над одной задачей за раз. Исключения из этого правила могут возникать, когда задача занимает лишь небольшую часть времени члена команды (например, тестирование программ, разработанных другим членом команды) или во время чрезвычайной ситуации.

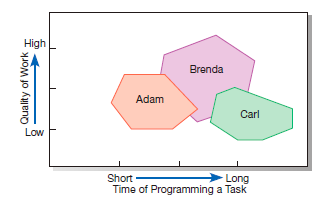
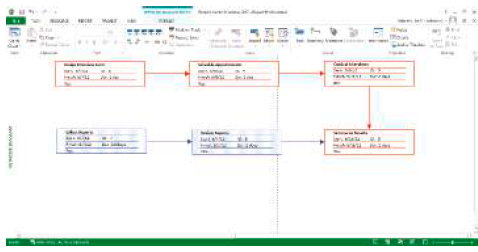


РИСУНОК 3-11

Компромиссы между качеством программного кода и скоростью программирования

1. *Разработка предварительного графика*. Во время этого действия вы используете сведения о задачах и доступности ресурсов, чтобы назначить оценки времени для каждого действия в структуре декомпозиции работ. Эти оценки времени позволят вам создать целевые даты начала и окончания проекта. Контрольные даты могут быть пересмотрены и изменены до тех пор, пока не будет составлен график, приемлемый для клиента. Для определения приемлемого графика может потребоваться найти дополнительные или другие ресурсы или изменить объем проекта. График может быть представлен в виде диаграммы Ганта, как показано на рисунке 3-10, или в виде сетевой диаграммы, как показано на рисунке 3-12. Сетевая диаграмма — это графическое изображение задач проекта и их взаимосвязей. Как и в случае с диаграммой Ганта, каждый тип задачи может быть выделен различными функциями на сетевой диаграмме. Отличительной чертой сетевой диаграммы является то, что упорядочение задач показано путем соединения задач, изображенных в виде прямоугольников или овалов, с их предыдущими и последующими задачами. Однако относительный размер узла (представляющего задачу) или промежуток между узлами не означает продолжительность задачи. На сетевой диаграмме рисуются только отдельные элементы задач, поэтому суммарные задачи 1, 2 и 6 — черные полосы — с рисунков 3-10 не показаны на рисунке 3-12. Мы опишем обе эти диаграммы далее в этой главе.



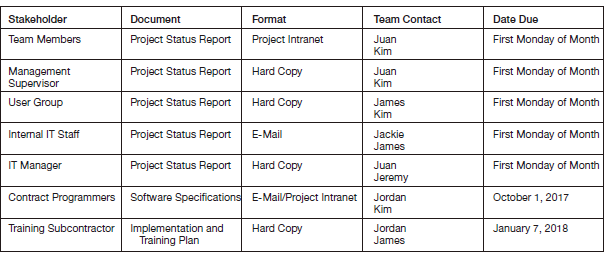
**РИСУНОК 3-12**

Сетевая диаграмма иллюстрирует задачи прямоугольниками (или овалами), а отношения и последовательности этих действий — стрелками

1. *Разработка коммуникационного плана.*  Цель этого мероприятия состоит в том, чтобы наметить процедуры коммуникации между руководством, членами проектной группы и заказчиком. План коммуникации включает в себя, когда и как команда будет предоставлять письменные и устные отчеты, как члены команды будут координировать работу, какие сообщения будут отправлены для объявления проекта заинтересованным сторонам и какие виды информации будут переданы поставщикам и внешним подрядчикам, участвующим в проекте. Важно, чтобы между всеми сторонами происходило свободное и открытое общение с учетом конфиденциальной информации и конфиденциальности с клиентом (Fuller et al., 2008; Кеттельхут, 1991; Кирш, 2000; «Вайдьянатан», 2013; Верма, 1996). При разработке плана коммуникации необходимо ответить на многочисленные вопросы, чтобы убедиться, что план является всеобъемлющим и полным, включая следующие:
   * Кто является заинтересованными сторонами этого проекта?
   * Какая информация нужна каждой заинтересованной стороне?
   * Когда и через какой промежуток времени эта информация должна быть получена?
   * Какие источники будут использоваться для сбора и генерации этой информации?
   * Кто будет собирать, хранить и проверять точность этой информации?
   * Кто будет систематизировать и упаковывать эту информацию в документ?
   * Кто будет контактным лицом для каждой заинтересованной стороны в случае возникновения каких-либо вопросов?
   * Какой формат будет использоваться для упаковки этой информации?
   * Какое средство коммуникации будет наиболее эффективным для доведения этой информации до заинтересованных сторон?

После того, как ответы на эти вопросы будут даны каждой заинтересованной стороне, может быть разработан комплексный план коммуникации. В этом плане будет изложено резюме коммуникационных документов, рабочих заданий, графиков и методов распределения. Кроме того, может быть разработана коммуникационная матрица проекта, которая содержит краткое изложение общего плана коммуникации (см. рис. 3-13). Эта матрица может быть легко распространена среди членов команды и проверена заинтересованными сторонами за пределами проектной группы, чтобы нужные люди получали нужную информацию в нужное время и в нужном формате.

1. *Определение стандартов и процедур проекта.*  Во время этого упражнения вы укажете, как различные результаты производятся и тестируются вами и вашей проектной командой. Например, команда должна решить, какие инструменты использовать, как можно изменить стандартный SDLC, какие методы SDLC будут использоваться, стили документации (например, шрифты и поля для руководств пользователя), как члены команды будут переносить статус назначенных им действий и терминологию. Установление стандартов проекта и процедур приемки работ является способом обеспечения разработки качественной системы. Кроме того, гораздо легче обучать новых членов команды, когда есть четкие стандарты. Организационные стандарты управления проектами и их проведения облегчают определение индивидуальных стандартов проекта и делают возможным обмен или совместное использование персонала между различными проектами.
2. *Идентификация и оценка риска.* Целью этой деятельности является выявление источников проектного риска и оценка последствий этих рисков (Wideman, 1992). Риски могут возникать из-за использования новых технологий, сопротивления потенциальных пользователей изменениям, наличия критически важных ресурсов, реакции конкурентов или изменений в регулирующих действиях из-за построения системы или неопытности членов команды в области технологий или бизнес-деятельности. Вы должны постоянно пытаться определить и оценить риск проекта.



**РИСУНОК 3-13**

Коммуникационная матрица проекта представляет собой высокоуровневое резюме плана коммуникации

Идентификация проектного риска необходима для разработки новой системы выполнения закупок PVF. Крис и Хуанита встретились, чтобы определить и описать возможные негативные последствия проекта и их вероятности. Несмотря на то, что мы перечисляем идентификацию рисков и определение объема проекта как два отдельных вида деятельности, они тесно связаны между собой и часто обсуждаются одновременно.

1. *Создание предварительного бюджета.* На этом этапе вам необходимо создать предварительный бюджет, в котором указаны запланированные расходы и доходы, связанные с вашим проектом. Обоснование проекта продемонстрирует, что выгоды стоят этих затрат. На рисунке 3-14 показан анализ затрат и выгод для нового проекта развития. Этот анализ показывает расчет чистой приведенной стоимости выгод и затрат проекта, а также анализ рентабельности инвестиций и денежных потоков. Мы подробно обсуждаем бюджеты проектов в главе 5.
2. *Разработка технического задания проекта.* Важным мероприятием, которое происходит ближе к концу этапа планирования проекта, является разработка Заявления об объеме проекта. В этом документе, разработанном в первую очередь для заказчика, описывается работа, которая будет выполнена, и четко описывается, что даст проект. Заявление об объеме проекта полезно для того, чтобы убедиться, что вы, клиент и другие члены проектной группы имеете четкое представление о предполагаемом размере, продолжительности и результатах проекта.
3. *Настройка базового плана проекта.* После того, как все предыдущие действия по планированию проекта будут завершены, вы сможете разработать базовый план проекта. Этот базовый план содержит оценку задач проекта и потребностей в ресурсах и используется для руководства следующим этапом проекта — выполнением. По мере поступления новой информации в ходе выполнения проекта базовый план будет продолжать обновляться.

Идентификация проектного риска необходима для разработки новой системы выполнения закупок PVF. Крис и Хуанита встретились, чтобы определить и описать возможные негативные последствия проекта и их вероятности. Несмотря на то, что мы перечисляем идентификацию рисков и определение объема проекта как два отдельных вида деятельности, они тесно связаны между собой и часто обсуждаются одновременно.

#### Выполнение проекта

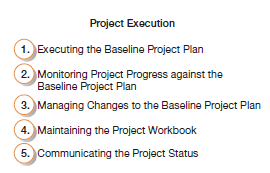
**Выполнение проекта** приводит в действие базовый план проекта. В контексте SDLC выполнение проекта происходит в основном на этапах анализа, проектирования и реализации. Во время разработки системы выполнения закупок Крис Мартин отвечал за пять ключевых действий во время выполнения проекта. Эти действия кратко изложены на рисунке 3-15 и описаны в оставшейся части этого раздела:

1. *Выполнение базового плана проекта*



**РИСУНОК 3-14**

Анализ финансовых затрат и выгод для проекта разработки систем (*Источник:* корпорация Майкрософт).



**РИСУНОК 3-15**

Пять мероприятий по реализации проекта

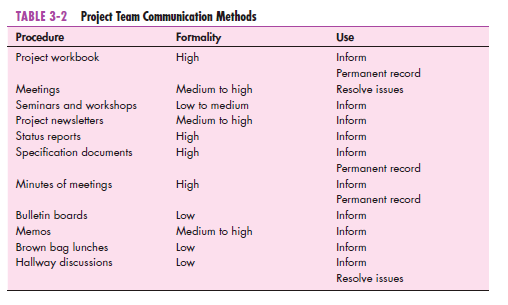
1. *Мониторинг хода выполнения проекта в соответствии с базовым планом проекта.*  Во время выполнения базового плана проекта следует следить за ходом выполнения. Если проект опережает (или отстает) от графика, вам, возможно, придется скорректировать ресурсы, действия и бюджеты. Мониторинг проектной деятельности может привести к внесению изменений в текущий план. Измерение времени и усилий, затрачиваемых на каждое действие, поможет вам подтвердить точность оценок для будущих проектов. С помощью диаграмм графика проекта, таких как диаграммы Ганта, можно показать прогресс по сравнению с планом, а с помощью сетевых диаграмм легко понять последствия задержек в деятельности. Мониторинг прогресса также означает, что руководитель группы должен оценивать и оценивать каждого члена команды, время от времени менять рабочие задания или запрашивать изменения в персонале, а также предоставлять обратную связь руководителю сотрудника.
2. *Управление изменениями базового плана проекта.* Вы столкнетесь с давлением, чтобы внести изменения в базовый план. В PVF политики предписывают, что в спецификацию проекта могут быть внесены только утвержденные изменения, и все изменения должны быть отражены в базовом плане и рабочей тетради проекта, включая все диаграммы. Например, если Хуанита предлагает внести существенные изменения в существующую структуру системы выполнения закупок, официальный запрос на изменение должен быть одобрен руководящим комитетом. В запросе должно быть объяснено, почему желательны изменения, и описаны все возможные последствия для предыдущих и последующих действий, ресурсов проекта и общего графика проекта. Крис должен был помочь Хуаните разработать такую просьбу. Эта информация позволяет руководящему комитету проекта легче оценить затраты и выгоды от значительных изменений на промежуточном этапе. В дополнение к изменениям, происходящим по официальному запросу, изменения также могут происходить из-за событий, не зависящих от вас. Фактически, многочисленные события могут инициировать изменение базового плана проекта, включая следующие возможности:
   * Сдвинутая дата завершения действия
   * Неуклюжее занятие, которое необходимо переделать
   * Идентификация нового вида деятельности, который становится очевидным позже в рамках проекта
   * Непредвиденные изменения в персонале из-за болезни, увольнения или увольнения

Когда происходит событие, которое задерживает завершение действия, у вас обычно есть два варианта: придумать способ вернуться к графику или пересмотреть план. Разработка способа вернуться к графику является предпочтительным подходом, потому что никаких изменений в план вносить не придется. Способность отвлечься и плавно обходить проблемы — критически важный навык, которым вам необходимо овладеть.

Как вы увидите далее в этой главе, графики проекта очень полезны для оценки влияния изменений. Используя такие диаграммы, вы можете быстро увидеть, повлияют ли на время завершения других действий изменения продолжительности данного действия или изменится дата завершения всего проекта. Часто вам придется найти способ перестроить действия, потому что окончательная дата завершения проекта может быть довольно фиксированной. Организация может быть наказана (даже судебным иском), если ожидаемая дата завершения не будет соблюдена.

1. *Наблюдение за книгой проекта*. Как и на всех этапах проекта, необходимо вести полный учет всех событий проекта. Рабочая тетрадь содержит документацию, необходимую новым членам команды для быстрого усвоения задач проекта. Он объясняет, почему были приняты проектные решения, и является основным источником информации для подготовки всех отчетов по проекту.
2. *Информирование о статусе проекта*. Руководитель проекта отвечает за то, чтобы все заинтересованные стороны — разработчики систем, менеджеры и клиенты — были в курсе состояния проекта. Другими словами, информирование о статусе проекта сосредоточено на *выполнении* плана коммуникации проекта и ответе на любые специальные информационные запросы заинтересованных сторон. Для распространения информации можно использовать широкий спектр методов, каждый из которых имеет свои сильные и слабые стороны. Некоторые способы проще для отправителя информации, но сложнее или менее удобны для получателя. С развитием цифровых сетей и Интернета происходит все больше и больше обмена цифровыми коммуникациями. Процедуры информирования о деятельности по проекту варьируются от официальных встреч до неформальных обсуждений в коридорах. Некоторые процедуры полезны для информирования других о статусе проекта, другие лучше для решения проблем, а третьи лучше для ведения постоянного учета информации и событий. На протяжении всего проекта обычно обмениваются двумя типами информации: результатами *работы* — результатами различных задач и действий, которые выполняются для завершения проекта, — и *планом* проекта — формальным всеобъемлющим документом, который используется для выполнения проекта; он содержит множество элементов, включая устав проекта, график проекта, бюджеты и план рисков. В таблице 3-2 перечислены многочисленные процедуры связи, уровень их формальности и наиболее вероятное использование. Какую бы процедуру вы ни использовали, частое общение помогает обеспечить успех проекта (Kettelhut, 1991; Кирш, 2000; Верма, 1996).

В этом разделе описана ваша роль в качестве руководителя проекта во время выполнения базового плана проекта. Легкость, с которой можно управлять проектом, в значительной степени зависит от качества предыдущих этапов проекта. Если разработать качественный план проекта, то гораздо больше шансов, что проект будет успешно выполнен. В следующем разделе описывается ваша роль во время закрытия проекта, заключительного этапа процесса управления проектом.



#### Закрытие проекта

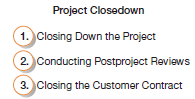
Целью **закрытого** проекта является доведение проекта до конца. Проекты могут завершиться естественным или неестественным прекращением. Естественное завершение происходит, когда требования проекта выполнены — проект завершен и является успешным. Неестественное прекращение происходит, когда проект останавливается до завершения (Keil et al., 2000). Несколько событий могут привести к неестественному завершению проекта. Например, можно узнать, что предположение, использованное для руководства проектом, оказалось ложным, что производительность систем или группы разработчиков была каким-то образом неадекватной, или что требования больше не актуальны или действительны в бизнес-среде клиента. Наиболее вероятные причины неестественного прекращения проекта связаны с нехваткой времени или денег, или и тем, и другим. Независимо от результата завершения проекта, необходимо выполнить несколько действий: закрытие проекта, проведение послепроектных проверок и закрытие контракта с заказчиком. В рамках SDLC закрытие проекта происходит после этапа реализации. Этап обслуживания системы обычно представляет собой непрерывную серию проектов, каждый из которых должен управляться индивидуально. На рисунке 3-17 кратко показана деятельность по закрытию проекта, которая более подробно описана в оставшейся части этого раздела:

1. *Закрытие проекта.* Во время закрытия вы выполняете несколько разнообразных действий. Например, если с вами работает несколько членов команды, завершение проекта может означать изменение работы и назначения для некоторых участников. Скорее всего, вам потребуется оценить каждого члена команды и предоставить оценку для личных дел и определения заработной платы. Вы также можете давать советы по карьере членам команды, писать письма начальству, восхваляя особые достижения членов команды, и отправлять благодарственные письма тем, кто помогал, но не был членом команды. Как руководитель проекта, вы должны быть готовы к возможным негативным кадровым проблемам, таким как увольнение, особенно если проект не увенчался успехом. При закрытии проекта также важно уведомить все заинтересованные стороны о том, что проект завершен, и завершить всю проектную документацию и финансовые отчеты, чтобы можно было провести окончательный обзор проекта. Вы также должны отметить достижения команды. Некоторые команды проведут вечеринку, и каждый член команды может получить памятные вещи (например, футболку с надписью «Я пережил проект X»). Цель состоит в том, чтобы отметить усилия команды по успешному завершению сложной задачи.

2. *Проведение постпроектных обзоров.* После того, как вы закрыли проект, следует провести окончательные проверки проекта с руководством и клиентами. Целью этих обзоров является определение сильных и слабых сторон результатов проекта, процессов, используемых для их создания, и процесса управления проектом. Важно, чтобы все понимали, что пошло правильно, а что пошло не так, чтобы улучшить процесс для следующего проекта. Помните, что методология разработки систем, принятая организацией, является живым руководством, которое должно постоянно совершенствоваться.

3. *Закрытие договора с клиентом.* Основное внимание в этом заключительном мероприятии уделяется обеспечению выполнения всех договорных условий проекта. Проект, регулируемый договорным соглашением, обычно не завершается до тех пор, пока не будет достигнуто согласие обеих сторон, часто в письменной форме. Таким образом, крайне важно, чтобы вы получили согласие от своего клиента на то, что все договорные обязательства были выполнены и что дальнейшая работа либо является его обязанностью, либо покрывается другим SSR или контрактом.

Закрытие - очень важное занятие. Проект не является завершенным до тех пор, пока он не будет закрыт, и именно при закрытии проекты считаются успешными или неудачными. Завершение также означает возможность начать новый проект и применить то, что вы узнали. Теперь, когда у вас есть представление о процессе управления проектами, в следующем разделе описываются конкретные методы, используемые при разработке систем для представления и планирования действий и ресурсов.



**РИСУНОК 3-17**

Три проекта закрыты собственные мероприятия

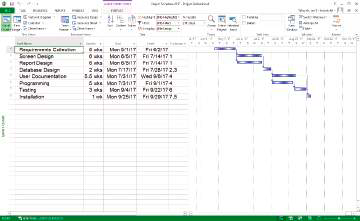
### Представление и планирование планов проекта

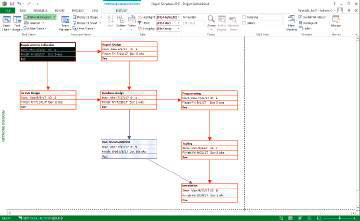
Менеджер проекта имеет широкий спектр методов, доступных для изображения и документирования планов проекта. Эти документы планирования могут принимать форму графических или текстовых отчетов, хотя графические отчеты стали наиболее популярными для отображения планов проектов. Наиболее часто используемыми методами являются диаграммы Ганта и сетевые диаграммы. Поскольку диаграммы Ганта (как правило) не показывают, как задачи должны быть упорядочены (приоритет), а просто показывают, когда задача должна начинаться и когда она должна заканчиваться, они часто более полезны для изображения относительно простых проектов или частей более крупного проекта, демонстрации деятельности одного работника или мониторинга хода выполнения действий по сравнению с запланированными датами завершения [(рис. 3-18).](#bookmark0) Напомним, что сетевая диаграмма показывает упорядочение действий путем соединения задачи с ее предыдущими и последующими задачами. Иногда предпочтительнее сетевая диаграмма; в других случаях диаграмма Ганта легче показывает определенные аспекты проекта. Вот ключевые различия между этими двумя диаграммами:

* Диаграммы Ганта наглядно показывают продолжительность задач, тогда как сетевая диаграмма наглядно показывает зависимости последовательности между задачами.
* Диаграммы Ганта визуально показывают временное перекрытие задач, тогда как сетевая диаграмма не показывает временное перекрытие, но показывает, какие задачи можно выполнять параллельно.
* Некоторые формы диаграмм Ганта могут визуально показывать время провисания, доступное в пределах самого раннего начала и последней продолжительности финиша. Сетевая диаграмма показывает это с помощью данных в прямоугольниках активности.

Менеджеры проектов также используют текстовые отчеты, которые отображают использование ресурсов по задачам, сложность проекта и распределение затрат для управления действиями. Например, на [рисунке 3-19](#bookmark0) показан экран из Microsoft Project для Windows, на котором суммируется вся деятельность проекта, их продолжительность в неделях, а также запланированные даты начала и окончания. Большинство руководителей проектов используют компьютерные системы для разработки своих графических и текстовых отчетов. Далее в этой главе мы обсудим эти автоматизированные системы более подробно.

Руководитель проекта будет периодически проверять состояние всех текущих действий по проекту, чтобы оценить, будут ли эти действия завершены досрочно, вовремя или с опозданием. Если рано или поздно, продолжительность действия, изображенная в столбце 2 [рисунка 3-19,](#bookmark0) может быть обновлена. После изменения запланированное время начала и окончания всех последующих задач также изменится. Внесение такого изменения также изменит диаграмму Ганта или сетевую диаграмму, используемую для представления задач проекта. Возможность легко вносить изменения в проект является очень мощной функцией большинства сред управления проектами. Это позволяет руководителю проекта легко определить, как изменения продолжительности задачи влияют на дату завершения проекта. Это также полезно для изучения влияния сценариев «что, если» добавления или сокращения ресурсов, таких как персонал, для деятельности.





**РИСУНОК 3-18**

Графические диаграммы, на которых изображены планы проектов

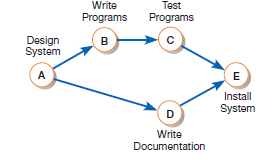
а) Диаграмма Ганта

b) Сетевая диаграмма

(*Источник:* корпорация Майкрософт.)

#### Представление планов проекта

Планирование и управление проектами требуют контроля времени, затрат и ресурсов. **Ресурсы** — это любое лицо, группа людей, часть оборудования или материал, используемые для выполнения деятельности. Сетевая диаграмма — это метод **планирования критического пути**, используемый для управления ресурсами. Критический путь относится к последовательности действий задачи, порядок и продолжительность которых напрямую влияют на дату завершения проекта.



**РИСУНОК 3-20**

Сетевая диаграмма, показывающая действия (представлены кружками) и последовательность этих действий (представлены стрелками)

Сетевая диаграмма является одним из наиболее широко используемых и наиболее известных методов планирования. Вы будете использовать сетевую диаграмму при выполнении задач

* четко определены и имеют четкую начальную и конечную точку,
* можно работать независимо от других задач,
* заказываются, и
* служат цели проекта

Основным преимуществом построения сетевых диаграмм является их способность отражать, как меняется время завершения для действий. Из-за этого он чаще используется, чем диаграммы Ганта, для управления такими проектами, как разработка информационных систем, где вариативность продолжительности деятельности является нормой. Сетевые диаграммы состоят из кругов или прямоугольников, представляющих виды деятельности, и соединительных стрелок, показывающих требуемые рабочие процессы, как показано на [рисунке 3-20.](#bookmark0)

#### Расчет ожидаемой продолжительности времени с помощью PERT

Сетевая диаграмма является одним из наиболее широко используемых и наиболее известных методов планирования. Вы будете использовать сетевую диаграмму при выполнении задач

1. четко определены и имеют четкую начальную и конечную точку,
2. можно работать независимо от других задач,
3. заказываются, и
4. служат цели проекта

Основным преимуществом построения сетевых диаграмм является их способность отражать, как меняется время завершения для действий. Из-за этого он чаще используется, чем диаграммы Ганта, для управления такими проектами, как разработка информационных систем, где вариативность продолжительности деятельности является нормой. Сетевые диаграммы состоят из кругов или прямоугольников, представляющих виды деятельности, и соединительных стрелок, показывающих требуемые рабочие процессы, как показано на [рисунке 3-20.](#bookmark0)

Где

ET=Ожидаемое время завершения действия

- реалистичное время завершения деятельности

- оптимистичное время завершения деятельности

- Пессимистичное время завершения деятельности

Например, предположим, что преподаватель попросил вас рассчитать ожидаемое время выполнения предстоящего задания по программированию. Для этого задания вы оцениваете оптимистичное время в два часа, пессимистичное время в восемь часов и, скорее всего, время в шесть часов. При использовании PERT ожидаемое время выполнения этого задания составляет 5,67 часа. Коммерческое программное обеспечение для управления проектами, такое как Microsoft Project, помогает использовать PERT для расчета ожидаемого времени. Кроме того, многие коммерческие инструменты позволяют настраивать взвешивание оптимистичных, пессимистичных и реалистичных сроков завершения.

#### Построение диаграммы Ганта и сетевой диаграммы в Pine Valley Furniture

Несмотря на то, что PVF исторически была производственной компанией, в последнее время она вышла на рынок прямых продаж для отдельных целевых рынков. Одним из самых быстрорастущих этих рынков является экономичная мебель, подходящая для студентов колледжей. Руководство обратилось с просьбой о разработке новой системы отслеживания деятельности по стимулированию сбыта (СПТС). Этот проект уже успешно прошел через начало проекта и в настоящее время находится на стадии детального планирования проекта, что соответствует фазе SDLC инициации и планирования проекта. SPTS будет использоваться для отслеживания покупок студентов колледжей в следующем осеннем семестре. Студенты обычно покупают недорогие кровати, книжные шкафы, парты, столы, стулья и комоды. Поскольку PVF обычно не имеет большого количества товаров по более низким ценам, руководство считает, что система отслеживания поможет предоставить информацию о рынке студентов колледжей, которая может быть использована для последующих рекламных акций (например, промежуточная распродажа футонов).

Проект заключается в проектировании, разработке и внедрении этой информационной системы до начала осеннего семестра с целью сбора данных о продажах в следующем крупном периоде закупок. Этот срок дает команде проекта 24 недели на разработку и внедрение системы. Совет по системным приоритетам PVF хочет принять решение на этой неделе, исходя из возможности завершения проекта в течение 24-недельного срока. Используя методологию планирования проекта PVF, руководитель проекта Джим Ву знает, что следующим шагом будет построение диаграммы Ганта и сетевой диаграммы проекта для представления базового плана проекта, чтобы он мог использовать эти диаграммы для оценки вероятности завершения проекта в течение 24 недель. Основное действие по планированию проекта сосредоточено на разделении проекта на управляемые действия, оценке времени для каждого из них и последовательности их порядка. Вот шаги, которые Джим выполнил, чтобы сделать это:

* 1. *Определите каждое действие, которое должно быть выполнено в проекте*. После обсуждения нового SPTS с руководством, отделом продаж и разработкой PVF Джим определил следующие основные мероприятия для проекта:

• Сбор требований

• Дизайн экрана

• Дизайн отчета

• Построение баз данных

• Создание пользовательской документации

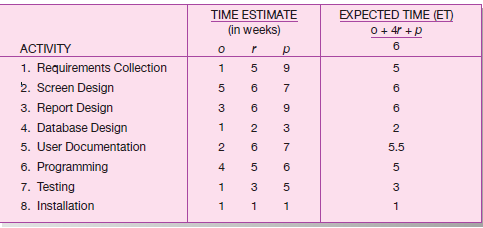
• Программирование программного обеспечения

• Тестирование системы

• Монтаж системы

1. *Определите оценки времени и рассчитайте ожидаемое время завершения для каждого действия*. Определив основные виды деятельности по проекту, Джим установил оптимистичные, реалистичные и пессимистичные оценки времени для каждого мероприятия. Затем эти цифры использовались для расчета ожидаемого времени завершения всех видов деятельности по проекту, как описано ранее с использованием PERT. На рисунке 3-21 показаны расчеты расчетного времени для каждого вида деятельности в рамках проекта СПТС.
2. *Определите последовательность действий и отношения приоритета между всеми действиями, построив диаграмму Ганта и сетевую диаграмму*. Этот шаг поможет вам понять, как связаны различные действия. Джим начинает с определения порядка, в котором должны происходить действия. Результаты этого анализа для проекта SPTS показаны на рисунке 3-22. Первая строка этого рисунка показывает, что сбору требований не предшествуют никакие действия. В строке 2 показано, что дизайну экрана должен предшествовать сбор требований. В строке 4 показано, что построению базы данных должны предшествовать как макет экрана, так и дизайн отчета. Таким образом, действиям может предшествовать ноль, один или несколько действий.

Используя информацию о предполагаемом времени и последовательности действий из рисунков 3-21 и 3-22, Джим теперь может построить диаграмму Ганта и сетевую диаграмму деятельности проекта. Чтобы построить диаграмму Ганта, для каждого действия рисуется горизонтальная полоса, отражающая его последовательность и продолжительность, как показано на рисунке 3-23. Однако диаграмма Ганта может не показывать прямых взаимосвязей между видами деятельности. Например, тот факт, что действие по проектированию базы данных начинается сразу после завершения разработки экрана и панелей конструктора отчетов, не означает, что эти два действия должны быть завершены до начала проектирования базы данных. Чтобы показать такие отношения приоритета, необходимо использовать сетевую диаграмму. Диаграмма Ганта на рисунке 3-23, однако, показывает отношения приоритета.

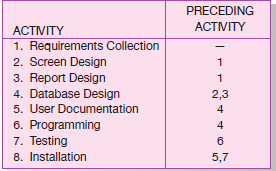


**РИСУНОК 3-21**

Расчет расчетных сроков реализации проекта СПТС

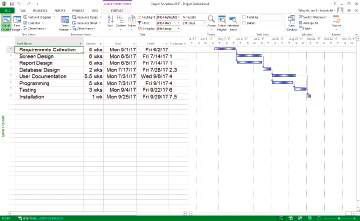
Сетевые диаграммы состоят из двух основных компонентов: стрелок и узлов. Стрелки отражают последовательность действий, тогда как узлы отражают действия, которые отнимают время и ресурсы. Сетевая диаграмма для проекта SPTS показана на рисунке 3-24. Эта диаграмма имеет восемь узлов, помеченных от 1 до 8.

1. *Определите критическую часть*. Критический путь сетевой диаграммы представлен последовательностью связанных действий, которые приводят к наибольшему общему периоду времени. Все узлы и действия в этой последовательности называются находящимися «на» **критическом пути**. Критический путь представляет собой кратчайшее время, в течение которого проект может быть завершен. Другими словами, любая деятельность на критическом пути, которая задерживается в завершении, задерживает весь проект. Однако узлы, не находящиеся на критическом пути, могут быть отложены (на некоторое время) без задержки окончательного завершения проекта. Узлы, не находящиеся на критическом пути, содержат **свободное время** и позволяют руководителю проекта проявлять некоторую гибкость в планировании.



**FIGURE 3-22**

Sequence of activities within the SPTS project



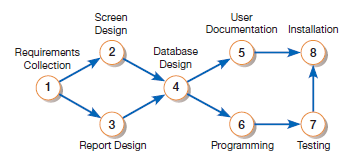
**РИСУНОК 3-23**

Диаграмма Ганта, иллюстрирующая последовательность и продолжительность каждого мероприятия проекта SPTS

На рисунке 3-25 показана сетевая диаграмма, построенная Джимом для определения критического пути и ожидаемые сроки завершения проекта СПТС. Чтобы определить критический путь, Джим рассчитал самое раннее и самое позднее ожидаемое время завершения для каждого действия. Он определил самое раннее ожидаемое время завершения каждого вида деятельности (TE), суммировав расчетное время (ET) для каждого вида деятельности слева направо (т. е. в порядке приоритета), начиная с задания 1 и переходя к упражнению 8. В этом случае TE для активности 8 равен 22 неделям. Если два или более действий предшествуют действию, при расчете ожидаемого времени завершения нового действия используется наибольшее ожидаемое время завершения этих действий. Например, поскольку упражнению 8 предшествуют оба вида деятельности 5 и 7, наибольшее ожидаемое время завершения между 5 и 7 равно 21, поэтому TE для вида деятельности 8 равно 21 + 1 или 22. Самое раннее ожидаемое время завершения последнего действия проекта представляет собой количество времени, которое должно потребоваться для завершения проекта. Однако, поскольку время каждого мероприятия может варьироваться, прогнозируемое время завершения представляет собой только приблизительную оценку. На самом деле проект может потребовать больше или меньше времени для завершения.

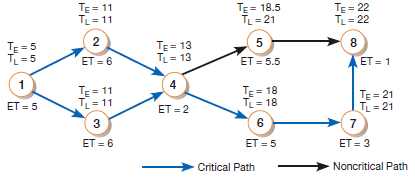
Последнее ожидаемое время завершения (TL) относится к времени, в течение которого деятельность может быть завершена без задержки проекта. Чтобы найти значения для T L каждого актива, Джим начал с упражнения 8 и установил T L, равный конечному TE (22 недели). Затем он работал справа налево к упражнению 1 и вычитал ожидаемое время для каждого действия. Время простоя для каждого действия равно разнице между его последним и самым ранним ожидаемым временем завершения (TL – TE). На рисунке 3-26 показаны расчеты времени задержки для всех видов деятельности в рамках проекта СПТС. Все действия с равным нулю временем провисания находятся на критическом пути. Таким образом, все виды деятельности, кроме деятельности 5, находятся на критическом пути. Часть диаграммы на рисунке 3-25 показывает два критических пути, между действиями 1-2-4 и 1-3-4, потому что оба этих параллельных действия имеют нулевое провисание.

В дополнение к возможности иметь несколько критических путей, на самом деле существует два возможных типа провисания. *Свободный резерв* относится к количеству времени, на которое задача может быть отложена без задержки раннего начала любых сразу следующих задач. *Общий слабину* относится к количеству времени, в течение которого задача может быть отложена без задержки завершения проекта. Понимание свободного и полного провисания позволяет руководителю проекта определить, где можно пойти на компромиссы, если необходимо внести изменения в график проекта. Для получения дополнительной информации о понимании Slack и о том, как его можно использовать для управления задачами, см. *Управление проектами: процесс, технология и практика,* Ганеш Вайдьянатан (2013).



**РИСУНОК 3-24**

Сетевая диаграмма, иллюстрирующая действия (круги) и последовательность (стрелки) этих действий



**РИСУНОК 3-25**

Сетевая диаграмма для проекта СПТС, показывающая расчетное время для каждого вида деятельности, а также самое раннее и самое позднее ожидаемое время завершения для каждого вида деятельности

### Использование программного обеспечения для управления проектами

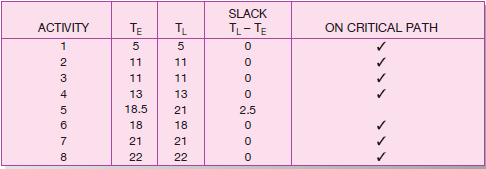
Доступен широкий спектр автоматизированных инструментов управления проектами, которые помогут вам управлять проектом разработки. Новые версии этих инструментов постоянно разрабатываются и выпускаются поставщиками программного обеспечения. Большинство доступных инструментов имеют набор общих функций, которые включают в себя возможность определять и упорядочивать задачи, назначать ресурсы задачам и легко изменять задачи и ресурсы. Инструменты управления проектами доступны для работы на IBM-совместимых персональных компьютерах, Macintosh и более крупных системах на базе мэйнфреймов и рабочих станций. Эти системы различаются по количеству выполняемых задач, сложности взаимосвязей, требованиям к обработке и хранению системы и, конечно же, стоимости. Цены на эти системы могут варьироваться от нескольких сотен долларов для систем на базе персональных компьютеров до более чем 100 000 долларов для крупномасштабных многопроектных систем. Тем не менее, многое можно сделать с такими системами, как Microsoft Project, а также с общедоступными и условно-бесплатными системами. Например, многочисленные условно-бесплатные программы управления проектами (например, OpenProj, Bugzilla и eGroupWare) могут быть загружены из Интернета (например, в [www.download.com](http://www.download.com/)). Поскольку эти системы постоянно меняются, вам следует сравнить магазин, прежде чем выбирать конкретный пакет.

Теперь мы проиллюстрируем типы действий, которые вы будете выполнять при использовании программного обеспечения для управления проектами. Microsoft Project для Windows — это система управления проектами, которая неизменно получает высокие оценки в обзорах компьютерных публикаций (см. [www.microsoft.com](http://www.microsoft.com/) и выполните поиск по слову «проект» — кроме того, если вы выполняете поиск в Интернете, есть много очень полезных руководств для улучшения ваших навыков работы с Microsoft Project). При использовании этой системы для управления проектом необходимо выполнить как минимум следующие действия:

• Установите дату начала или окончания проекта.

• Вводите задачи и назначайте связи задач.

• Выберите метод планирования для просмотра отчетов по проекту.



**РИСУНОК 3-26**

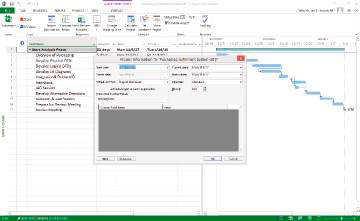
Расчет времени затишья деятельности по проекту СПТС; Все действия, кроме No 5, находятся на критическом пути

#### Определение даты начала проекта

Определение общей информации о проекте включает в себя получение имени проекта и руководителя проекта, а также даты начала или окончания проекта. Даты начала и окончания используются для планирования будущих мероприятий или других мероприятий задним числом (см. ниже) в зависимости от их продолжительности и взаимосвязи с другими мероприятиями. На [рисунке 3-27](#bookmark0) показан пример из Microsoft Project для Windows экрана ввода данных для установления даты начала или окончания проекта. На этом экране показан проект PVF Purchase Fulfillment System. Здесь дата начала проекта - понедельник, 6 ноября 2017 года.

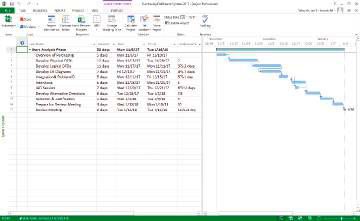
#### Ввод задач и назначение связей между задачами

Следующим шагом в определении проекта является определение задач проекта и их взаимосвязей. Для проекта «Система выполнения закупок» Крис определил 11 задач, которые необходимо выполнить, когда он выполнил начальные действия по системному анализу для проекта (задача 1 — «Фаза начала анализа» — это суммарная задача, которая используется для группировки связанных задач). Экран ввода задачи, показанный на рисунке 3-28, похож на финансовую программу для работы с электронными таблицами. Пользователь перемещает курсор в ячейку с помощью клавиш со стрелками или мыши, а затем просто вводит текстовое имя и числовую продолжительность для каждого действия. Запланированное начало и запланированное окончание вводятся автоматически в зависимости от даты начала и продолжительности проекта. Чтобы задать связь действия, в столбец Предшественники вводится идентификационный номер (или номера) действия, которое должно быть завершено до начала текущего действия. Дополнительные коды в этом столбце делают отношения приоритета более точными. Например, рассмотрим столбец «Предшественник» для идентификатора 6. Запись в этой ячейке говорит о том, что упражнение 6 не может начаться раньше, чем за один день до окончания действия 5. (Microsoft Project предоставляет множество различных вариантов приоритета и задержек, как показано в этом примере, но их обсуждение выходит за рамки нашего рассмотрения.) Программное обеспечение для управления проектами использует эту информацию для построения диаграмм Ганта, сетевых диаграмм и других отчетов, связанных с проектом.



**РИСУНОК 3-27**

Установление даты начала проекта в Microsoft Project для Windows \(*Источник:* корпорация Майкрософт.)



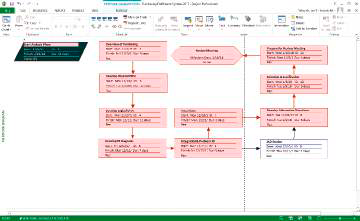
**РИСУНОК 3-28**

Ввод задач и назначение связей между задачами в проекте Майкрософт для Windows (*Источник:* корпорация Майкрософт)

#### Выбор метода планирования для просмотра отчетов по проектам

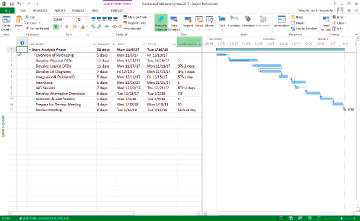
После того, как информация обо всех мероприятиях по проекту введена, очень легко просмотреть информацию в различных графических и текстовых форматах с помощью дисплеев или печатных отчетов. Например, на рисунке 3-28 показана информация о проекте на экране диаграммы Ганта, тогда как на рисунке 3-29 показана информация о проекте в виде сетевой диаграммы. Вы можете легко изменить способ просмотра информации, выбрав ее в меню Вид, показанном на рисунке 3-29.

Как упоминалось ранее, в промежуточных отчетах по проектам для руководства часто сравнивается фактический прогресс с планами. На рисунке 3-30 показано, как Microsoft Project отображает ход выполнения сплошной линией на панели действий. На этом рисунке задача 2 выполнена, а задача 3 почти завершена (на 75 процентов), но остается небольшой процент работы, о чем свидетельствуют незавершенные сплошные линии внутри полосы для этой задачи. Предполагая, что этот экран представляет статус проекта по состоянию на четверг, 11 ноября 2016 года, третье мероприятие выполняется примерно по графику, но второе мероприятие отстает от ожидаемой даты завершения. Табличные отчеты могут обобщать одну и ту же информацию.



**РИСУНОК 3-29**

Просмотр сведений о проекте в виде сетевой диаграммы в Microsoft Project для Windows



**РИСУНОК 3-30**

Диаграмма Ганта, показывающая ход выполнения мероприятий (правый кадр) по сравнению с запланированными мероприятиями (левый кадр)

Это краткое введение в программное обеспечение для управления проектами только поцарапало поверхность, чтобы показать вам мощь и особенности этих систем. Другие функции, которые широко доступны и особенно полезны для проектов с участием нескольких человек, связаны с использованием и использованием ресурсов. Функции, связанные с ресурсами, позволяют определять такие характеристики, как стандартные ставки затрат и ежедневная доступность, с помощью календаря, в котором записываются праздничные дни, рабочее время и отпуска. Эти функции особенно полезны для выставления счетов и оценки затрат по проекту. Часто ресурсы распределяются между несколькими проектами, что может существенно повлиять на график проекта.

В зависимости от того, как выставляются счета за проекты в организации, назначение и выставление счетов за ресурсы для задач может быть очень трудоемким занятием для большинства менеджеров проектов. Функции, предоставляемые этими мощными инструментами, могут значительно облегчить планирование и управление проектами, чтобы эффективно использовать проектные и управленческие ресурсы.

### Резюме

Основное внимание в этой главе было уделено управлению проектами информационных систем и роли руководителя проекта в этом процессе. Менеджер проекта обладает как техническими, так и управленческими навыками и в конечном итоге несет ответственность за определение размера, объема и требований к ресурсам для проекта. После того, как проект считается осуществимым организацией, менеджер проекта гарантирует, что проект соответствует потребностям клиента и выполняется в рамках бюджета и временных ограничений. Чтобы управлять проектом, руководитель проекта должен выполнить четыре основных действия: инициацию проекта, планирование проекта, выполнение проекта и закрытие проекта. Основное внимание при инициировании проекта уделяется оценке размера, охвата и сложности проекта, а также установлению процедур для поддержки последующей деятельности по проекту. Основное внимание при планировании проекта уделяется определению четких, дискретных мероприятий и работы, необходимой для выполнения каждого мероприятия. Основное внимание при осуществлении проекта уделяется претворению в жизнь планов, разработанных в ходе инициирования и планирования проекта. Проект закрыт фокусируется на том, чтобы довести проект до конца.

Диаграммы Ганта и сетевые диаграммы являются мощными графическими методами, используемыми при планировании и контроле проектов. Как диаграммы Ганта, так и методы планирования сетевых диаграмм требуют, чтобы в проекте были действия, которые могут быть определены как имеющие четкое начало и конец, могут работать независимо от других действий, упорядочены и таковы, что их завершение означает конец проекта. Диаграммы Ганта используют горизонтальные полосы для представления начала, продолжительности и окончания действия. Сетевая диаграмма — это метод планирования критического пути, который показывает взаимосвязи между действиями. Планирование критического пути относится к методам планирования, при которых порядок и продолжительность деятельности проекта напрямую влияют на дату завершения проекта. Эти диаграммы показывают, когда действия могут начинаться и заканчиваться, какие действия не могут быть отложены без задержки всего проекта, сколько времени у каждого действия и прогресс по сравнению с запланированными действиями. Способность сетевой диаграммы использовать оценки вероятности при определении критических путей и сроков делает ее широко используемым методом для очень сложных проектов.

Доступен широкий спектр автоматизированных инструментов для помощи менеджеру проекта. Большинство инструментов имеют набор общих функций, включая возможность определять и упорядочивать задачи, назначать ресурсы задачам и изменять задачи и ресурсы. Системы различаются по количеству поддерживаемых действий, сложности взаимосвязей, требованиям к обработке и хранению, а также стоимости.

## Приложение: Объектно-ориентированный анализ и проектирование

### Уникальные характеристики проекта OOSAD

В этой главе мы описали, как проекты являются человеком при использовании структурированного подхода к разработке. Эти концепции и методы очень устойчивы к широкому спектру проектов и подходов к разработке. Однако при разработке системы с использованием более итеративного подхода к проектированию, такого как прототипирование или объектно-ориентированный анализ и проектирование, необходимо учитывать некоторые дополнительные вопросы. В этом разделе мы обсудим некоторые уникальные характеристики этих типов проектов (см. Fuller et al., 2008; George et al., 2007).

Когда система разрабатывается с использованием итеративного подхода, это означает, что в течение всего проекта часть конечной системы строится на каждом этапе итерации. Таким образом, система развивается постепенно, так что к последней итерации проекта вся система будет завершена (см. рис. 3-31). Для того, чтобы система развивалась таким образом, руководитель проекта должен понимать несколько уникальных характеристик проекта OOSAD.

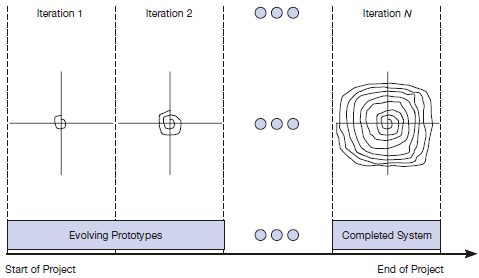
### Определите систему как набор компонентов

Чтобы управлять проектом как серией итераций, менеджер проекта должен разделить общую систему на набор компонентов; при объединении этого набора получится вся система (см. рис. 3-32). Каждый из этих отдельных компонентов системы часто называют «вертикальным срезом» всей системы; Это ключевая особенность системы, которую можно продемонстрировать пользователям. Кроме того, каждый фрагмент не должен быть подсистемой, охватывающей «горизонтально» всю систему, поскольку эти горизонтальные фрагменты обычно не фокусируются на конкретной системной функции и обычно не подходят для демонстрации пользователям. По сути, каждый вертикальный срез представляет собой сценарий использования системы (см. главу 7 для получения дополнительной информации о диаграммах вариантов использования). Кроме того, обратите внимание на рисунке 3-32, что управление и планирование проекта — это деятельность, которая продолжается на протяжении всего жизненного цикла проекта.

Одним из результатов определения всей системы как набора компонентов является вероятность того, что компоненты, построенные ранее в проекте, потребуют большей доработки, чем те, которые были разработаны позже в проекте. Например, на ранних стадиях проекта отсутствующие компоненты или непонимание ключевых архитектурных особенностей потребуют, чтобы компоненты, разработанные на ранних этапах проекта, были существенно изменены по мере продвижения проекта, чтобы успешно интегрировать эти компоненты в единую всеобъемлющую систему. Это означает, что доработка является естественной частью проекта OOSAD и что не следует чрезмерно беспокоиться, когда это происходит. Это просто характеристика итеративного и инкрементального процесса разработки OOSAD.

#### Сначала решайте сложные задачи

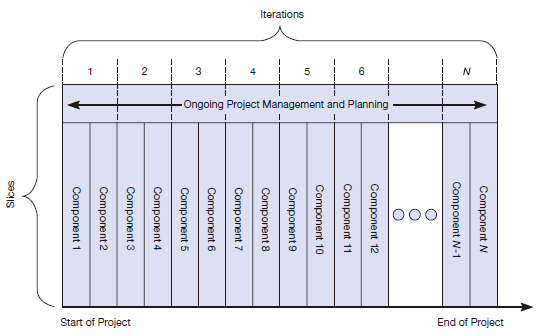
Еще одна характерная черта подхода OOSAD заключается в том, что он в первую очередь решает сложные проблемы. При разработке классических структурированных систем сложная проблема, такая как выбор физической среды реализации, решается на поздних этапах процесса разработки. В результате, следование классическому подходу к разработке систем, как правило, приводит к тому, что принятие некоторых ключевых архитектурных решений по системам откладывается до конца проекта. Такой подход иногда проблематичен, потому что такие решения часто определяют, является ли проект успешным или неудачным. С другой стороны, скорейшее решение сложных проблем позволяет изучить их до того, как будут израсходованы значительные ресурсы. Это снижает риск проекта.



**РИСУНОК 3-31**

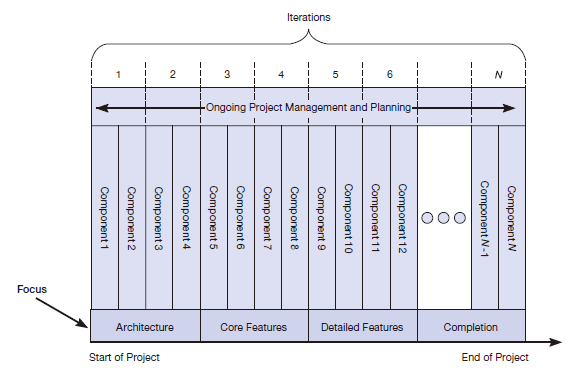
В процессе OOSAD система постепенно развивается в течение всего срока реализации проекта

Кроме того, скорейшее решение более сложных задач, связанных с архитектурой систем, помогает завершить все последующие компоненты, поскольку большинство из них будут основываться на этих базовых архитектурных возможностях. (В некоторых проектах самые сложные компоненты зависят от более простых компонентов. В этих случаях нужно сначала выполнить более простые ломтики, прежде чем переходить к более сложным. Тем не менее, следует как можно скорее сосредоточиться на сложных проблемах.) С точки зрения планирования проекта это означает, что на протяжении всего жизненного цикла проекта происходит естественное развитие и упорядочение компонентов. Первоначальная итерация или две должны быть сосредоточены на системной архитектуре, такой как база данных или сетевая инфраструктура. После завершения работы над архитектурой реализуются основные возможности системы, такие как создание и удаление записей. После завершения работы над основными компонентами системы реализуются подробные характеристики системы, которые помогают точно настроить ключевые возможности системы. На заключительных этапах итерации основное внимание уделяется действиям, которые завершают проект (например, доработка интерфейса, руководства пользователя и обучение; см. рис. 3-33).



**РИСУНОК 3-32**

Объектно-ориентированные проекты разработки разрабатываются с использованием постоянного управления и развивающейся функциональности системы



**РИСУНОК 3-33**

Фокус и порядок компонентов системы меняются в течение жизненного цикла проекта

#### Использование итераций для управления проектом

Во время каждой итерации проекта формируются все действия жизненного цикла разработки систем (см. рис. 3-34). Это означает, что каждая итерация проекта включает в себя управление и планирование, анализ, проектирование, внедрение и эксплуатацию. Для каждой итерации входными данными для процесса являются выделенные компоненты проекта — вертикальные срезы или варианты использования, которые должны быть выполнены во время этой итерации, и результаты предыдущей итерации. Результаты этой итерации затем используются в качестве входных данных для следующей итерации. Например, по мере разработки и реализации компонентов многое узнают о том, как необходимо будет реализовать последующие компоненты. Обучение, происходящее во время каждой итерации, помогает руководителю проекта лучше понять, как будут проектироваться последующие компоненты, какие проблемы могут возникнуть, какие ресурсы необходимы и сколько времени и сложности компонента потребуется для завершения. В результате, большинство опытных менеджеров проектов считают, что было бы ошибкой делать планы проекта слишком подробными в начале проекта, когда многое еще неизвестно.

#### Не планируйте слишком много заранее

Во время каждой итерации все больше и больше будет узнавать о том, как необходимо будет проектировать последующие компоненты, сколько времени может потребоваться для завершения каждого из них и так далее. Поэтому было бы ошибкой строить очень подробные планы на далекое будущее, потому что вполне вероятно, что эти планы окажутся ошибочными. В OOSAD по мере завершения каждой итерации цель состоит в том, чтобы узнать больше о создаваемой системе, возможностях команды разработчиков, сложности среды разработки и так далее. По мере того, как это понимание приобретается в ходе проекта, менеджер проекта может делать все более и более точные прогнозы и планы. В результате составление подробных планов для всех итераций проекта, скорее всего, приведет к большой трате времени. Руководитель проекта должен быть озабочен только составлением подробных планов на следующую итерацию или две. По мере того, как руководитель проекта будет учиться в ходе проекта, он или она сможет постоянно уточнять графики, оценки времени и потребности в ресурсах с помощью все более и более точных оценок (см. рис. 3-35).

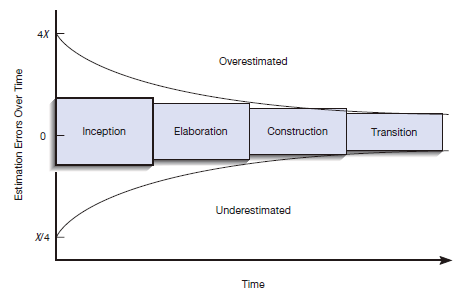


**РИСУНОК 3-34**

Рабочий процесс итерации (*Источник:* на основе Royce, 1998; George et al., 2007.)

#### Сколько и как долго длятся итераций?

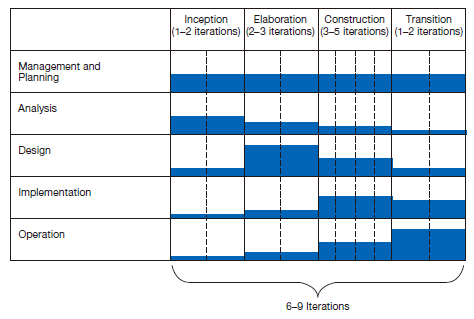
Один из вопросов, который возникает у многих людей при первом знакомстве с OOSAD, связан с количеством и продолжительностью итераций. Итерации рассчитаны на фиксированный промежуток времени, обычно от двух до восьми недель, но они могут быть короткими до одной недели (особенно для небольших проектов). В течение одной итерации может быть завершено несколько компонентов (вариантов использования). Однако важно не пытаться упаковать разработку слишком большого количества компонентов в одну итерацию. Опыт показывает, что наличие большего количества итераций с меньшим количеством компонентов, которые необходимо завершить, лучше, чем иметь только несколько итераций с большим количеством компонентов, которые необходимо завершить.



**РИСУНОК 3-35**

Оценка планирования со временем улучшается (*источник:* На основе Royce, 1998; George et al., 2007.)

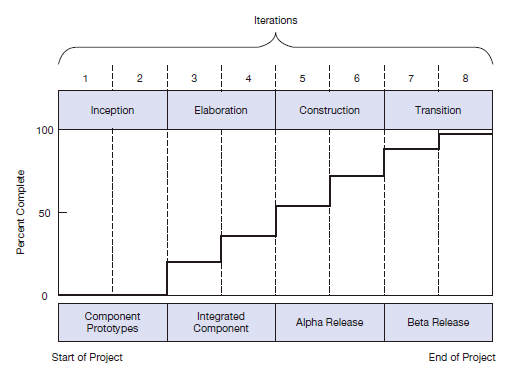
Только путем итерации — завершения полного цикла разработки систем — можно получить значительное обучение, которое поможет руководителю проекта лучше планировать последующие итерации.



**РИСУНОК 3-36**

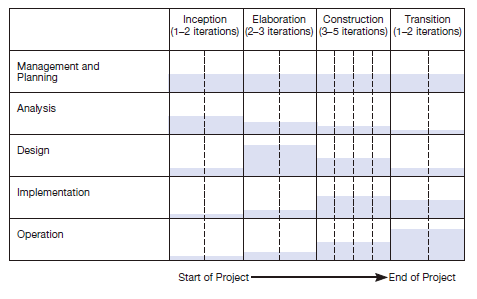
Проект OOSAD обычно имеет от шести до девяти итераций

Начальная фаза обычно влечет за собой одну итерацию, но нередко для этого требуется две или более итераций в больших, сложных проектах. Точно так же разработка часто завершается за одну или две итерации, но опять же на это могут повлиять сложность и размер системы. Построение может варьироваться от двух до нескольких итераций, и переход обычно происходит в течение одной или двух итераций. Таким образом, опытные менеджеры проектов OOSAD обычно используют от шести до девяти итераций при проектировании и построении системы (см. рис. 3-36). Обратите внимание, что все завершенные компоненты интегрируются в комплексную систему в конце каждой итерации. Во время первой итерации, скорее всего, будут созданы простые прототипы компонентов, такие как открытие, закрытие и сохранение файлов. Однако по мере продвижения проекта прототипы становятся все более сложными, пока вся система не будет завершена (см. рис. 3-37).



**РИСУНОК 3-37** 0

По мере развития проекта развивается и функциональность системы



**РИСУНОК 3-38**

Уровень и направленность деятельности в процессе разработки систем меняются от начала до конца проекта

### Резюме

При управлении проектом OOSAD руководитель проекта должен определить проект как набор компонентов. После определения эти компоненты могут быть проанализированы и упорядочены таким образом, чтобы наиболее сложные компоненты были реализованы в первую очередь. Проект OOSAD управляется серией итераций, каждая итерация содержит все фазы цикла разработки систем. На каждой итерации создается все больше и больше системы (компонент за компонентом), и все больше и больше узнают о строящейся системе, возможностях команды разработчиков и сложности среды разработки. По мере того, как это обучение увеличивается с течением времени, менеджер проекта может более точно планировать проектную деятельность. Поэтому не рекомендуется планировать долгосрочные мероприятия в мельчайших подробностях; Детальное планирование должно выполняться только для текущей и последующей итерации. Большинство проектов имеют от шести до девяти итераций, но крупные проекты могут иметь еще несколько. Итерация — это фиксированный период времени, обычно около двух недель, но он может быть короче или длиннее в зависимости от характеристик проекта.

# Часть 2 Планирование

## Обзор

Спрос на новые или заменяющие системы превышает возможности и ресурсы большинства организаций для осуществления проектов по разработке систем либо самостоятельно, либо с консультантами. Это означает, что организации должны установить приоритеты и направление развития систем, которые принесут проекты развития с наибольшими чистыми выгодами. Как системный аналитик, вы должны проанализировать требования к пользовательской информации, а также помочь сделать экономическое обоснование или обосновать, почему система должна быть построена и проект разработки проведен.

Причина создания любой новой или усовершенствованной информационной системы (ИС) заключается в том, чтобы повысить ценность организации. Как системные аналитики, мы должны использовать ресурсы разработки систем для создания сочетания систем, которые приносят наибольшую пользу организации. Как мы можем определить ценность систем для бизнеса и определить те приложения, которые обеспечивают наиболее важные выгоды? Вторая часть посвящена этой теме, первой фазе жизненного цикла разработки систем (SDLC), которую мы называем планированием. Ценность бизнеса заключается в поддержке наиболее важных бизнес-целей и оказании помощи организации в реализации ее бизнес-стратегии. Все системы, независимо от того, поддерживают ли они операционные или стратегические функции, должны быть связаны с бизнес-целями. В двух главах этой части книги показано, как установить эту связь.

Источником системных проектов являются либо инициативы по планированию ИС (упреждающая идентификация систем), либо запросы пользователей или специалистов по ИС (реакция на проблемы или возможности) на новые или усовершенствованные системы. В главе 4 мы описываем связи между корпоративным планированием, планированием ИБ, а также определением и отбором проектов. Мы не включаем планирование ИБ в состав SDLC, но результаты планирования ИС сильно влияют на рождение и проведение системных проектов. В главе 4 приводится убедительный аргумент в пользу того, что планирование ИБ дает не только представление о том, какие системы нужны организации, но и описывает стратегии, необходимые для оценки жизнеспособности любого потенциального системного проекта.

Более частым источником идентификации проекта являются запросы на системное обслуживание (SSR) от бизнес-менеджеров и специалистов по ИБ, как правило, для очень сфокусированных систем или постепенных улучшений в существующих системах. Бизнес-менеджеры запрашивают новую или замену системы, когда они считают, что улучшенные информационные услуги помогут им выполнять свою работу. Специалисты по ИБ могут запрашивать обновления системы, когда технологические изменения делают текущие реализации системы устаревшими или когда производительность существующей системы нуждается в улучшении. В любом случае запрос на обслуживание должен быть понят руководством, и должно быть разработано обоснование системы и связанного с ней проекта.

Мы продолжаем рассмотрение дела Petrie Electronics после главы 4. В данном случае мы показываем, как идея нового проекта ИБ была стимулирована синергией между корпоративным стратегическим планированием и креативностью индивидуального бизнес-менеджера.

Глава 5 посвящена тому, что происходит после того, как проект был идентифицирован и выбран: следующий шаг в создании экономического обоснования, инициирование и планирование предлагаемого системного запроса. Этот план способствует лучшему пониманию масштабов потенциального изменения системы и характера необходимых системных функций. На основе этого предварительного понимания системных требований разрабатывается план проекта, в котором показаны как подробные шаги и ресурсы, необходимые для проведения фазы анализа жизненного цикла, так и более общие шаги для последующих этапов. Также описываются осуществимость и потенциальные риски запрашиваемой системы, а также проводится экономический анализ затрат и выгод, чтобы показать потенциальное влияние изменения системы. В дополнение к экономической целесообразности или обоснованности системы оцениваются технические, организационные, политические, правовые, плановые и другие возможности. Выявляются потенциальные риски — нежелательные результаты — и разрабатываются планы по устранению этих возможностей. Инициация и планирование проекта заканчиваются, когда официальное предложение по проекту разработки систем завершено и представлено на утверждение лицу, которое должно выделить ресурсы на разработку систем. В случае одобрения проект переходит в фазу анализа SDLC.

Мы проиллюстрируем типичную фазу инициации и планирования проекта в случае с Petrie Electronics после главы 5. В этом случае мы покажем, как компания разработала техническое задание проекта и рассмотрела различные аспекты на этапе инициации и планирования проекта.

## *Глава 4 Идентификация и отбор проектов разработки систем*

### *Знакомство*

Сфера применения информационных систем сегодня – это целое предприятие. Руководители, работники умственного труда и все другие члены организации рассчитывают на легкий доступ к информации и ее извлечение независимо от ее местонахождения. Неинтегрированные системы, использовавшиеся в прошлом, часто называемые «островами информации», заменяются совместными, интегрированными корпоративными системами, которые могут легко поддерживать обмен информацией. Хотя для достижения цели наведения мостов между этими «островами» потребуется некоторое время, она представляет собой четкое направление развития информационных систем. Использование систем планирования общеорганизационных ресурсов (ERP) таких компаний, как SAP (www.sap.com) и Oracle (www.oracle.com), позволило связать эти «острова» во многих организациях. Кроме того, по мере того, как использование Интернета продолжает развиваться для поддержки деловой активности, системная интеграция стала первостепенной задачей организаций (Fox, 2013; Люфтман, 2004; «Оверби», 2006; «Вайлгум», 2010; Westerman et al., 2014; Weill and Ross, 2009).

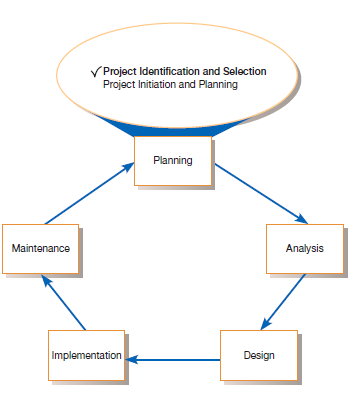
Получение интегрированных вычислений в масштабах всего предприятия представляет собой серьезную проблему как для управления корпоративными, так и для информационных систем. Например, учитывая распространение персональных и ведомственных вычислений, в которых были созданы разрозненные системы и базы данных, как организация может контролировать и поддерживать все эти системы и данные? Во многих случаях они просто не могут; практически невозможно отследить, у кого какие системы и какие данные, где есть совпадения или несоответствия, а также точность информации. Причина, по которой существует множество персональных и ведомственных систем и баз данных, заключается в том, что пользователи либо не знают об информации, которая существует в корпоративных базах данных, либо не могут легко получить к ней доступ, поэтому они создают и поддерживают свою собственную информацию и системы. Интеллектуальная идентификация и выбор системных проектов, как для новых, так и для заменяющих систем, является важным шагом в получении контроля над системами и данными. Многие директора по информационным технологиям (ИТ-директора) надеются, что с появлением ERP-систем, улучшенной системной интеграцией и быстрым развертыванием корпоративных интернет-решений эти острова будут сокращены или ликвидированы (Fox, 2013; Harvard Business Review, 2011; Люфтман, 2004; Ньюболд и Азуа, 2007; Олавсруд, 2014; «Вейл и Росс», 2009; Wailgum, 2010)

Приобретение, разработка и обслуживание информационных систем потребляют значительные ресурсы для большинства организаций. Это говорит о том, что организациям может быть полезно следовать формальному процессу определения и отбора проектов. Этому вопросу посвящен первый этап жизненного цикла разработки систем — идентификация и выбор проекта. В следующем разделе вы узнаете об общем методе определения и отбора проектов, а также о результатах и результатах этого процесса. Далее следуют краткие описания корпоративного стратегического планирования и планирования информационных систем, двух видов деятельности, которые могут значительно улучшить процесс идентификации и отбора проектов.

### *Идентификация и отбор проектов по разработке систем*

Первым этапом SDLC является планирование, состоящее из идентификации и выделения проекта, а также инициирования и планирования проекта (см[. рис. 4-1).](#bookmark0) Во время идентификации и отбора проектов старший менеджер, бизнес-группа, менеджер ИБ или руководящий комитет определяют и оценивают все возможные проекты по разработке систем, которые может предпринять организационная единица. Затем те проекты, которые, как считается, с наибольшей вероятностью принесут значительные организационные выгоды с учетом имеющихся ресурсов, отбираются для последующей деятельности в области развития. Организации различаются по своему подходу к выявлению и отбору проектов. В некоторых организациях идентификация и отбор проектов являются весьма формальным процессом, в котором проекты являются результатом более широкого общего процесса планирования. Например, крупная организация может следовать формальному процессу идентификации проекта, в соответствии с которым предлагаемый проект строго сравнивается со всеми конкурирующими проектами. В качестве альтернативы небольшая организация может использовать неформальные процедуры отбора проектов, которые позволяют менеджеру ИС самого высокого ранга самостоятельно выбирать проекты или позволяют отдельным бизнес-единицам принимать решения по проектам после согласия на предоставление финансирования проекта.

Запросы на разработку информационных систем поступают из различных источников. Одним из источников являются просьбы руководителей и бизнес-подразделений о замене или расширении существующей системы для получения необходимой информации или предоставления новых услуг операторам. Другим источником запросов являются менеджеры ИБ, которые хотят сделать систему более эффективной и менее затратной в эксплуатации или хотят перенести ее в новую операционную среду. Конечным источником проектов является формальная группа планирования, которая определяет проекты для улучшения, чтобы помочь организации достичь своих корпоративных целей (например, новая система для обеспечения лучшего обслуживания клиентов). Независимо от того, как данная организация фактически выполняет процесс идентификации и отбора проектов, существует общая последовательность действий. В следующих разделах мы описываем общий процесс определения и отбора проектов и получения результатов и результатов этого процесса.



**Рисунок 4-1**

Жизненный цикл разработки систем с выделением идентификации и выбора проекта

#### Процесс выявления и отбора проектов развития ИБ

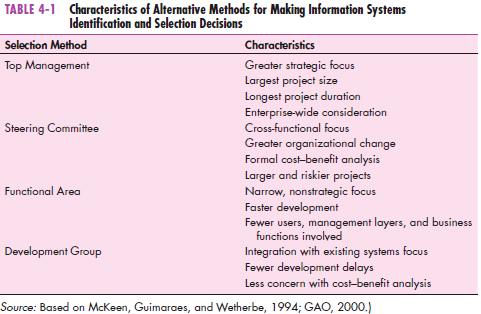
Идентификация и отбор проекта состоит из трех основных видов деятельности:

1. Выявление потенциальных проектов развития
2. Классификация и ранжирование проектов развития ИБ
3. Выбор проектов развития ИБ

Каждый из этих шагов описан ниже:

* + 1. *Выявление потенциальных проектов развития.* Организации различаются в отношении того, как они идентифицируют проекты. Этот процесс может быть выполнен с помощью
       - ключевой член высшего руководства, либо генеральный директор малой или средней организации, либо старший руководитель более крупной организации;
       - руководящий комитет, состоящий из широкого круга руководителей, интересующихся системами;
       - пользовательские отделы, в которых либо руководитель запрашивающего подразделения, либо комитет из запрашивающего отдела решает, какие проекты подавать (часто вы, как системный аналитик, будете помогать пользователям готовить такие запросы); или
       - группа разработки или старший менеджер по ИБ.

Было обнаружено, что все методы идентификации имеют сильные и слабые стороны. Исследования показали, например, что проекты, определяемые высшим руководством, чаще имеют стратегическую организационную направленность. С другой стороны, проекты, определяемые руководящими комитетами, чаще отражают разнообразие обязательств и, следовательно, имеют кросс-функциональную направленность. Проекты, определяемые отдельными отделами или бизнес-единицами, чаще всего имеют узкую, тактическую направленность. Наконец, доминирующей характеристикой проектов, определенных группой разработчиков, является легкость, с которой существующее оборудование и системы будут интегрироваться с предлагаемым проектом. Другие факторы, такие как стоимость проекта, продолжительность, сложность и риск, также зависят от источника данного проекта. Характеристики каждого метода отбора кратко изложены в [таблице 4-1.](#bookmark0)  В дополнение к тому, кто принимает решение, характеристики, характерные для организации, такие как уровень диверсификации фирмы, уровень вертикальной интеграции или степень возможностей роста, также могут влиять на любое решение об инвестициях или выборе проекта (Dewan et al., 1998; «Лиса», 2013; Harvard Business Review, 2009; Люфтман, 2004; Yoo et al., 2006; Томас и Фернандес, 2008; Weill and Ross, 2009).



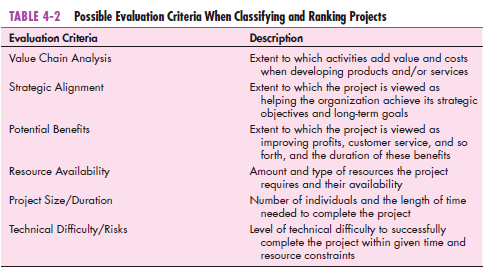
Из всех возможных источников проекта те, которые определены высшим руководством и руководящими комитетами, чаще всего отражают более широкие потребности организации. Это связано с тем, что высшее руководство и руководящие комитеты, скорее всего, будут иметь более широкое представление об общих бизнес-целях и ограничениях. Поэтому проекты, определенные высшим руководством или различным руководящим комитетом, называются исходящими из нисходящего источника.

Проекты, определяемые функциональным менеджером, бизнес-подразделением или группой разработки информационных систем, часто разрабатываются для конкретных бизнес-потребностей в рамках данного бизнес-подразделения. Другими словами, эти проекты могут не отражать общие цели организации. Это не означает, что проекты, определяемые отдельными руководителями, бизнес-подразделениями или группой разработки ИБ, являются недостаточными, а только то, что они могут не учитывать более широкие организационные вопросы. Проектные инициативы, исходящие от менеджеров, бизнес-подразделений или группы разработчиков, обычно называются исходящими из восходящего источника. Это типы проектов, в которых вы, как системный аналитик, будете играть самую раннюю роль в жизненном цикле в рамках вашей постоянной поддержки пользователей. Вы поможете менеджерам пользователей описать информационные потребности и причины выполнения проекта, которые будут оцениваться при выборе среди всех представленных проектов, какие из них будут одобрены для перехода к этапу инициирования и планирования проекта SDLC.

Таким образом, проекты определяются как по принципу «сверху вниз», так и по принципу «снизу вверх». Формальность процесса определения и отбора проектов может существенно различаться в разных организациях. Кроме того, поскольку ограниченные ресурсы не позволяют разработать все предлагаемые системы, большинство организаций имеют процесс классификации и ранжирования достоинств каждого проекта. Таким образом, те проекты, которые будут сочтены несовместимыми с общими целями Организации, избыточными по функциональности для какой-либо существующей системы или ненужными, будут сняты с рассмотрения. Эта тема обсуждается далее.

* + 1. *Классификация и ранжирование проектов развития ИБ.* Вторым важным направлением деятельности в процессе идентификации и отбора проектов является оценка относительных достоинств потенциальных проектов. Как и в случае с процессом идентификации проектов, классификация и ранжирование проектов могут выполняться топ-менеджерами, руководящим комитетом, бизнес-подразделениями или группой разработки ИБ. Кроме того, критерии, используемые при присвоении относительных достоинств данного проекта, могут различаться. Часто используемые критерии оценки проектов кратко изложены в таблице 4-2. В любой организации в процессе классификации и ранжирования может использоваться один или несколько критериев.

Как и в случае с процессом идентификации и отбора проектов, фактические критерии, используемые для оценки проектов, будут варьироваться в зависимости от организации. Если, например, организация использует руководящий комитет, она может проводить ежемесячные или ежеквартальные заседания для рассмотрения проектов и использовать широкий спектр критериев оценки. На этих совещаниях будут рассматриваться новые запросы на проекты в связи с уже определенными проектами, а также осуществляться мониторинг текущих проектов. Относительные рейтинги проектов используются для руководства конечным действием этого процесса идентификации — выбором проекта.



Важным методом оценки проектов, который широко используется для оценки проектов разработки информационных систем, является **анализ цепочки создания стоимости** (Foss and Saebi, 2015; «Портер», 1985; Ван ден Берг и Питерсма, 2015). Анализ цепочки создания стоимости — это процесс анализа деятельности организации по производству продуктов и/или услуг для определения того, где добавлена стоимость и понесены затраты. Как только организация получает четкое представление о своей цепочке создания стоимости, могут быть достигнуты улучшения в операциях и производительности организации. Информационные системы, обеспечивающие наибольшую выгоду для производственно-сбытовой цепочки, будут иметь приоритет над проектами с меньшими выгодами.

Как вы уже догадались, информационные системы стали одним из основных способов для организаций вносить изменения и улучшения в свои цепочки создания стоимости. Многие организации, например, используют Интернет для обмена важной деловой информацией с поставщиками и клиентами, такой как заказы, счета-фактуры и квитанции. Чтобы провести анализ цепочки создания стоимости для организации, думайте об организации как о большом процессе ввода-вывода (см. [рис. 4-2).](#bookmark0) На одном конце находятся входные данные организации, например, закупаемые материалы. Внутри организаций эти поставки и ресурсы каким-то образом интегрированы для производства продуктов и услуг. На другом конце находятся результаты, которые представляют продукты и услуги, которые продаются, продаются, а затем распределяются среди клиентов. При анализе производственно-сбытовой цепочки необходимо сначала понять каждую деятельность, функцию и процесс, в которых добавлена или должна быть добавлена стоимость. Затем определите затраты (и факторы, которые влияют на затраты или вызывают их колебания) в каждой из областей. Разобравшись в своей цепочке создания стоимости и затратах, вы можете сравнить (сравнить) свою производственно-сбытовую цепочку и связанные с ней расходы с цепочками других организаций, предпочтительно ваших конкурентов. Проводя эти сравнения, можно определить приоритеты применения проектов информационных систем.

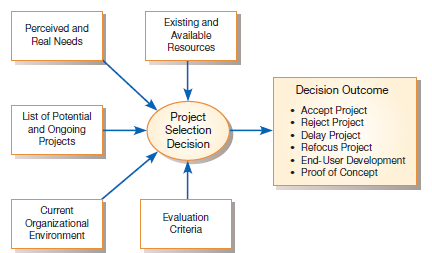
* + 1. *Выбор проектов разработки ИБ*. Заключительным действием в процессе идентификации и отбора проектов является фактический отбор проектов для дальнейшего развития. Отбор проектов — это процесс рассмотрения как краткосрочных, так и долгосрочных проектов и выбора тех, которые с наибольшей вероятностью достигнут бизнес-целей. Кроме того, по мере того, как условия ведения бизнеса меняются с течением времени, относительная важность любого отдельного проекта может существенно измениться. Таким образом, идентификация и отбор проектов является очень важной и постоянной деятельностью.

При принятии решений о выборе проекта необходимо учитывать множество факторов. На рисунке 4-3 показано, что решение об отборе требует, чтобы предполагаемые потребности организации, существующие системы и текущие проекты, наличие ресурсов, критерии оценки, текущие условия ведения бизнеса и точки зрения лиц, принимающих решения, играли роль в решениях по выбору проектов. Этот процесс принятия решений может привести к многочисленным результатам. Конечно, проекты могут быть приняты или отклонены. Принятие проекта обычно означает, что финансирование для проведения следующего этапа SDLC было одобрено. Отказ означает, что проект больше не будет рассматриваться для разработки. Однако проекты также могут быть условно приняты; они могут быть приняты до утверждения или наличия необходимых ресурсов или демонстрации того, что особенно сложный аспект системы может быть разработан. Проекты также могут быть возвращены первоначальным заявителям, которым предлагается разработать или приобрести запрошенную систему. Наконец, заявителей проекта может быть предложено изменить и повторно подать свой запрос после внесения предлагаемых изменений или уточнений.



**Рисунок 4-2**

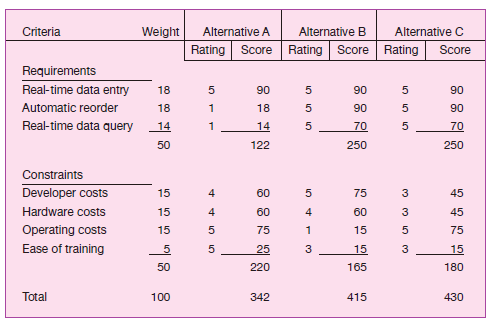
Организации можно рассматривать как цепочку создания стоимости, превращающую сырье в продукты для клиентов



**Рисунок 4-3**

Решения по выбору проекта должны учитывать множество факторов и могут иметь множество результатов

Один из методов выбора между различными проектами или при рассмотрении альтернативных конструкций для данной системы проиллюстрирован на [рисунке 4-4.](#bookmark0)  Например, предположим, что для данной системы, которая была идентифицирована и выбрана, есть три альтернативных проекта, которые могут быть реализованы — A, B или C. Предположим также, что на ранних совещаниях по планированию были определены три ключевых системных требования и четыре ключевых ограничения, которые можно было бы использовать для принятия решения о том, какую альтернативу выбрать. В левом столбце [рисунка 4-4](#bookmark0) перечислены три системных требования и четыре ограничения. Поскольку не все требования и ограничения имеют одинаковую важность, они взвешиваются на основе их относительной важности. Другими словами, вам не нужно одинаково взвешивать требования и ограничения; конечно, можно сделать требования более или менее важными, чем ограничения. Весовые коэффициенты определяются в ходе обсуждений между аналитической группой, пользователями, а иногда и менеджерами. Весовые коэффициенты, как правило, довольно субъективны, и по этой причине они должны определяться в процессе открытого обсуждения для выявления основополагающих предположений с последующей попыткой достичь консенсуса между заинтересованными сторонами. Обратите внимание, что сумма весов как для требований, так и для ограничений составляет 100 процентов.



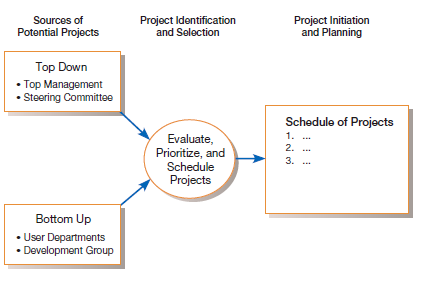
**Рисунок 4-4**

Альтернативные проекты и решения по проектированию систем могут быть реализованы с помощью взвешенного многокритериального анализа

Затем каждое требование и ограничение оценивается по шкале от 1 до 5. Оценка 1 указывает на то, что альтернатива не очень хорошо соответствует требованию или что альтернатива нарушает ограничение. Оценка 5 указывает на то, что альтернатива соответствует или превосходит требование или явно соблюдает ограничение. Рейтинги еще более субъективны, чем веса, и также должны определяться путем открытого обсуждения между пользователями, аналитиками и менеджерами. Для каждого требования и ограничения балл рассчитывается путем умножения рейтинга для каждого требования и каждого ограничения на его вес. Последним шагом является добавление взвешенных баллов для каждой альтернативы. Обратите внимание, что мы включили три набора итогов: для требований, для ограничений и общих итогов. Если вы посмотрите на итоговые показатели требований, вариант B или C является лучшим выбором, потому что оба соответствуют или превосходят все требования. Однако, если вы смотрите только на ограничения, альтернатива A является лучшим выбором, потому что она не нарушает никаких ограничений. Когда мы объединяем итоговые данные по требованиям и ограничениям, мы видим, что лучшим выбором является альтернатива C. Однако вопрос о том, будет ли на самом деле выбран вариант С для разработки, является другим вопросом. Лица, принимающие решения, могут выбрать вариант А, зная, что он не отвечает двум ключевым требованиям, потому что он имеет наименьшую стоимость. Короче говоря, то, что может быть лучшим выбором для проекта разработки систем, не всегда может быть тем, что в конечном итоге разрабатывается. Проведя тщательный анализ, организации могут значительно повысить эффективность принятия решений.

#### Конечные результаты и результаты

Основным конечным результатом первой части этапа планирования является график конкретных проектов развития ИС, поступающих как из нисходящих, так и из восходящих источников, для перехода к следующей части этапа планирования — инициированию проекта и планированию (см. [рис. 4-5).](#bookmark0) Результатом этого этапа является уверенность в том, что отбор проектов был тщательно продуман с четким пониманием того, как каждый проект может помочь организации в достижении ее целей. Из-за принципа **инкрементальности** выбранный проект не обязательно приводит к созданию работающей системы. После каждого последующего этапа SDLC вы, другие члены проектной группы и должностные лица организации будете пересматривать свой проект, чтобы определить, изменились ли бизнес-условия или более подробное понимание затрат, выгод и рисков системы предполагает, что проект не так полезен, как считалось ранее.



**Рисунок 4-5**

Проекты по разработке информационных систем реализуются как по принципу «сверху вниз», так и по принципу «снизу вверх»

Многие организации обнаружили, что для принятия правильных решений по выбору проектов требуется четкое понимание общей организационной бизнес-стратегии и задач. Это означает, что четкое понимание бизнеса и желаемой роли информационных систем в достижении целей организации является предпосылкой для улучшения процесса идентификации и отбора. В следующем разделе мы приводим краткий обзор процесса, которому следуют многие организации, включая корпоративное стратегическое планирование и планирование информационных систем, при определении стратегии и целей своей деятельности, а также при определении роли информационных систем в своих планах.

### *Планирование корпоративных и информационных систем*

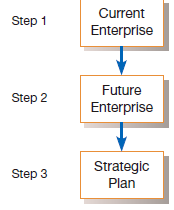
Несмотря на то, что существует множество мотивов для тщательного планирования идентификации и отбора проектов (см. Atkinson, 1990; «Дайч», 2015; Harvard Business Review, 2009; «Келли», 2006; Люфтман, 2004; Weill and Ross, 2009), организации традиционно не использовали систематический процесс планирования при определении того, как распределять ресурсы ИБ. Вместо этого проекты часто являются результатом попыток решить изолированные организационные проблемы. По сути, организации задаются вопросом: «Какой процесс (прикладная программа) требуется для решения этой конкретной проблемы в том виде, в каком она существует сегодня?» Трудность этого подхода заключается в том, что требуемый организационный процесс, вероятно, будет меняться с течением времени по мере изменения окружающей среды. Например, компания может решить изменить свой метод выставления счетов клиентам, или университет может изменить свою процедуру регистрации студентов. Когда такие изменения происходят, как правило, необходимо снова модифицировать существующие информационные системы.

Напротив, подходы, основанные на планировании, по существу задают вопрос: «Какие требования к информации (или данным) удовлетворят потребности в принятии решений или бизнес-процессы предприятия сегодня и в будущем?» Основное преимущество этого подхода заключается в том, что информационные потребности организации с меньшей вероятностью изменятся (или будут меняться медленнее), чем ее бизнес-процессы. Например, если организация принципиально не изменит свой бизнес, лежащие в ее основе структуры данных могут оставаться достаточно стабильными в течение более 10 лет. Однако процедуры, используемые для доступа к данным и их обработки, могут меняться много раз в течение этого периода. Таким образом, задача большинства организаций состоит в том, чтобы разработать всеобъемлющие информационные модели, содержащие данные, которые относительно независимы от языков и программ, используемых для их обработки, создания и обновления.

Чтобы извлечь выгоду из подхода, основанного на планировании, для определения и выбора проектов, организация должна проанализировать свои информационные потребности и тщательно спланировать свои проекты. Без тщательного планирования организации могут создавать базы данных и системы, поддерживающие отдельные процессы, но не предоставляющие ресурс, который можно было бы легко использовать в рамках всей организации. Кроме того, по мере изменения бизнес-процессов отсутствие интеграции данных и систем будет препятствовать скорости, с которой организация может эффективно вносить изменения в бизнес-стратегию или процессы.

Потребность в совершенствовании идентификации и отбора проектов информационных систем становится очевидной, если учесть такие факторы, как:

* 1. Стоимость информационных систем неуклонно растет и приближается к 40 процентам от общих расходов в некоторых организациях
  2. Многие системы не могут обрабатывать приложения, которые пересекают границы организации.
  3. Многие системы часто не решают критические проблемы бизнеса в целом или не поддерживают стратегические приложения
  4. Избыточность данных часто выходит из-под контроля, и пользователи могут быть мало уверены в качестве данных
  5. Затраты на техническое обслуживание систем выходят из-под контроля, так как старые, плохо спланированные системы должны постоянно пересматриваться.
  6. Отставание в работе приложений часто увеличивается на три года и более, и разочарованные конечные пользователи вынуждены создавать (или покупать) свои собственные системы, часто создавая избыточные базы данных и несовместимые системы в процессе



**Рисунок 4-6**

Корпоративное стратегическое планирование состоит из трех этапов

Тщательное планирование и отбор проектов сами по себе, конечно, не решат всех этих проблем. Вместе с тем мы считаем, что дисциплинированный подход, обусловленный приверженностью высшего руководства, является необходимым условием наиболее эффективного применения информационных систем для достижения организационных целей. В этом разделе основное внимание уделяется тому, чтобы дать вам четкое представление о том, как можно определить и выбрать конкретные проекты разработки с более широкой организационной направленностью. В частности, мы описываем корпоративное стратегическое планирование и планирование информационных систем, два процесса, которые могут значительно повысить качество идентификации проектов и решений по выбору. В этом разделе также описываются типы информации о бизнес-направлениях и общих системных требованиях, которые могут влиять на решения о выборе и направлять направление утвержденных проектов.

#### Корпоративное и стратегическое планирование

Предпосылкой для принятия эффективных решений по выбору проекта является получение четкого представления о том, где находится организация, ее видение того, где она хочет быть в будущем, и как осуществить переход к желаемому будущему состоянию. [На рисунке 4-6](#bookmark0) показан трехэтапный процесс. Первый шаг направлен на то, чтобы получить представление о текущем предприятии. Другими словами, если вы не знаете, где вы находитесь, невозможно сказать, куда вы идете. Затем высшее руководство должно определить, где оно хочет, чтобы предприятие находилось в будущем. Наконец, после получения представления о текущем и будущем предприятии может быть разработан стратегический план, который будет направлять этот переход. Процесс разработки и уточнения моделей текущего и будущего предприятия, а также стратегии перехода часто называют корпоративным стратегическим планированием. Во время корпоративного стратегического планирования руководители, как правило, разрабатывают заявление о миссии, заявления о будущих целях и стратегии, призванные помочь организации достичь своих целей.

У всех успешных организаций есть миссия. В заявлении о миссии компании обычно очень просто указывается, каким бизнесом занимается компания. Например, заявление о миссии Pine Valley Furniture (PVF) показано на [рисунке 4-7.](#bookmark0) Изучив миссию PVF, становится ясно, что она занимается производством и продажей высококачественной деревянной мебели широкой публике, предприятиям и учреждениям, таким как университеты и больницы. Также ясно, что PVF не занимается изготовлением стальных картотечных шкафов или продажей своей продукции через оптовые дистрибьюторы. Исходя из этой миссии, можно сделать вывод, что PVF не нуждается в информационной системе розничных продаж; вместо этого высококачественная информационная система о людских ресурсах соответствовала бы ее цели.



**Рисунок 4-7**

Миссия (Мебель Pine Valley)

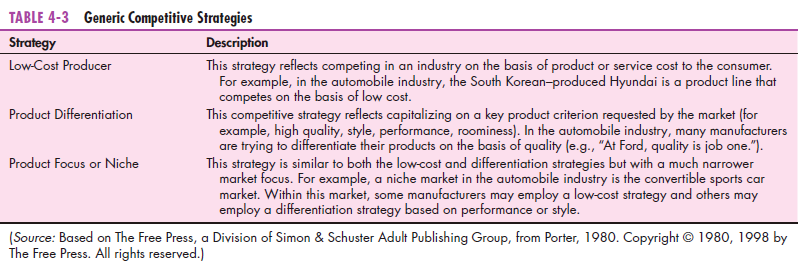


**Рисунок 4-8**

Постановка корпоративных целей (Мебель Pine Valley)

Определив свою миссию, организация может определить свои цели. **Объективные заявления** относятся к «широким и вневременным» целям организации. Эти цели могут быть выражены в виде ряда утверждений, которые являются либо качественными, либо количественными, но которые, как правило, не содержат деталей, которые могут существенно измениться с течением времени. Цели часто называют *критическими факторами успеха.*  Здесь мы просто будем использовать термин *«цели».*  Цели PVF показаны на [рисунке 4-8,](#bookmark0) причем большинство из них относятся к какому-либо аспекту организационной миссии. Например, вторая цель связана с тем, как PVF рассматривает свои отношения с клиентами. Эта цель предполагает, что PVF может захотеть инвестировать в веб-систему отслеживания заказов, которая будет способствовать высококачественному обслуживанию клиентов. После того, как компания определила свою миссию и задачи, можно сформулировать конкурентную стратегию.

**Конкурентная стратегия** — это метод, с помощью которого организация пытается достичь своей миссии и целей. По сути, стратегия представляет собой план игры организации для игры в конкурентном деловом мире. В своей классической книге о конкурентной стратегии Майкл Портер (1980) определил три общие стратегии — производитель с низкими издержками, дифференциация продукта и ориентация продукта или ниша — для достижения корпоративных целей (см[. таблицу 4-3).](#bookmark0) Эти общие стратегии позволяют легче сравнивать две компании в одной отрасли, которые могут не использовать одну и ту же конкурентную стратегию.



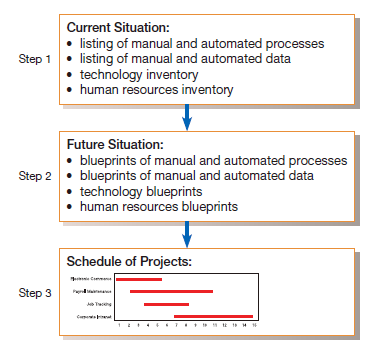
Кроме того, организации, использующие различные конкурентные стратегии, часто имеют различные информационные потребности для содействия принятию решений. Например, Rolls-Royce и Kia Motors — это две автомобильные линии с разными стратегиями: одна — это престижная линия в *нише* ультра-люкс, а другая — относительно *недорогая* линия для общего автомобильного рынка. Rolls-Royce может создавать информационные системы для сбора и анализа информации об удовлетворенности клиентов, чтобы помочь в достижении ключевых целей компании. В качестве альтернативы Kia может создавать системы для отслеживания использования оборудования и материалов для управления деятельностью, связанной с ее стратегией низких затрат.

Для эффективного использования ресурсов, таких как создание маркетинга и продаж, или для создания наиболее эффективных информационных систем, организация должна четко понимать свою миссию, цели и стратегию. Отсутствие понимания сделает невозможным знание того, какие виды деятельности необходимы для достижения бизнес-целей. С точки зрения разработки информационных систем, понимая, какие виды деятельности наиболее важны для достижения бизнес-целей, организация имеет гораздо больше шансов определить те виды деятельности, которые должны поддерживаться информационными системами. Иными словами, *только благодаря четкому пониманию миссии, целей и стратегий организации проекты развития ИС должны быть определены и отобраны.*  Процесс планирования того, как информационные системы могут быть использованы для оказания помощи организациям в достижении их целей, находится в центре внимания следующего раздела.

#### Планирование информационных систем

Второй процесс планирования, который может сыграть значительную роль в качестве идентификации проекта и принятия решений по выбору, называется **планированием информационных систем (ISP).** ISP — это упорядоченное средство оценки информационных потребностей организации и определения информационных систем, баз данных и технологий, которые наилучшим образом удовлетворят эти потребности (Amrollahi et al., 2014; Carlson et al., 1989; «Кэссиди», 2005; Люфтман, 2004; «Оверби», 2008; Паркер и Бенсон, 1989; Сегарс и Гровер, 1999; Weill and Ross, 2009). Это означает, что во время ISP вы (или, что более вероятно, старшие менеджеры ИБ, ответственные за план ИБ) должны моделировать текущие и будущие информационные потребности организации и разрабатывать стратегии и планы проектов для переноса текущих информационных систем и технологий в желаемое будущее состояние. ISP — это нисходящий процесс, который учитывает внешние силы — отрасль, экономику, относительный размер, географический регион и т. д., — которые имеют решающее значение для успеха фирмы. Это означает, что интернет-провайдер должен смотреть на информационные системы и технологии с точки зрения того, как они помогают бизнесу достигать своих целей, определенных в ходе корпоративного стратегического планирования.

Три ключевых вида деятельности этого процесса моделирования представлены на рисунке 4-9. Как и корпоративное стратегическое планирование, ISP представляет собой трехэтапный процесс, в котором первым шагом является оценка текущих активов, связанных с ИС, — человеческих ресурсов, данных, процессов и технологий. Далее разрабатываются целевые чертежи этих ресурсов. Эти проекты отражают желаемое будущее состояние ресурсов, необходимых организации для достижения ее целей, определенных в ходе стратегического планирования. Наконец, определяется ряд запланированных проектов, которые помогут организации перейти из текущего состояния в желаемое будущее. (Конечно, запланированные проекты из процесса ISP являются лишь одним из источников проектов. Другие включают восходящие запросы от менеджеров и бизнес-подразделений, такие как SSR на рисунке 3-2.)



**Рисунок 4-9**

Планирование информационных систем представляет собой трехэтапный процесс

Например, проект может быть направлен на реконфигурацию телекоммуникационной сети для ускорения передачи данных или может реструктурировать работу и потоки данных между бизнес-областями. Проекты могут включать в себя не только разработку новых информационных систем или модификацию существующих, но и приобретение и управление новыми системами, технологиями и платформами. Эти три вида деятельности параллельны корпоративному стратегическому планированию, и эта взаимосвязь показана на рисунке 4-10. Для поддержки процесса ISP были разработаны многочисленные методологии, такие как планирование бизнес-систем (BSP) и информационная инженерия (IE) (см. Amrollahi et al., 2014; Сегарс и Гровер, 1999); Большинство из них содержат следующие три ключевых вида деятельности:

* + 1. *Опишите текущую ситуацию*. Наиболее широко используемый подход к описанию текущей организационной ситуации в целом называется нисходящим планированием. **Нисходящее планирование** пытается получить широкое представление об информационных потребностях всей организации. Этот подход начинается с проведения всестороннего анализа миссии, целей и стратегии организации и определения информационных требований, необходимых для достижения каждой цели. Этот подход к ИСП подразумевает под своим названием высокоуровневую организационную перспективу с активным участием руководства высшего уровня. Нисходящий подход к ПУИ имеет ряд преимуществ по сравнению с другими подходами к планированию, которые кратко изложены в таблице 4-4.

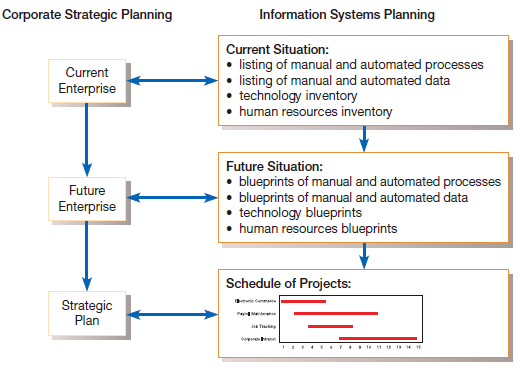
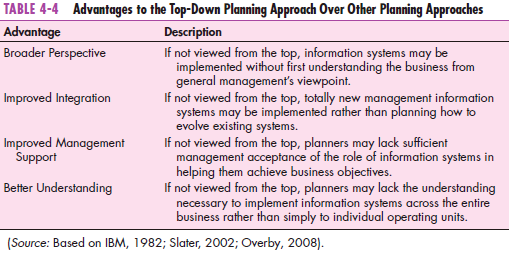
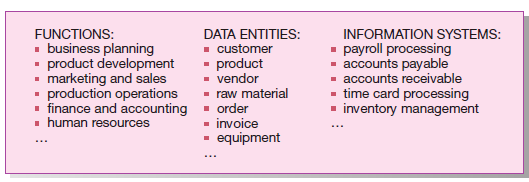


Рисунок 4-10 Параллельная деятельность корпоративного стратегического планирования и планирования информационных систем



В отличие от нисходящего подхода к **планированию, подход к планированию «снизу вверх**» требует выявления бизнес-проблем и возможностей, которые используются для определения проектов. Использование подхода «снизу вверх» для создания планов ИБ может быть более быстрым и менее затратным, чем использование подхода «сверху вниз»; Его преимущество также заключается в выявлении насущных организационных проблем. Тем не менее, подход «снизу вверх» часто не учитывает информационные потребности *всей* организации. Это может привести к созданию разрозненных информационных систем и баз данных, которые являются избыточными или нелегко интегрируются без существенной доработки.

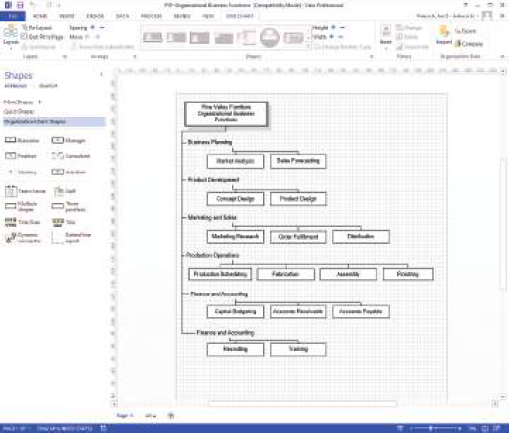
Процесс описания текущей ситуации начинается с выбора группы планирования, в которую входят руководители, уполномоченные моделировать существующую ситуацию. Чтобы получить это понимание, команде необходимо будет проанализировать корпоративные документы; опрос менеджеров, руководителей и клиентов; и проводить подробные обзоры конкурентов, рынков, продуктов и финансов. Тип информации, которая должна быть собрана для представления текущей ситуации, включает идентификацию всех организационных локаций, подразделений, функций, процессов, данных (или сущностей данных) и информационных систем.



**Рисунок 4-11**

Информация о планировании информационных систем (Мебель Pine Valley))

Например, в рамках PVF местонахождение организации будет состоять из списка всех географических район Мебель Pine Valley) Мебель Pine Valley), в которых работает организация (например, местонахождение домашнего офиса и филиалов). Организационные подразделения представляют собой список людей или бизнес-подразделений, которые работают в организации. Таким образом, организационные подразделения будут включать вице-президента по производству, менеджера по продажам, продавца и клерка. Функции представляют собой межорганизационные наборы действий, используемых для выполнения повседневных бизнес-операций. Примерами бизнес-функций могут быть исследования и разработки, развитие сотрудников, закупки и продажи. Процессы представляют собой список ручных или автоматизированных процедур, предназначенных для поддержки бизнес-функций. Проверка бизнес-процессов может включать обработку заработной платы, выставление счетов клиентам и доставку продукции. Сущности данных представляют собой список элементов информации, создаваемых, обновляемых, удаляемых или используемых в бизнес-процессах. Информационные системы представляют собой автоматизированные и неавтоматизированные системы, используемые для преобразования данных в информацию для поддержки бизнес-процессов. Например, на рисунке 4-11 показаны части бизнес-функций, сущностей данных и информационных систем PVF. После сбора высокоуровневой информации каждый элемент обычно можно разложить на более мелкие блоки по мере выполнения более детального планирования. На рисунке 4-12 показана декомпозиция нескольких высокоуровневых бизнес-функций PVF на более подробные вспомогательные функции.

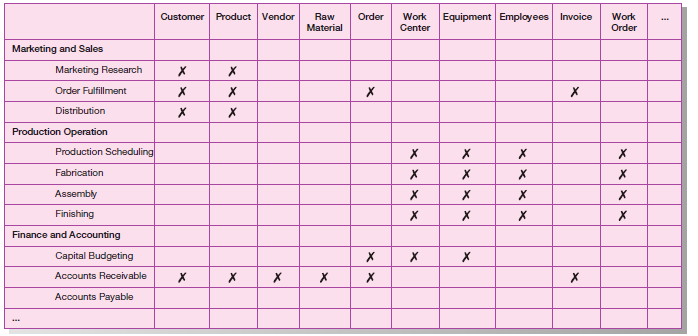


**Рисунок 4-12**

Функциональная декомпозиция информации о планировании информационных систем (Мебель Pine Valley) (*Источник:* Microsoft Corporation.)

После создания этих списков можно разработать серию матриц для перекрестных ссылок на различные элементы организации. Типичные типы матриц включают следующее:

1. Местоположение для функции: Эта матрица определяет, какие бизнес-функции выполняются в различных организационных подразделениях.
2. Местоположение к подразделению: Эта матрица определяет, какие организационные подразделения расположены в определенном бизнес-местоположении или взаимодействуют с ним.
3. От единицы к функции: Эта матрица определяет отношения между организационными подразделениями и каждой бизнес-функцией.
4. Функция к цели: Эта матрица определяет, какие функции являются важными или желательными для достижения каждой организационной цели.
5. От функции к процессу: Эта матрица определяет, какие процессы используются для поддержки каждой бизнес-функции.
6. Сущность от функции к данным: Эта матрица определяет, какие бизнес-функции используют те или иные сущности данных.
7. Сущность процесса-данных: Эта матрица определяет, какие данные собираются, используются, обновляются или удаляются в каждом процессе.
8. Процесс-информационная система: Эта матрица определяет, какие информационные системы используются для поддержки каждого процесса.
9. Data Entity-to-Information System: Эта матрица определяет, какие данные создаются, обновляются, доступны или удаляются в каждой системе.
10. Информационная система-цель: Эта матрица определяет, какие информационные системы поддерживают каждую бизнес-цель, определенную в ходе организационного планирования.



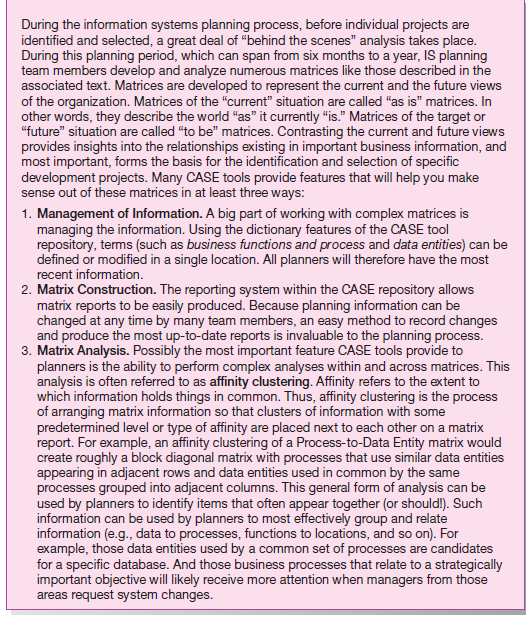
**Рисунок 4-13**

Матрица «сущность данных-функция» (мебель Pine Valley)

Различные матрицы будут иметь разные отношения в зависимости от того, что представляется. [Например, на рисунке 4-13](#bookmark0) показана часть матрицы Преобразование сущности данных в функцию для PVF. «X» в различных ячейках матрицы показывает, какие бизнес-функции используют те или иные сущности данных. Более подробная картина использования данных будет показана в матрице Сущность «процесс-данные» (здесь не показана), в которой ячейки будут закодированы как «C» для связанного процесса, который создает или захватывает данные для связанной сущности данных, «R» для извлечения (или использования), «U» для обновления и «D» для удаления. Это означает, что разные матрицы могут иметь разные отношения в зависимости от того, что представляется. Из-за этой гибкости и простоты представления информации аналитики используют широкий спектр матриц, чтобы получить четкое представление о текущей ситуации в организации и спланировать ее будущее (Керр, 1990). Учебник по использованию матриц для ISP приведен на рисунке 4-14.

* + 1. Описание целевых ситуаций, тенденций и ограничений. После описания текущей ситуации следующим шагом в процессе ISP является определение целевой ситуации, отражающей желаемое будущее состояние организации. Это означает, что целевая ситуация состоит из желаемого состояния местоположений, единиц, функций, процессов, данных и ИС (см. рис. 4-9). Например, если желаемое будущее состояние организации состоит в том, чтобы иметь несколько новых филиалов или новую линейку продуктов, которая требует нескольких новых должностей сотрудников, функций, процессов и данных, то большинство списков и матриц должны быть обновлены, чтобы отразить это видение. Целевая ситуация должна разрабатываться с учетом технологических и деловых тенденций, а также организационных ограничений. Это означает, что необходимо также составлять перечни тенденций и ограничений предпринимательской деятельности, с тем чтобы обеспечить отражение этих проблем в целевой ситуации.

Таким образом, чтобы создать целевую ситуацию, планировщики должны сначала отредактировать свои первоначальные списки и записать желаемые местоположения, подразделения, функции, процессы, данные и информационные системы в рамках ограничений и тенденций организационной среды (например, время, ресурсы, технологическая эволюция, конкуренция и т. д.). Затем матрицы обновляются, чтобы связать информацию таким образом, чтобы она соответствовала желаемому будущему состоянию. Затем планировщики сосредотачиваются на различиях между текущими и будущими списками и матрицами для определения проектов и стратегий перехода.

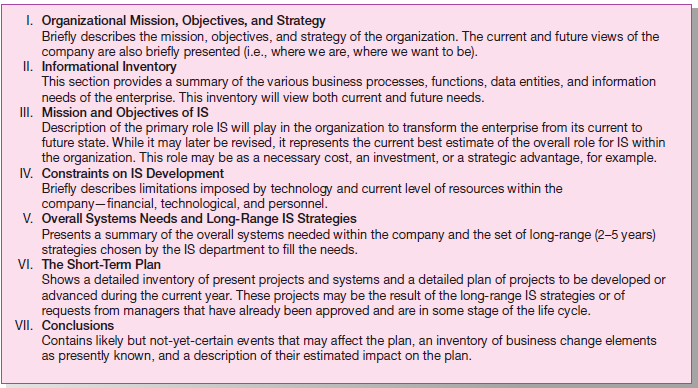


**Рисунок 4-14**

Осмысление матриц планирования

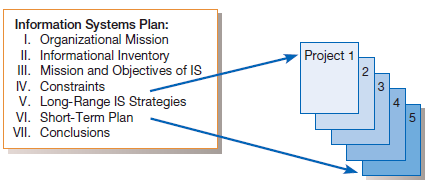
* + 1. Разработка стратегии и планов перехода. После того, как создание текущей и целевой ситуаций завершено, группа планирования ИС разрабатывает подробную стратегию и план перехода. Этот план должен быть очень всеобъемлющим, отражать широкие, долгосрочные вопросы, а также содержать достаточную детализацию для руководства на всех уровнях управления в отношении того, что, как, когда и кем должно быть сделано в организации. Компоненты типового плана информационных систем показаны на рисунке 4-15.

План ИБ, как правило, представляет собой очень всеобъемлющий документ, в котором рассматриваются как краткосрочные, так и долгосрочные потребности организационного развития. Краткосрочные и долгосрочные потребности в области развития, определенные в плане, обычно выражаются в виде ряда проектов (см. диаграмму 4-16). Проекты долгосрочного плана, как правило, закладывают основу для более поздних проектов (например, преобразование баз данных из старой технологии в новую). Проекты из краткосрочного плана состоят из конкретных шагов по заполнению разрыва между текущими и желаемыми системами или реагированию на динамические условия бизнеса. Нисходящие (или ориентированные на план) проекты присоединяются к набору восходящих или ориентированных на потребности проектов, представленных менеджерами в виде запросов на системные услуги для формирования краткосрочного плана развития систем. В совокупности краткосрочные и долгосрочные проекты задают четкие направления для процесса отбора проектов. Краткосрочный план включает в себя не только те проекты, которые были определены в процессе планирования, но и проекты, отобранные из числа заявок по принципу «снизу вверх». Общий план ИБ также может влиять на все проекты развития. Например, миссия ИБ и ограничения ИБ могут привести к тому, что проекты будут выбирать определенные технологии или делать акцент на определенных особенностях применения при проектировании систем.



**Рисунок 4-15**

Набросок плана информационных систем



**Рисунок 4-16**

Проекты разработки систем вытекают из плана информационных систем

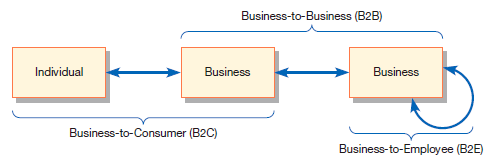
В этом разделе мы описали общий процесс разработки плана ИБ. ISP — это подробный процесс и неотъемлемая часть принятия решения о том, как наилучшим образом развернуть информационные системы и технологии для достижения организационных целей. Однако подробное обсуждение ПУИ выходит за рамки данной главы, однако из нашего обсуждения должно быть ясно, что идентификация и отбор проектов на основе планирования принесут организации существенные выгоды. Вероятно, вам также ясно, что, будучи системным аналитиком, вы обычно не участвуете в планировании ИБ, потому что этот процесс требует участия высшего руководства ИБ и корпоративного руководства. С другой стороны, результаты планирования ИБ, такие как матрицы планирования, подобные той, что показана на рисунке 4-13, могут быть источником очень ценной информации при выявлении и обосновании проектов.

### *Приложения для электронной торговли. Идентификация и отбор проектов разработки систем*

Идентификация и отбор проектов разработки систем для интернет-приложений электронной коммерции ничем не отличается от процесса, применяемого для более традиционных приложений. Тем не менее, при разработке интернет-приложения необходимо учитывать некоторые особенности. В этом разделе мы осветим некоторые из тех вопросов, которые имеют непосредственное отношение к процессу определения и выбора проектов разработки систем, связанных с Интернетом.

#### Основы работы с Интернетом

Название **Интернет** происходит от понятия «межсетевое взаимодействие», то есть соединение хост-компьютеров и их сетей для формирования еще более крупной глобальной сети. И это, по сути, то, чем является Интернет — большая всемирная сеть сетей, которые используют общий протокол для связи друг с другом. Взаимосвязанные сети включают в себя компьютеры под управлением Windows, Linux, IOS и многих других типов сетей и компьютеров. Интернет является наиболее заметным представлением глобальных сетей. Использование Интернета для поддержки повседневной предпринимательской деятельности широко называется **электронной коммерцией (ЭК).** Тем не менее, не все возможности Интернета одинаковы. Например, существует три основных класса интернет-приложений EC: бизнес-потребитель (B2C), бизнес-бизнес ( B2B) и **бизнес-сотрудник (B2E).** [На рисунке 4-17](#bookmark0) показаны три возможных режима ЭК с использованием Интернета. B2C относится к бизнес-транзакциям между отдельными потребителями и предприятиями. B2B относится к деловым операциям между деловыми партнерами, такими как поставщики и посредники. B2E относится к использованию Интернета в рамках одного бизнеса для поддержки развития сотрудников и внутренних бизнес-процессов. B2E иногда называют интрасетью.



**Рисунок 4-17**

Три возможных способа электронной торговли

Электронная коммерция B2E и B2B являются примерами двух способов, которыми организации общаются с помощью технологий в течение многих лет. Например, B2E очень похож на «глобальную» локальную сеть (LAN). Организации, использующие возможности B2E, будут выбирать различные приложения или ресурсы, расположенные в интрасети, такие как база данных контактов клиентов или система управления запасами, к которым могут получить доступ только члены организации. Аналогичным образом, B2B используют Интернет для предоставления возможностей, аналогичных установленной вычислительной модели, **электронному обмену данными (EDI).** ЭОД относится к использованию телекоммуникационных технологий для прямой передачи деловых документов между организациями. Используя ЭОД, торговые партнеры (поставщики, производители, клиенты и т.д.) устанавливают межкомпьютерные связи, которые позволяют им обмениваться данными в электронном виде. Например, компания, использующая EDI, может отправить поставщику электронный заказ на поставку вместо бумажного запроса. Доставка бумажного заказа поставщику может занять несколько дней, в то время как заказ на поставку EDI займет всего несколько секунд. Передача данных типа EDI через Интернет, обычно называемая транзакциями B2B, стала стандартом, по которому организации общаются друг с другом в мире электронной коммерции.

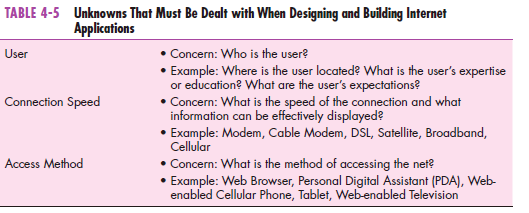
При разработке приложения B2E или B2B разработчики знают, кто является пользователями, какие приложения будут использоваться, скорость сетевого соединения и тип поддерживаемых устройств связи (например, веб-браузеры, такие как Firefox, или смартфоны с доступом в Интернет, такие как iPhone). С другой стороны, при разработке интернет-приложения EC (далее просто EC) существует бесчисленное множество неизвестных, которые разработчики должны различить, чтобы построить полезную систему. [В таблице 4-5](#bookmark0) приведен пример многочисленных неизвестных, с которыми необходимо иметь дело при проектировании и создании приложения EC. Эти неизвестные могут привести к компромиссам, основанным на тщательном анализе того, кем могут быть пользователи, где они могут находиться и как они могут быть подключены к Интернету. Несмотря на все эти трудности, с которыми приходится бороться, нет недостатка в интернет-приложениях, возникающих по всему миру. Одной из компаний, которая решила выйти в Интернет со своим собственным сайтом EC, является PVF.

#### Интернет-магазин мебели в Пайн-Вэлли

Совет директоров PVF обратился с просьбой о создании проектной группы для изучения возможности разработки системы ЕС. В частности, исследование рынка показало, что существует хорошая возможность для покупки мебели через Интернет, особенно в следующих областях:

* + - * 1. Корпоративная мебель
        2. Мебель для домашнего офиса
        3. Студенческая мебель

Совет директоров хочет включить все три целевых рынка в свой долгосрочный план ЕС, но первоначально хочет сосредоточиться на системе покупки корпоративной мебели. Члены совета директоров считают, что этот сегмент обладает наибольшим потенциалом для обеспечения адекватной отдачи от инвестиций и станет хорошим строительным блоком для выхода на клиентоориентированные рынки. Поскольку система закупок корпоративной мебели будет ориентирована на рынок мебели для бизнеса, будет легче определить операционные требования системы. Кроме того, эта система EC должна хорошо интегрироваться с двумя существующими в настоящее время системами: Buying Fulfillment и Customer Tracking. Вместе эти атрибуты делают его идеальным кандидатом для инициирования веб-стратегии PVF. На протяжении оставшейся части книги мы будем следить за эволюцией проекта WebStore до тех пор, пока он не станет работоспособным для PVF.



## *Резюме*

В этой главе мы описали первое крупное мероприятие этапа планирования SDLC — идентификацию и отбор проектов. Идентификация и отбор проектов состоит из трех основных видов деятельности: выявление потенциальных проектов развития, классификация и ранжирование проектов и отбор проектов для развития. Для выполнения этого процесса могут быть назначены различные организационные члены или подразделения, включая высшее руководство, разнообразный руководящий комитет, бизнес-подразделения и функциональных менеджеров, группу разработки или самого старшего руководителя ИБ. Потенциальные проекты могут быть оценены и отобраны с использованием широкого спектра критериев, таких как анализ производственно-сбытовой цепочки, согласование со стратегией бизнеса, потенциальные выгоды, наличие ресурсов и потребности, а также риски.

Качество процесса идентификации и отбора проектов может быть улучшено, если решения будут руководствоваться корпоративным стратегическим планированием и ISP. Корпоративное стратегическое планирование — это процесс определения миссии, целей и стратегии организации. Решающее значение в этом процессе имеет выбор конкурентной стратегии, в которой указывается, как организация планирует достичь своих целей.

ISP — это упорядоченное средство для оценки информационных потребностей организации и определения систем и баз данных, которые наилучшим образом удовлетворят эти потребности. ISP — это нисходящий процесс, который учитывает внешние силы, которые управляют бизнесом, и факторы, имеющие решающее значение для успеха фирмы. ISP оценивает текущий перечень систем и желаемое будущее состояние организации и ее системы, а затем определяет, какие проекты необходимы для преобразования систем в соответствии с будущими потребностями организации.

Корпоративное планирование и планирование ИБ тесно взаимосвязаны. Концептуально эти отношения можно рассматривать с помощью различных матриц, которые показывают, как организационные цели, местоположения, единицы, функции, процессы, сущности данных и системы связаны друг с другом. Будут отобраны те проекты, которые считаются наиболее важными для поддержки организационной стратегии.

Интернет — это глобальная сеть, состоящая из тысяч взаимосвязанных отдельных сетей, которые взаимодействуют друг с другом по общему протоколу. Электронная торговля (ЭК) относится к использованию Интернета для поддержки повседневной предпринимательской деятельности. EC между бизнесом и потребителем относится к транзакциям между отдельными потребителями и предприятиями. EC от бизнеса к сотруднику относится к использованию Интернета в одной организации. EC между предприятиями относится к использованию Интернета между фирмами

Цель этой главы заключалась в том, чтобы дать вам более четкое представление о том, как организации определяют и выбирают проекты. Совершенствование идентификации и отбора проектов необходимо по следующим причинам: стоимость информационных систем быстро растет, системы не могут обрабатывать приложения, выходящие за рамки организационных границ, системы часто не решают критически важные организационные задачи, избыточность данных часто выходит из-под контроля, а расходы на обслуживание систем продолжают расти. Таким образом, эффективная идентификация и отбор проектов имеют важное значение для того, чтобы организации могли получить наибольшие выгоды от информационных систем.

## *Глава 5 Инициирование и планирование проектов по разработке систем*

### *Знакомство*

На первом этапе планирования жизненного цикла разработки систем (SDLC) выполняются два основных действия. Первый, идентификация и отбор проектов, фокусируется на мероприятиях, в ходе которых признается необходимость в новой или усовершенствованной системе. Это мероприятие не связано с конкретным проектом, а скорее определяет портфель проектов, которые будут осуществляться организацией. Таким образом, идентификация и выбор проекта часто рассматриваются как «предпроектный» шаг в жизненном цикле. Это признание потенциальных проектов может быть частью более широкого процесса планирования, планирования информационных систем или по запросам менеджеров и бизнес-подразделений. Независимо от того, как проект идентифицирован и выбран, следующим шагом является проведение более детальной оценки во время инициации и планирования проекта. В этой оценке основное внимание уделяется не тому, как будет функционировать предлагаемая система, а пониманию масштабов предлагаемого проекта и возможности его завершения с учетом имеющихся ресурсов. Крайне важно, чтобы организации понимали, следует ли выделять ресурсы на проект; в противном случае могут быть допущены очень дорогостоящие ошибки (Лаплант, 2006; Нэш, 2008). Таким образом, основное внимание в этой главе уделяется этому процессу. Инициация и планирование проекта — это когда проекты принимаются к разработке, отклоняются или перенаправляются. Именно здесь вы, как системный аналитик, начинаете играть важную роль в процессе разработки систем.

В следующем разделе кратко рассматривается процесс инициирования и планирования проекта. Затем описываются многочисленные методы оценки осуществимости проекта. Затем мы обсуждаем процесс построения базового плана проекта, в котором систематизируется информация, обнаруженная в ходе технико-экономического обоснования. После того, как этот план будет разработан, может быть проведен официальный обзор проекта. Тем не менее, прежде чем проект сможет перейти к следующему этапу жизненного цикла разработки систем — анализу — план проекта должен быть рассмотрен и принят. В заключительном основном разделе главы мы приводим обзор процесса рассмотрения проекта.

### *ИНИЦИИРОВАНИЕ И ПЛАНИРОВАНИЕ ПРОЕКТОВ ПО РАЗРАБОТКЕ СИСТЕМ*

Ключевым соображением при проведении инициации и планирования проекта (PIP) является принятие решения о том, когда заканчивается PIP и когда начинается анализ, следующий этап SDLC. Это вызывает беспокойство, поскольку многие действия, выполняемые вовремя PIP, также могут быть завершены во время анализа. Журналист (2014) говорит о трех важных вопросах, которые необходимо учитывать при принятии решения о разделении между PIP и анализом:

1. Сколько усилий нужно затратить на проект

Процесс инициации и планирования?

2. Кто отвечает за выполнение инициации проекта

А процесс планирования?

3. Почему инициация и планирование проекта являются такой сложной задачей

активность?

Найти ответ на первый вопрос, сколько усилий должно быть затрачено на процесс PIP, часто бывает непросто. Однако практический опыт показал, что время и усилия, затраченные на инициирование и планирование мероприятий, легко окупаются на более поздних этапах проекта. Правильное и проницательное планирование проекта, включая определение объема проекта, а также определение деятельности по проекту, может легко сократить время на более поздних этапах проекта. Например, тщательный технико-экономический анализ, который приводит к решению о том, что проект не стоит продолжать, может сэкономить значительные затраты ресурсов. Фактическое количество затраченного времени будет зависеть от размера и сложности проекта, а также от опыта вашей организации в построении подобных систем. Эмпирическое правило заключается в том, что от 10 до 20 процентов всех усилий по разработке должны быть затрачены на исследование PIP. Таким образом, вы не должны отказываться от того, чтобы провести значительное время в PIP, чтобы полностью понять мотивацию запрашиваемой системы.

Что касается второго вопроса, кто отвечает за выполнение PIP, большинство организаций назначают опытного системного аналитика или команду аналитиков для крупных проектов для выполнения PIP. Аналитик будет работать с предполагаемыми заказчиками (менеджерами и пользователями) системы и другим техническим персоналом по разработке при подготовке окончательного плана. Опытные аналитики, работающие с клиентами, которые полностью понимают их потребности в информационных услугах, должны иметь возможность выполнять PIP без подробного анализа, типичного для фазы анализа жизненного цикла. Менее опытные аналитики с клиентами, которые лишь смутно понимают их потребности, скорее всего, потратят больше усилий вовремя PIP, чтобы быть уверенными в том, что объем проекта и план работы выполнимы.

Что касается третьего вопроса, то ГПИ рассматривается как сложная деятельность, поскольку цель исследования ГПИ состоит в том, чтобы превратить расплывчатый документ с запросом системы в осязаемое описание проекта. Это открытый процесс. Аналитик должен четко понимать мотивацию и цели предлагаемой системы. Таким образом, эффективная коммуникация между системным аналитиком, пользователями и руководством имеет решающее значение для создания значимого плана проекта. Добиться согласия всех сторон о направлении проекта может быть сложно для межведомственных проектов, где разные стороны имеют разные бизнес-цели. Таким образом, более сложные организационные настройки для проектов приведут к увеличению времени, необходимого для анализа существующих и предлагаемых систем в ходе PIP.

В оставшейся части этой главы мы опишем необходимые действия, используемые для ответа на эти вопросы. В следующем разделе мы вернемся к действиям по инициированию и планированию проекта, первоначально изложенным в главе 3 в разделе «Управление проектом информационных систем». Далее следует краткое описание результатов и результатов этого процесса.

### *Процесс инициирования и планирования проектов развития*

Как следует из названия, два основных вида деятельности происходят во время инициации и планирования проекта (рис. 5-1). Поскольку этапы процесса инициации и планирования проекта были объяснены в главе 3, в этой главе мы сосредоточимся на описании нескольких методов, которые используются при выполнении этого процесса. Поэтому мы лишь кратко рассмотрим процесс PIP.

Инициация проекта фокусируется на действиях, предназначенных для оказания помощи в организации команды для проведения планирования проекта. Во время инициации одному или нескольким аналитикам назначается работа с клиентом, то есть членом бизнес-группы, которая запрашивала или будет затронута проектом, для установления стандартов работы и процедур связи. Примеры выполняемых видов деятельности приведены в таблице 5-1. В зависимости от размера, объема и сложности проекта некоторые мероприятия по инициированию проекта могут быть ненужными или могут быть очень сложными. Кроме того, многие организации установили процедуры оказания помощи в проведении совместных мероприятий по посвящению. Одним из ключевых видов деятельности по инициированию проекта является разработка устава проекта (определенного в главе 3).

Планирование проектов, являющееся вторым видом деятельности в рамках ГПГ, отличается от общего планирования информационных систем, в котором основное внимание уделяется оценке потребностей всей организации в информационных системах (обсуждается в главе 4). Планирование проекта — это процесс определения четких, дискретных действий и работы, необходимой для выполнения каждого мероприятия в рамках одного проекта. Целью процесса планирования проекта является разработка базового плана проекта (BPP) и заявления об объеме проекта (PSS) (Моррис и Сембер, 2008). БПП становится основой для остальной части проекта развития. В ПСС, составленном командой, четко изложены цели и ограничения проекта для заказчика. Как и в случае с процессом инициации проекта, размер, объем и сложность проекта будут определять полноту процесса планирования проекта и итоговых документов. Кроме того, необходимо будет сделать многочисленные предположения о наличии ресурсов и потенциальных проблемах. Анализ этих допущений, а также системных затрат и выгод формирует **экономическое обоснование**. Перечень мероприятий, выполняемых в ходе планирования проекта, приведен в таблице 5-2.

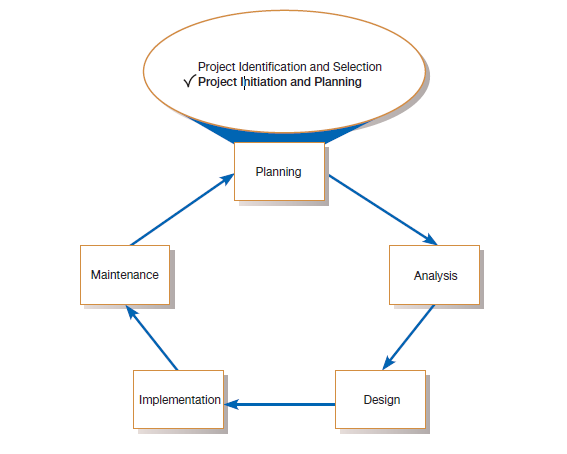
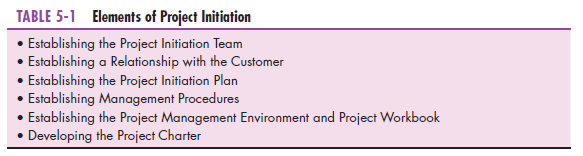


Рисунок 5-1 Жизненный цикл разработки систем с выделением инициации и планирования проекта

#### Конечные результаты и результаты

Основными итогами и конечными результатами этапа инициирования и планирования проекта являются базовый план проекта и заявление об объеме проекта. Базовый план проекта (BPP) содержит всю информацию, собранную и проанализированную во время инициирования и планирования проекта. План отражает наилучшую оценку масштаба проекта, выгод, затрат, рисков и потребностей в ресурсах с учетом текущего понимания проекта. В BPP указываются подробные проектные действия для следующего этапа жизненного цикла — анализа — и менее подробные сведения о последующих этапах проекта (поскольку они зависят от результатов этапа анализа). Аналогичным образом, выгоды, затраты, риски и потребности в ресурсах станут более конкретными и поддающимися количественной оценке по мере продвижения проекта. BPP используется комитетом по отбору проектов, чтобы помочь решить, следует ли принять, перенаправить или отменить проект. В случае выбора BPP становится основополагающим документом для всей последующей деятельности SDLC; тем не менее, ожидается, что он также будет развиваться по мере развития проекта. То есть, по мере получения новой информации на последующих этапах SDLC, базовый план будет обновляться. Далее в этой главе мы опишем, как построить BPP.





Заявление об объеме проекта (PSS) — это краткий документ, подготовленный для клиента, в котором описывается, что будет реализовано в рамках проекта, и описываются все работы, необходимые для завершения проекта. PSS гарантирует, что и вы, и ваш клиент получите общее понимание проекта. Это также очень полезный инструмент общения. PSS очень прост в создании, поскольку он обычно состоит из высокоуровневой сводки информации BPP (описанной ниже). В зависимости от ваших отношений с клиентом роль PSS может варьироваться. С одной стороны, PSS может быть использован в качестве основы для официального контрактного соглашения, в котором излагаются твердые сроки, затраты и спецификации. С другой стороны, PSS можно просто использовать в качестве средства коммуникации для определения текущих наилучших оценок того, что даст проект, когда он будет завершен и какие ресурсы он может потребовать. Например, контрактная программная или консалтинговая фирма может установить очень формальные отношения с клиентом и использовать PSS, который является обширным и формальным. В качестве альтернативы внутренняя группа разработчиков может разработать PSS, который состоит всего из одной-двух страниц и предназначен для информирования клиентов, а не для установления договорных обязательств и сроков.

### ОЦЕНКА ОСУЩЕСТВИМОСТИ ПРОЕКТА

Все проекты осуществимы при наличии неограниченных ресурсов и бесконечного времени (Pressman, 2014). К сожалению, большинство проектов приходится разрабатывать в условиях ограниченного бюджета и времени. Это означает, что оценка осуществимости проекта является обязательным мероприятием для всех проектов информационных систем и потенциально крупным мероприятием. Это требует, чтобы вы, как системный аналитик, оценивали широкий спектр факторов. Как правило, относительная важность этих факторов будет варьироваться от проекта к проекту. Хотя специфика данного проекта будет диктовать, какие факторы являются наиболее важными, большинство факторов осуществимости представлены следующими категориями:

•Экономический

•Технический

•Эксплуатационный

•Планирование

• Юридические и договорные

•Политический

В совокупности кульминация этих технико-экономических обоснований формирует экономическое обоснование, оправдывающее расходование ресурсов на проект. В оставшейся части этого раздела мы рассмотрим различные вопросы осуществимости. Мы начинаем с рассмотрения вопросов, связанных с экономической целесообразностью, а затем демонстрируем методы проведения этого анализа. Затем следует обсуждение методов оценки технического риска проекта. Наконец, обсуждаются вопросы, не связанные напрямую с экономической и технической целесообразностью, но не менее важные для обеспечения успеха проекта.

Чтобы помочь вам лучше понять процесс технико-экономического обоснования, мы рассмотрим проект в Мебель Pine Valley (PVF). Для этого проекта вице-президент PVF по маркетингу Джеки Джадсон (Jackie Judson) подал запрос на системное обслуживание (SSR) для разработки системы отслеживания клиентов (CTS) (рис. 5-2). Джеки считает, что эта система позволит маркетинговой группе PVF лучше отслеживать покупательскую активность клиентов и тенденции продаж. Она также считает, что, если КТС будет построена, она принесет много материальных и нематериальных выгод ПВФ. Этот проект был выбран Советом по системным приоритетам PVF для инициирования проекта и исследования по планированию. Во время начала проекта старшему системному аналитику Джиму Ву было поручено работать с Джеки, чтобы инициировать и планировать проект. На данном этапе проекта все мероприятия по инициированию проекта завершены. В настоящее время Джеки и Джим сосредоточены на планировании проекта, чтобы завершить BPP

#### Оценка экономической целесообразности

Целью оценки экономической целесообразности является выявление финансовых выгод и затрат, связанных с проектом развития (Лаплант, 2006). Экономическую целесообразность часто называют анализом затрат и выгод. Во время инициации и планирования проекта вы не сможете точно определить все выгоды и затраты, связанные с конкретным проектом. Тем не менее, важно, чтобы вы потратили достаточно времени на выявление и количественную оценку этих элементов, иначе вы не сможете провести адекватный экономический анализ и провести значимые сравнения между конкурирующими проектами. Здесь мы опишем типичные выгоды и затраты, возникающие в результате разработки информационной системы, и предоставим несколько полезных рабочих листов для учета затрат и выгод. Кроме того, представлено несколько распространенных методов расчета затрат и выгод. Эти рабочие листы и методы используются после каждого этапа SDLC по мере проверки проекта, чтобы решить, следует ли продолжить, перенаправить или завершить проект.

Определение преимуществ проекта Информационная система может предоставить организации множество преимуществ. Например, новая или обновленная информационная система может автоматизировать монотонные задания и уменьшить количество ошибок; предоставлять инновационные услуги клиентам и поставщикам; и повысить организационную эффективность, скорость, гибкость и моральный дух. В целом, выгоды можно рассматривать как материальные, так и нематериальные. Материальные выгоды относятся к предметам, которые можно измерить в долларах и с уверенностью. Примерами ощутимых выгод могут быть сокращение расходов на персонал, снижение транзакционных издержек или повышение прибыли. Важно отметить, что не все материальные выгоды могут быть легко оценены количественно. Например, ощутимую выгоду, которая позволяет компании выполнять задачу в 50 процентах случаев, может быть трудно количественно оценить с точки зрения экономии в долларах. Наиболее ощутимые выгоды будут укладываться в следующие категории:

• Снижение и предотвращение затрат

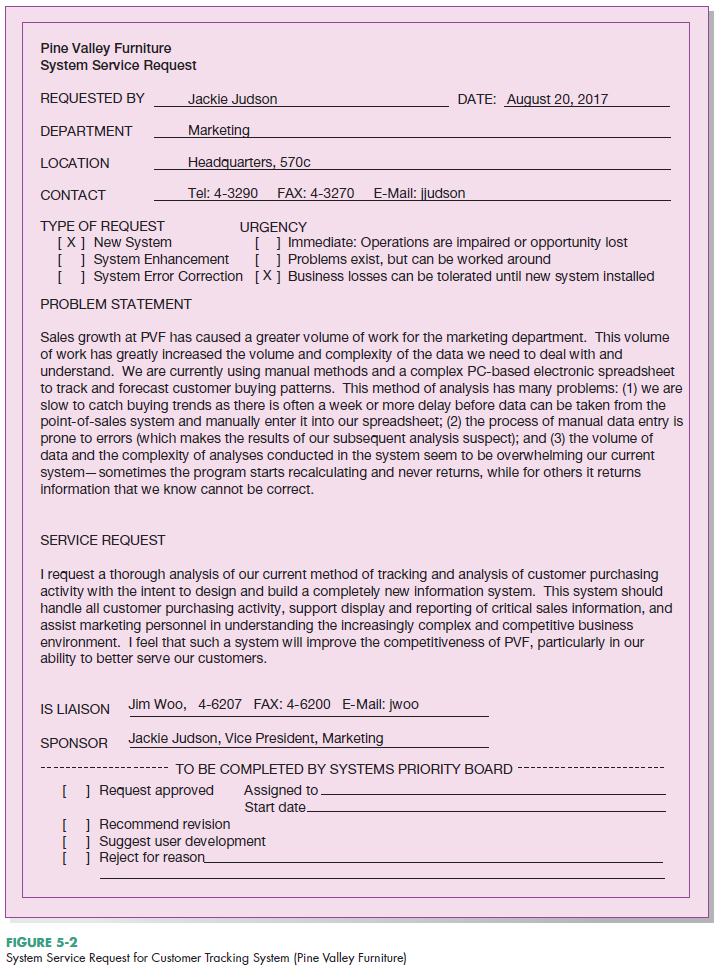
• Уменьшение ошибок

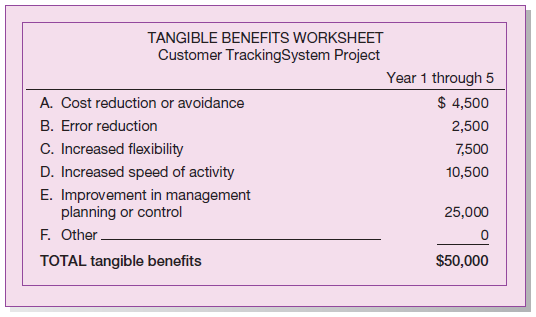
• Повышенная гибкость

• Повышенная скорость активности

• Совершенствование управленческого планирования и контроля

• Открытие новых рынков и расширение возможностей сбыта

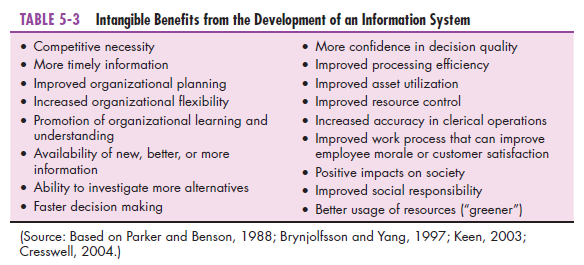




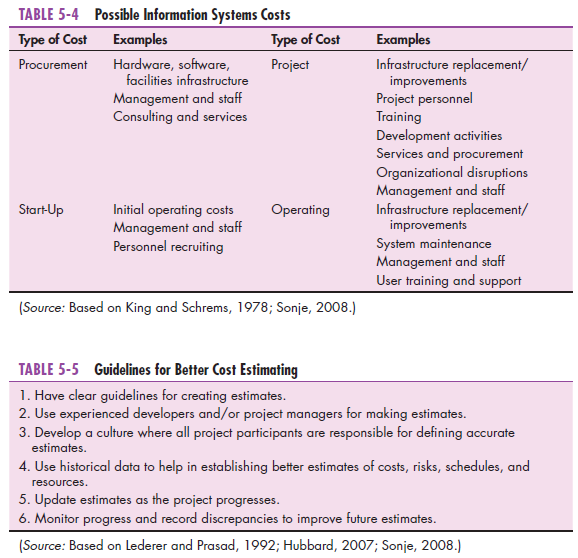
В рамках CTS в PVF Джим и Джеки определили несколько ощутимых преимуществ, которые обобщены на рабочем листе материальных преимуществ, показанном на рисунке 5-3. Джеки и Джим должны были установить значения, показанные на рисунке 5-3, после сбора информации от пользователей текущей системы отслеживания клиентов. Сначала они опросили человека, ответственного за сбор, ввод и анализ точности текущих данных отслеживания клиентов. Этот человек подсчитал, что 10 процентов ее времени было потрачено на исправление ошибок при вводе данных. Учитывая, что зарплата этого человека составляет 25 000 долларов, Джеки и Джим оценили пособие по уменьшению ошибок в 2 500 долларов. Джеки и Джим также взяли интервью у менеджеров, которые использовали текущие отчеты об отслеживании клиентов. Используя эту информацию, они смогли оценить другие ощутимые выгоды. Они узнали, что выгоды от сокращения затрат или их избегания могут быть получены благодаря более эффективному управлению запасами. Кроме того, повышение гибкости, вероятно, произойдет за счет сокращения времени, обычно необходимого для ручной реорганизации данных для различных целей. Кроме того, совершенствование управленческого планирования или контроля должно стать результатом более широкого круга анализов в рамках новой системы. В целом, согласно прогнозам этого анализа, выгоды от этой системы составят примерно 50 000 долл.

Джим и Джеки также определили несколько нематериальных преимуществ системы. Хотя эти выгоды не могут быть количественно оценены, они все же будут описаны в окончательном варианте BPP. Нематериальные выгоды относятся к предметам, которые не могут быть легко измерены в долларах или с уверенностью. Нематериальные выгоды могут иметь прямые организационные выгоды, такие как улучшение морального духа сотрудников, или они могут иметь более широкие социальные последствия, такие как сокращение образования отходов или потребления ресурсов. Потенциальные материальные выгоды, возможно, придется рассматривать как нематериальные во время инициации и планирования проекта, потому что вы не сможете количественно оценить их в долларах или с уверенностью на данном этапе жизненного цикла. На более поздних этапах такие нематериальные активы могут стать ощутимыми преимуществами, поскольку вы лучше понимаете последствия системы, которую вы проектируете. В этом случае БПП обновляется, а экономическое обоснование пересматривается, чтобы обосновать продолжение проекта до следующего этапа. В таблице 5-3 перечислены многочисленные нематериальные выгоды, часто связанные с развитием информационной системы. Фактические выгоды будут варьироваться от системы к системе. После определения выгод от проекта необходимо определить затраты по проекту.

Как и в случае с выгодами, информационная система может иметь как материальные, так и нематериальные затраты. Материальные затраты относятся к предметам, которые вы можете легко измерить в долларах и с уверенностью. С точки зрения разработки ИБ материальные затраты включают в себя такие статьи, как затраты на оборудование, затраты на рабочую силу и эксплуатационные расходы, включая обучение сотрудников и реконструкцию зданий. С другой стороны, нематериальные затраты — это статьи, которые вы не можете легко измерить в долларах или с уверенностью. Нематериальные затраты могут включать потерю деловой репутации клиентов, морального духа сотрудников или операционную неэффективность.



В таблице 5-4 приводится сводная информация об общих расходах, связанных с разработкой и эксплуатацией информационной системы. Прогнозирование затрат, связанных с разработкой информационной системы, является неточной наукой. Однако исследователи ИБ определили несколько руководящих принципов для совершенствования процесса оценки затрат (см. таблицу 5-5). Как недооценка, так и переоценка затрат являются проблемами, которых вы должны избегать (Лаплант, 2006; Ледерер и Прасад, 1992; «Нэш», 2008; Уайт и Луи, 2005). Недооценка приводит к перерасходу средств, в то время как переоценка приводит к ненужному распределению ресурсов, которые можно было бы использовать более эффективно.



Одной из целей анализа затрат и выгод является точное определение совокупной стоимости владения (TCO) для инвестиций (Nash, 2008). TCO ориентирован на понимание не только общей стоимости приобретения, но и всех затрат, связанных с текущим использованием и обслуживанием системы. Следовательно, помимо материальных и нематериальных затрат, вы можете различать затраты на разработку, связанные с ИС, как разовые или периодические (то же самое верно для выгод, хотя мы не обсуждаем эту разницу для выгод). Единовременные затраты относятся к расходам, связанным с инициированием и разработкой проекта и запуском системы. Эти расходы обычно включают в себя такие действия, как разработка систем, покупка нового оборудования и программного обеспечения, обучение пользователей, подготовка площадки и преобразование данных или систем. При проведении экономического анализа затрат и выгод следует создать рабочий лист для фиксации этих расходов. Для очень крупных проектов единовременные затраты могут быть рассчитаны на один или несколько лет. В этих случаях для каждого года следует создавать отдельный лист единовременных затрат. Такое разделение облегчит выполнение расчетов текущей стоимости (описано ниже). Периодические расходы относятся к расходам, возникающим в результате продолжающегося развития и использования системы. Примеры этих затрат обычно включают следующее:

• Сопровождение прикладного программного обеспечения

• Дополнительные расходы на хранение данных

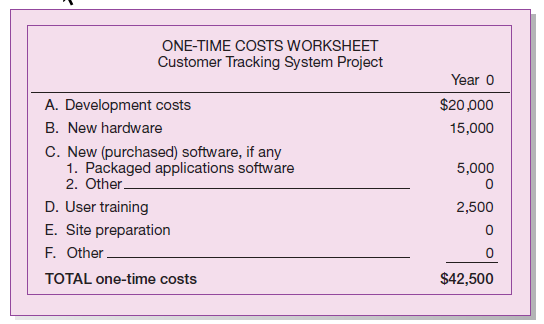
• Инкрементные коммуникации

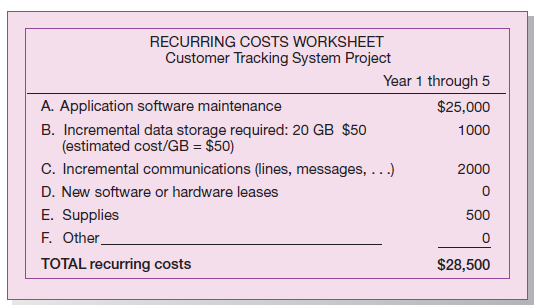
• Аренда нового программного и аппаратного обеспечения

• Расходные материалы и другие расходы (например, бумага, бланки, персонал центра обработки данных)

Как разовые, так и периодические расходы могут состоять из статей, которые носят постоянный или переменный характер. Постоянные затраты — это затраты, которые выставляются или производятся через регулярные промежутки времени и, как правило, по фиксированной ставке (арендный платеж за объект). Переменные затраты — это статьи, которые варьируются в зависимости от использования (плата за междугороднюю телефонную связь).

В процессе определения стоимости проекта Джим и Джеки определили как единовременные, так и периодические расходы по проекту. Эти расходы кратко изложены на рисунках 5-4 и 5-5. Эти цифры показывают, что этот проект повлечет за собой единовременные расходы в размере 42 500 долл. Единовременные затраты были установлены путем обсуждения системы с боссом Джима, который считал, что на разработку системы потребуется около четырех месяцев (5000 долларов в месяц). Чтобы эффективно работать с новой системой, отделу маркетинга необходимо будет модернизировать как минимум пять своих текущих рабочих станций (по 3000 долларов каждая). Кроме того, потребуются лицензии на программное обеспечение для каждой рабочей станции (по 1000 долларов США каждая) и скромная плата за обучение пользователей (десять пользователей по 250 долларов США каждая).

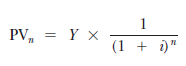


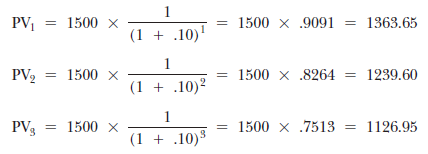


Как видно из рисунков 5-5, Джим и Джеки считают, что предлагаемая система будет очень динамичной и потребует в среднем пяти месяцев ежегодного обслуживания, в первую очередь для улучшений, поскольку пользователи ожидают от системы большего. Также следует ожидать других текущих расходов, таких как увеличение хранилища данных, коммуникационного оборудования и расходных материалов. Теперь вы должны иметь представление о типах категорий выгод и затрат, связанных с проектом информационных систем. Должно быть ясно, что существует множество потенциальных выгод и затрат, связанных с данным проектом. Кроме того, поскольку разработка и срок полезного использования системы могут охватывать несколько лет, эти выгоды и затраты должны быть нормализованы до текущих значений, чтобы проводить значимые сравнения затрат и выгод. В следующем разделе мы рассмотрим взаимосвязь между временем и деньгами.

Большинство методов, используемых для определения экономической целесообразности, охватывают концепцию временной стоимости денег (TVM), которая отражает представление о том, что деньги, доступные сегодня, стоят больше, чем такая же сумма завтра. Как уже говорилось ранее, разработка информационной системы сопряжена как с единовременными, так и с периодическими затратами. Кроме того, выгоды от разработки систем, вероятно, произойдут когда-нибудь в будущем. Поскольку многие проекты могут конкурировать за одни и те же инвестиционные доллары и иметь разную ожидаемую продолжительность полезного использования, все затраты и выгоды должны рассматриваться в связи с их текущей стоимостью при сравнении вариантов инвестиций.

Простой пример поможет в понимании ТВМ. Предположим, вы хотите купить подержанный автомобиль у знакомой, и она просит вас сделать три платежа по 1500 долларов в течение трех лет, начиная со следующего года, на общую сумму 4500 долларов. Если бы она согласилась на единовременный платеж во время продажи (и если бы у вас были деньги!), как вы думаете, на какую сумму она согласилась бы? Должен ли единовременный платеж составлять 4500 долларов? Должно ли быть больше или меньше? Чтобы ответить на этот вопрос, мы должны рассмотреть временную стоимость денег. Большинство из нас с радостью приняли бы 4500 долларов сегодня, а не три платежа по 1500 долларов, потому что доллар сегодня (или 4500 долларов, если на то пошло) стоит больше, чем доллар завтра или в следующем году, учитывая, что деньги можно инвестировать. Ставка, по которой деньги могут быть заимствованы или инвестированы, называется стоимостью капитала и называется ставкой дисконтирования для расчетов TVM. Предположим, что продавец мог бы положить деньги, полученные за продажу автомобиля, в банк и получить 10-процентный доход от своих инвестиций. Простая формула может быть использована при расчете текущей стоимости трех платежей по 1500 долларов:



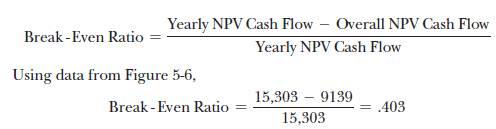




Ячейка H11 рабочего листа, показанная на рисунке 5-6, суммирует NPV общих материальных выгод от проекта. Ячейка H19 суммирует NPV общих затрат по проекту. Чистая приведенная стоимость проекта (35 003 долл. США) показывает, что в целом выгоды от проекта превышают затраты (см. ячейку H22).

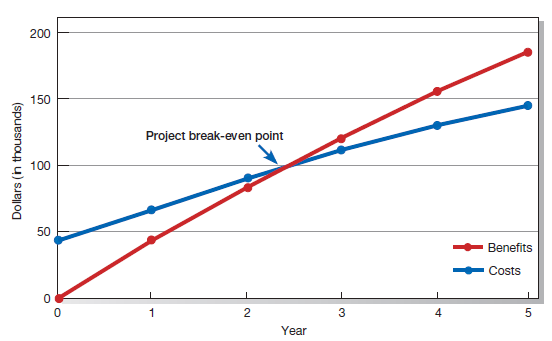
Общая рентабельность инвестиций (ROI) для проекта также показана на листе в ячейке H25. Поскольку альтернативные проекты, скорее всего, будут иметь разные выгоды и затраты и, возможно, разную продолжительность жизни, общее значение рентабельности инвестиций очень полезно для сравнения проектов на экономической основе. Конечно, этот пример показывает рентабельность инвестиций для всего проекта; анализ рентабельности инвестиций может быть рассчитан для каждого года проекта.

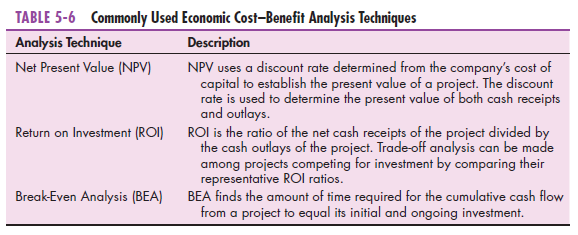
Последний анализ, показанный на рисунке 5-6, является анализом безубыточности. Цель анализа безубыточности состоит в том, чтобы выяснить, в какой момент (если вообще когда-либо) выгоды равны затратам (т.е. когда наступает безубыточность). Для проведения этого анализа определяется NPV годовых денежных потоков. Здесь годовые денежные потоки рассчитываются путем вычитания как единовременных затрат, так и приведенной стоимости периодических расходов из приведенной стоимости годовых выгод. Общая чистая приведенная стоимость денежного потока отражает общий денежный поток за все предыдущие годы. Изучение строки 30 рабочего листа показывает, что безубыточность наступает между 2 и 3 годами. Поскольку год 3 является первым, в котором общий показатель денежного потока NPV является неотрицательным, определение того, какой момент в течение года наступает безубыточность, может быть получено следующим образом:



Таким образом, безубыточность проекта наступает примерно через 2,4 года. Графическое представление этого анализа показано на рисунке 5-7. Используя информацию, полученную в результате экономического анализа, Совет по приоритетам систем PVF будет в гораздо лучшем положении, чтобы понять потенциальное экономическое влияние CTS. Из этого анализа должно быть ясно, что без такой информации было бы практически невозможно узнать о затратах и выгодах предлагаемой системы и невозможно принять обоснованное решение относительно одобрения или отклонения запроса на обслуживание.

Вы можете использовать множество методов для расчета экономической целесообразности проекта. Поскольку срок полезного использования большинства информационных систем составляет более одного года, и они будут приносить выгоды и нести расходы в течение более одного года, в большинстве методов анализа экономической целесообразности используется концепция TVM. Некоторые из этих методов анализа затрат и выгод довольно просты, в то время как другие более сложны. В таблице 5-6 описаны три наиболее часто используемых метода проведения анализа экономической целесообразности. Для более подробного обсуждения TVM или методов анализа затрат и выгод в целом заинтересованному читателю рекомендуется ознакомиться с вводным учебником по финансовому или управленческому учету.





Системный проект, подлежащий утверждению для продолжения, может не достигать безубыточности или иметь рентабельность инвестиций выше некоторого организационного порога, как это было оценено в ходе инициации и планирования проекта. Поскольку вы, возможно, не сможете количественно оценить многие выгоды или затраты на этом этапе проекта, такие финансовые препятствия для проекта могут быть недостижимыми. В этом случае для продвижения проекта может быть достаточно просто провести как можно более тщательный экономический анализ, включая составление длинного списка нематериальных активов. Еще один вариант заключается в проведении экономического анализа, показанного на рисунке 5-7, с использованием пессимистичных, оптимистичных и ожидаемых оценок выгод и затрат во время инициации и планирования проекта. Этого диапазона возможных результатов, наряду с перечнем нематериальных выгод и поддержкой со стороны запрашивающего бизнес-подразделения, часто бывает достаточно, чтобы позволить проекту перейти к этапу анализа. Тем не менее, вы должны быть как можно точнее в экономическом анализе, особенно когда инвестиционный капитал недостаточен. В этом случае, возможно, потребуется провести некоторые типичные мероприятия на этапе анализа во время инициирования и планирования проекта, чтобы четко определить неэффективность и недостатки существующей системы и объяснить, как новая система сможет преодолеть эти проблемы. Таким образом, создание экономического обоснования для системного проекта является неограниченной деятельностью; Объем анализа зависит от конкретного проекта, заинтересованных сторон и условий ведения бизнеса. Кроме того, проведение анализа экономической целесообразности для новых типов информационных систем часто бывает очень сложным.

#### Оценка технической осуществимости

Цель оценки технической осуществимости состоит в том, чтобы получить представление о способности организации построить предлагаемую систему. Этот анализ должен включать оценку понимания группой разработчиков возможного целевого оборудования, программного обеспечения и операционных сред, которые будут использоваться, а также размера системы, сложности и опыта группы с аналогичными системами. В этом разделе мы обсудим структуру, которую вы можете использовать для оценки технической осуществимости проекта, в которой уровень риска проекта может быть определен после ответа на несколько фундаментальных вопросов. Важно отметить, что все проекты сопряжены с риском, и этот риск не обязательно является чем-то, чего следует избегать. Тем не менее, верно и то, что, поскольку организации, как правило, ожидают большей отдачи от своих инвестиций в более рискованные проекты, понимание источников и типов технических рисков оказывается ценным инструментом при оценке проекта. Кроме того, необходимо управлять рисками, чтобы их можно было свести к минимуму; поэтому вы должны как можно раньше определить потенциальные риски в проекте. Потенциальные последствия отсутствия оценки рисков и управления ими могут включать следующее:

• Неспособность получить ожидаемые выгоды от проекта

• Неточная смета расходов по проекту

• Неточные оценки продолжительности проекта

• Неспособность достичь адекватного уровня производительности системы

• Неспособность адекватно интегрировать новую систему с существующим оборудованием, программным обеспечением или организационными процедурами.

Вы можете управлять рисками в проекте, изменяя план проекта, чтобы избежать рисковых факторов, назначая членов проектной группы для тщательного управления аспектами риска и настраивая методы мониторинга, чтобы определить, действительно ли потенциальный риск материализуется.

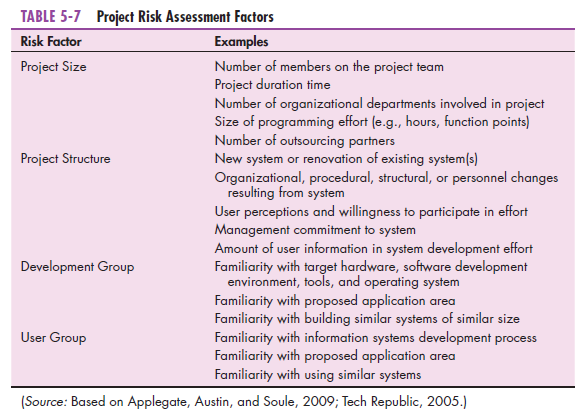
Величина технического риска, связанного с данным проектом, зависит от четырех основных факторов: размера проекта, структуры проекта, опыта группы разработчиков в области применения и технологии, а также опыта группы пользователей в проектах разработки систем и области применения (см. также Вишня, 2000). Аспекты каждой из этих областей риска кратко изложены в таблице 5-7. При использовании этих факторов для проведения оценки технических рисков вытекают четыре общих правила:

1. Крупные проекты более рискованны, чем маленькие. Размер проекта, конечно, относится к относительному размеру проекта, с которым обычно работает группа разработчиков. «Маленький» проект для одной группы разработчиков может быть относительно «большим» для другой. Типы факторов, влияющих на размер проекта, перечислены в таблице 5-7.

2. Система, в которой требования легко достижимы и хорошо структурированы, будет менее рискованной, чем система, в которой требования беспорядочны, плохо структурированы, плохо определены или подлежат суждению человека. Например, разработка системы начисления заработной платы имеет требования, которые могут быть легко получены из-за юридических требований к отчетности и стандартных процедур бухгалтерского учета. С другой стороны, разработка системы поддержки руководителей должна быть адаптирована к конкретному стилю принятия управленческих решений и важнейшим факторам успеха организации, что делает ее развитие более рискованным (см. таблицу 5-7).

3. Разработка системы, использующей общепринятую или стандартную технологию, будет менее рискованной, чем система, использующая новую или нестандартную технологию. Проект имеет большую вероятность возникновения непредвиденных технических проблем, когда группе разработчиков не хватает знаний, связанных с аспектом технологической среды. Менее рискованным подходом является использование стандартных средств разработки и аппаратных сред. Опытные разработчики систем нередко говорят о сложности использования передовых (или, по их словам, новейших) технологий (см. таблицу 5-7).

4. Проект менее рискован, когда группа пользователей знакома с процессом разработки систем и областью применения, чем если группа пользователей не знакома с ними. Успешные проекты ИБ требуют активного участия и сотрудничества между группами пользователей и разработчиков. Пользователи, знакомые с областью применения и процессом разработки систем, с большей вероятностью поймут необходимость их участия и то, как это участие может повлиять на успех проекта (см. таблицу 5-7).



Проект с высоким риском все еще может быть реализован. Многие организации рассматривают риск как проблему портфеля: рассматривая все проекты, можно иметь разумный процент проектов с высоким, средним и низким уровнем риска. Учитывая, что некоторые проекты с высоким риском могут столкнуться с проблемами, организация не может позволить себе иметь их слишком много. Наличие слишком большого количества проектов с низким уровнем риска может быть недостаточно агрессивным, чтобы совершить серьезные прорывы в инновационном использовании систем. Каждая организация должна принять решение о приемлемом сочетании проектов различной степени риска.

Матрица для оценки относительных рисков, связанных с только что описанными общими правилами, показана на рисунке 5-8. Используя правила фактора риска для оценки уровня технического риска CTS, Джим и Джеки пришли к следующему выводу о своем проекте:

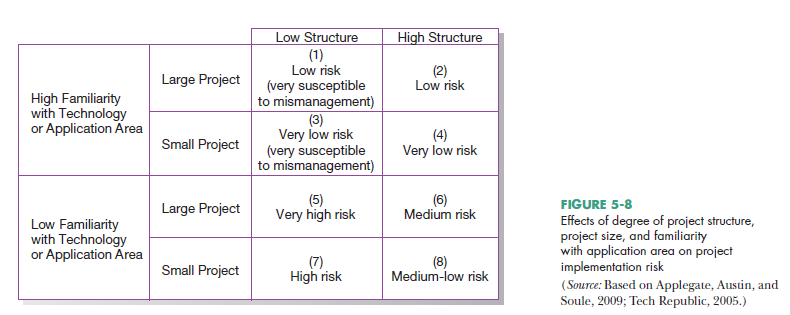
1. Проект является относительно небольшим проектом для организации разработки PVF. Исходные данные для системы легко доступны, поэтому создание системы не будет большим делом.

2. Требования к проекту хорошо структурированы и легко достижимы. Фактически, существующая система на основе электронных таблиц доступна аналитикам для изучения и изучения.

3. Группа разработчиков знакома с технологией, которая, вероятно, будет использоваться для построения системы, потому что система просто расширит текущие возможности системы.

4. Группа пользователей знакома с областью применения, потому что они уже используют систему электронных таблиц на базе ПК, описанную на рисунке 5-3.

Учитывая эту оценку риска, Джим и Джеки сопоставили свою информацию с рамками риска на рисунке 5-8. Они пришли к выводу о том, что этот проект следует рассматривать как проект с «очень низким» техническим риском (ячейка 4 рисунка). Хотя этот метод полезен для понимания технической осуществимости, многие другие вопросы могут повлиять на успех проекта. Эти нефинансовые и нетехнические вопросы описаны в следующем разделе.



#### Оценка других проблем осуществимости

В этом разделе мы кратко завершим обсуждение вопросов технико-экономического обоснования проекта, рассмотрев другие формы технико-экономического обоснования, которые вам, возможно, придется учитывать при формулировании экономического обоснования системы во время планирования проекта.

Оценка эксплуатационной осуществимости Первый связан с изучением вероятности того, что проект достигнет желаемых целей, называемых эксплуатационной осуществимостью. Его цель состоит в том, чтобы получить представление о том, в какой степени предлагаемая система, вероятно, решит бизнес-проблемы или воспользуется возможностями, изложенными в запросе на обслуживание системы или исследовании идентификации проекта. Для проекта, мотивированного планированием информационных систем, эксплуатационная осуществимость включает обоснование проекта на основе соответствия или необходимости выполнения плана информационных систем. Фактически, экономическое обоснование любого проекта можно улучшить, показав ссылку на бизнес-план или план информационных систем. Ваша оценка эксплуатационной осуществимости должна также включать анализ того, как предлагаемая система повлияет на организационные структуры и процедуры. Системы, которые оказывают существенное и широкое влияние на структуру или процедуры организации, как правило, являются более рискованными проектами. Таким образом, вам важно иметь четкое представление о том, как информационная система впишется в текущую повседневную деятельность организации.

Еще одна проблема, связанная с технико-экономическим обоснованием, связана с продолжительностью проекта и называется оценкой осуществимости графика. Цель оценки осуществимости графика состоит в том, чтобы вы, как системный аналитик, получили представление о вероятности того, что все потенциальные временные рамки и графики дат завершения могут быть соблюдены и что соблюдение этих сроков будет достаточным для удовлетворения потребностей организации. Например, система, возможно, должна быть введена в эксплуатацию к установленному правительством крайнему сроку, к определенному моменту бизнес-цикла (например, к началу сезона, когда внедряются новые продукты) или, по крайней мере, к тому времени, когда ожидается, что конкурент внедрит аналогичную систему. Кроме того, детальные мероприятия могут быть осуществимы только при наличии ресурсов, когда это предусмотрено в графике. Например, график не должен предусматривать тестирование системы в периоды спешки или ключевые совещания по проекту во время ежегодных отпусков или праздничных периодов. График мероприятий, составленный в ходе инициирования и планирования проекта, будет очень точным и подробным для этапа анализа. Предполагаемые виды деятельности и связанные с ними сроки для деятельности после этапа анализа, как правило, не так подробны (например, на программирование модуля отчета о заработной плате уйдет две недели), а находятся на уровне фазы жизненного цикла (например, потребуется шесть недель для физического проектирования, четыре месяца для программирования и т. д.). Это означает, что оценка осуществимости графика во время инициации и планирования проекта является скорее «грубым» анализом того, может ли система быть завершена в рамках ограничений бизнес-возможностей или желаний пользователей. Оценивая осуществимость графика, вы также должны оценить компромиссы в планировании. Например, такие факторы, как размер проектной группы, наличие ключевого персонала, субподрядная или аутсорсинговая деятельность, а также изменения в средах разработки, могут рассматриваться как способствующие возможному влиянию на окончательный график. Как и в случае со всеми формами технико-экономического обоснования, выполнимость графика будет переоцениваться после каждого этапа, когда вы сможете с большей уверенностью указать детали каждого шага для следующего этапа.

Третья проблема связана с оценкой вопросов юридической и договорной осуществимости. В этой области вам необходимо получить представление о любых потенциальных юридических последствиях, связанных с построением системы. Возможные соображения могут включать нарушения авторских прав или неразглашения, трудовое законодательство, антимонопольное законодательство (которое может ограничивать создание систем для обмена данными с другими организациями), правила внешней торговли (например, некоторые страны ограничивают доступ иностранных корпораций к данным сотрудников) и стандарты финансовой отчетности, а также текущие или незавершенные договорные обязательства. Договорные обязательства могут включать в себя право собственности на программное обеспечение, используемое в совместных предприятиях, лицензионные соглашения на использование аппаратного или программного обеспечения, соглашения о неразглашении с партнерами или элементы трудового соглашения (например, профсоюзное соглашение может исключать определенную компенсацию или возможности мониторинга работы, которые пользователь может захотеть использовать в системе). Распространенная ситуация заключается в том, что разработка новой прикладной системы для использования на новых компьютерах может потребовать новых или расширенных и более дорогостоящих лицензий на системное программное обеспечение. Как правило, юридическая и договорная осуществимость является более важным фактором, если ваша организация исторически использовала стороннюю организацию для конкретных систем или услуг, с которыми вы теперь планируете справиться самостоятельно. В этом случае владение исходным кодом программы другой стороной может затруднить расширение существующей системы или связывание новой системы с существующей приобретенной системой.

Оценка политической осуществимости, последняя проблема осуществимости сосредоточена на оценке политической осуществимости, в которой вы пытаетесь получить представление о том, как ключевые заинтересованные стороны в организации рассматривают предлагаемую систему. Поскольку информационная система может влиять на распределение информации внутри организации и, следовательно, на распределение власти, построение информационной системы может иметь политические последствия. Заинтересованные стороны, не поддерживающие проект, могут предпринять шаги по блокировке, нарушению или изменению предполагаемой направленности проекта.

Таким образом, в зависимости от конкретной ситуации, при планировании проекта необходимо учитывать многочисленные вопросы осуществимости. Этот анализ должен учитывать экономические, технические, эксплуатационные, графические, юридические, договорные и политические вопросы, связанные с проектом. В дополнение к этим соображениям, на выбор проекта организацией могут влиять вопросы, выходящие за рамки обсуждаемых здесь. Например, проекты могут быть отобраны для строительства, несмотря на высокие затраты по проекту и высокий технический риск, если система рассматривается как стратегическая необходимость; то есть организация рассматривает проект как критически важный для выживания организации. В качестве альтернативы могут быть выбраны проекты, поскольку считается, что они требуют мало ресурсов и сопряжены с небольшим риском. Проекты также могут быть выбраны из-за силы или убедительности менеджера, предлагающего систему. Это означает, что на выбор проекта могут влиять факторы, выходящие за рамки тех, которые обсуждаются здесь, и помимо элементов, которые могут быть проанализированы. Понимая реальность того, что проекты могут быть выбраны на основе факторов, не поддающихся анализу, ваша роль как системного аналитика заключается в тщательном изучении элементов, которые могут быть оценены. Ваш анализ гарантирует, что комитет по рассмотрению проекта имеет как можно больше информации при принятии решений об утверждении проекта. В следующем разделе мы обсудим, как обычно проверяются планы проектов.

### *ПОСТРОЕНИЕ И ПЕРЕСМОТР БАЗОВОГО ПЛАНА ПРОЕКТА*

Вся информация, собранная во время инициации и планирования проекта, собирается и систематизируется в документ, называемый базовым планом проекта. После завершения BPP может быть проведен официальный обзор проекта с клиентами проекта и другими заинтересованными сторонами. Эта презентация, пошаговое руководство, обсуждается далее в этой главе. Основное внимание в этом обзоре уделяется проверке всей информации и предположений в базовом плане, прежде чем приступать к реализации проекта.

#### ПОСТРОЕНИЕ БАЗОВОГО ПЛАНА ПРОЕКТА

Как упоминалось ранее, размер проекта и организационные стандарты будут диктовать полноту процесса инициирования и планирования проекта, а также BPP. Тем не менее, большинство опытных сборщиков систем считают, что планирование проекта и четкий план проекта неоценимы для успеха проекта. Схема BPP представлена на рисунке 5-9, который показывает, что он состоит из четырех основных разделов:

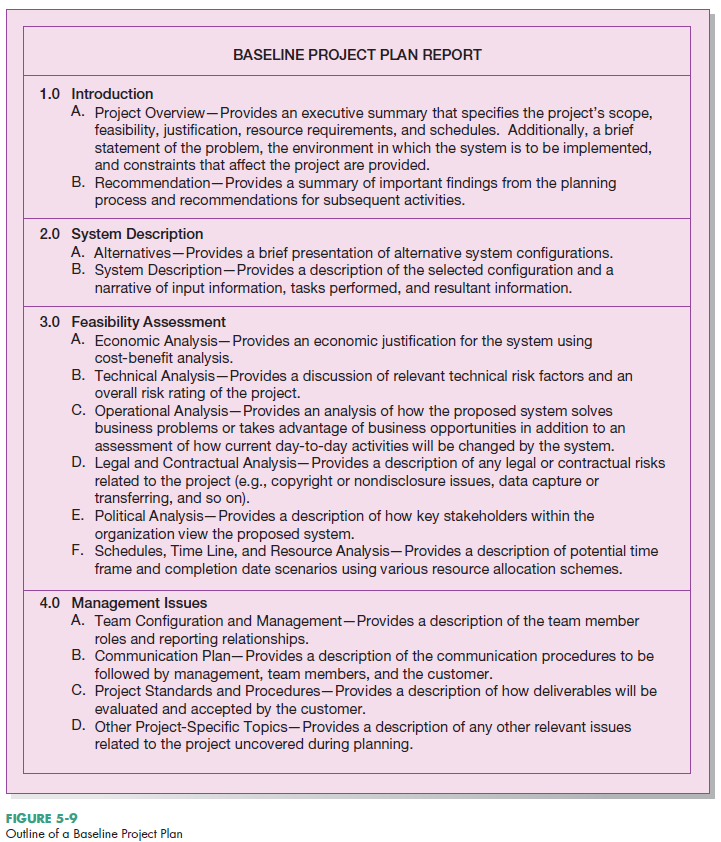
1. Введение

2. Описание системы

3. Технико-экономическое обоснование

4. Вопросы управления

Вводный раздел базового плана проекта Цель введения состоит в том, чтобы дать краткий обзор всего документа и наметить рекомендуемый курс действий для проекта. Весь раздел «Введение» часто ограничивается всего несколькими страницами. Несмотря на то, что раздел «Введение» является первым разделом BPP, он часто является последним разделом, который необходимо написать. Только после выполнения большинства мероприятий по планированию проекта можно создать четкий обзор и рекомендации. Одним из видов деятельности, который должен быть выполнен на начальном этапе, является определение объема проекта.



При определении объема CTS в PVF Джиму Ву сначала нужно было получить четкое представление о целях проекта. Для этого Джим кратко взял интервью у Джеки Джадсон и нескольких ее коллег, чтобы получить четкое представление об их потребностях. Он также потратил несколько часов на изучение функциональности существующей системы, процессов и требований к использованию данных для выполнения действий по отслеживанию клиентов. Эти мероприятия предоставили ему информацию, необходимую для определения масштабов проекта и определения возможных альтернативных решений. Альтернативные системные решения могут относиться к различным областям системы, платформам для развертывания или подходам к приобретению системы. Мы развиваем идею альтернативных решений, называемых стратегиями проектирования, когда обсуждаем фазу анализа жизненного цикла. Во время инициации и планирования проекта наиболее важным элементом стратегии проектирования является область применения системы. Таким образом, определение области применения будет зависеть от следующих факторов:

• Какие организационные подразделения (оперативные функции и отделы) могут быть затронуты предлагаемой системой или изменениями в системе или использовать их?

• С какими существующими системами предлагаемая система должна взаимодействовать или быть согласованной, или какие существующие системы могут быть изменены в связи с заменой системы?

• Кто внутри и за пределами запрашивающей организации (или организации в целом) может быть заинтересован в предлагаемой системе?

• Какой диапазон потенциальных возможностей системы будет рассмотрен?

Заявление об объеме проекта CTS показано на рисунке 5-10.

Для CTS объем проекта был определен с использованием только текстовой информации. Однако нередко определение объема проекта осуществляется с помощью таких диаграмм, как диаграммы потоков данных и модели сущностей-связей. Например, на рисунке 5-11 показана диаграмма потоков данных контекстного уровня, используемая для определения области применения системы для системы выполнения закупок PVF. Другие пункты в разделе «Введение» БПП являются просто резюме других разделов документа.

Второй раздел BPP — это описание системы, которое содержит краткое описание возможных альтернативных решений в дополнение к тому, которое считается наиболее подходящим для данной ситуации. Обратите внимание, что это описание находится на очень высоком уровне и в основном повествовательное по форме. Следующие примеры показывают, что альтернативы могут быть сформулированы просто:

1. Веб-система онлайн

2. Мэйнфрейм с центральной базой данных

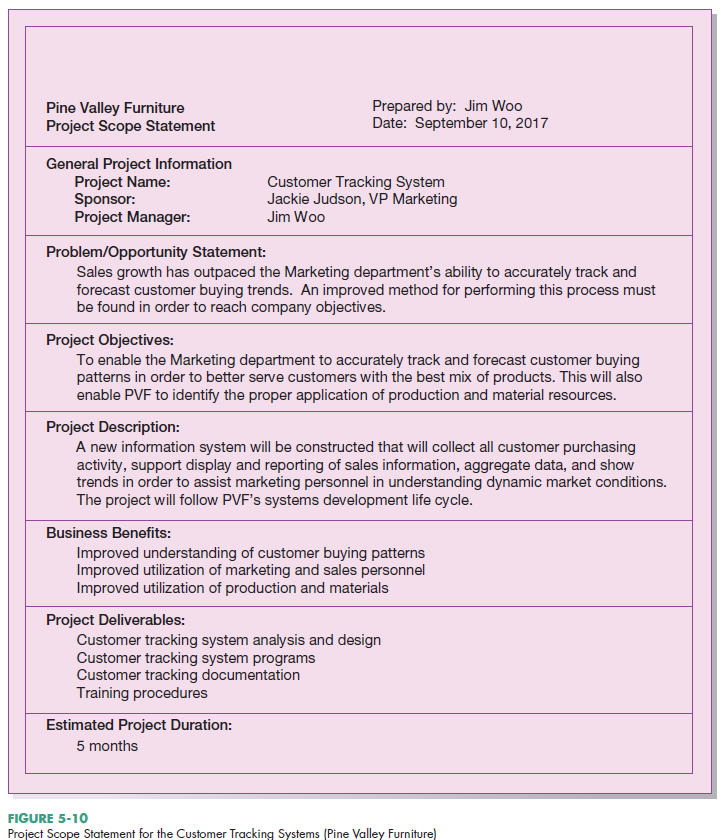
3. Локальная вычислительная сеть с децентрализованными базами данных

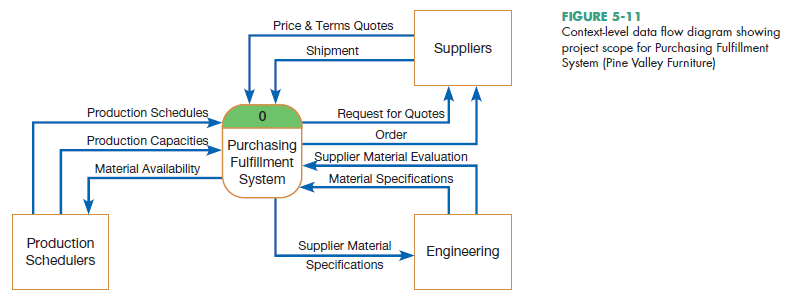
4. Пакетный ввод данных с онлайн-поиском

5. Покупка готового пакета

Если проект одобрен для строительства или покупки, вам нужно будет собрать и структурировать информацию более подробно и строго на этапе анализа и более глубоко оценить эти и другие альтернативные направления для системы. На данном этапе проекта ваша цель состоит только в том, чтобы определить наиболее очевидные альтернативные решения.

Когда Джим и Джеки рассматривали системные альтернативы для CTS, они сосредоточились на двух основных вопросах. Во-первых, они обсудили, как будет приобретена система, и рассмотрели три варианта: приобрести систему, если удастся найти систему, отвечающую потребностям PVF, передать разработку системы на аутсорсинг сторонней организации или построить систему в рамках PVF. Второй вопрос был посвящен определению полноты функциональных возможностей системы. Чтобы выполнить эту задачу, Джим попросил Джеки написать серию заявлений с перечислением типов задач, которые, по ее мнению, маркетинговый персонал сможет выполнить при использовании CTS. Этот список заявлений стал основой описания системы и сыграл важную роль в принятии решения о приобретении. Рассмотрев уникальные потребности маркетинговой группы, оба решили, что лучшим решением было бы построить систему в рамках PVF.

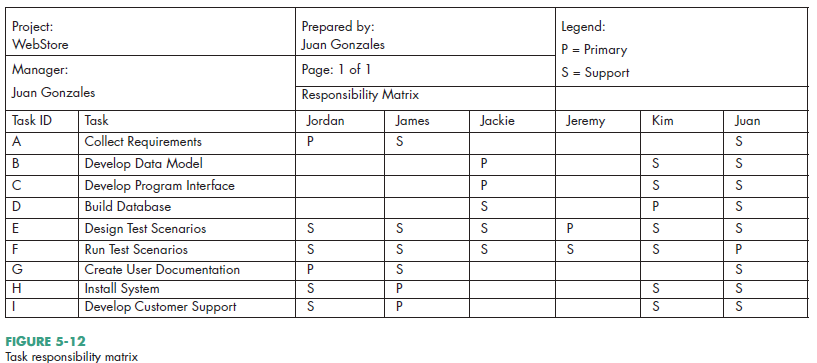




В третьем разделе «Технико-экономическое обоснование» изложены вопросы, связанные с затратами и выгодами проекта, техническими трудностями и другими подобными проблемами. Это также раздел, в котором высокоуровневые графики проектов указываются с помощью сетевых диаграмм и диаграмм Ганта. Вспомним из главы 3, что этот процесс называется структурой декомпозиции работ. Во время инициации и планирования проекта оценки задач и мероприятий, как правило, не детализируются. Точная разбивка работ может быть сделана только для следующих одного или двух видов деятельности жизненного цикла. После определения основных задач проекта можно сделать оценку потребностей в ресурсах. Как и в случае с определением задач и видов деятельности, эта деятельность в первую очередь связана с получением приблизительных оценок потребностей в людских ресурсах, поскольку люди являются наиболее дорогостоящим ресурсным элементом. После того, как вы определите основные задачи и потребности в ресурсах, можно разработать предварительный график. Для определения приемлемого графика может потребоваться найти дополнительные или другие ресурсы или изменить объем проекта. Наибольший объем усилий по планированию проекта, как правило, затрачивается на эти мероприятия по технико-экономическому обоснованию.

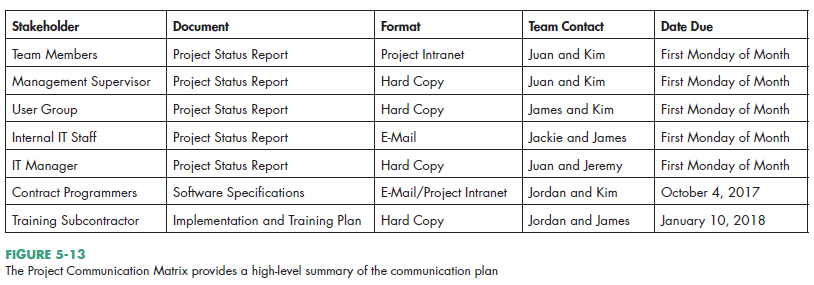
В третьем разделе «Технико-экономическое обоснование» изложены вопросы, связанные с затратами и выгодами проекта, техническими трудностями и другими подобными проблемами. Это также раздел, в котором высокоуровневые графики проектов указываются с помощью сетевых диаграмм и диаграмм Ганта. Вспомним из главы 3, что этот процесс называется структурой декомпозиции работ. Во время инициации и планирования проекта оценки задач и мероприятий, как правило, не детализируются. Точная разбивка работ может быть сделана только для следующих одного или двух видов деятельности жизненного цикла. После определения основных задач проекта можно сделать оценку потребностей в ресурсах. Как и в случае с определением задач и видов деятельности, эта деятельность в первую очередь связана с получением приблизительных оценок потребностей в людских ресурсах, поскольку люди являются наиболее дорогостоящим ресурсным элементом. После того, как вы определите основные задачи и потребности в ресурсах, можно разработать предварительный график. Для определения приемлемого графика может потребоваться найти дополнительные или другие ресурсы или изменить объем проекта. Наибольший объем усилий по планированию проекта, как правило, затрачивается на эти мероприятия по технико-экономическому обоснованию.

Теперь вы должны иметь представление о том, как строится BPP и какие типы информации он содержит. Его создание не должно быть проектом само по себе, а скорее шагом в общем процессе разработки систем. Развитие БПП преследует две основные цели. Во-первых, это помогает гарантировать, что заказчик и группа разработчиков разделяют общее понимание проекта. Во-вторых, это помогает предоставить спонсирующей организации четкое представление о масштабах, преимуществах и продолжительности проекта.



#### ПЕРЕСМОТР БАЗОВОГО ПЛАНА ПРОЕКТА

Прежде чем можно будет приступить к следующему этапу SDLC, пользователи, руководство и группа разработчиков должны просмотреть BPP, чтобы убедиться, что он имеет смысл. Этот обзор проводится до того, как BPP будет представлен или представлен органу по утверждению проекта, такому как руководящий комитет IS или лицо, которое должно финансировать проект. Цель этого обзора состоит в том, чтобы убедиться в том, что предлагаемая система соответствует организационным стандартам и что все соответствующие стороны понимают и соглашаются с информацией, содержащейся в БПП. Распространенный метод выполнения этой проверки (а также проверок на последующих этапах жизненного цикла) называется структурированным пошаговым руководством. Пошаговые руководства представляют собой экспертные групповые обзоры любого продукта, созданного в процессе разработки систем, и широко используются профессиональными организациями по разработке. Опыт показывает, что пошаговые инструкции являются очень эффективным способом обеспечения качества информационной системы и стали обычной повседневной деятельностью для многих системных аналитиков. Большинство прохождений не являются строго формальными или чрезмерно длинными по продолжительности. Однако важно составить конкретную повестку дня для прохождения, чтобы все участники понимали, что должно быть охвачено, и ожидаемое время завершения. На проходных встречах необходимо, чтобы отдельные лица играли определенные роли. Эти роли заключаются в следующем (Yourdon, 1989):



•Согласующий. Этот человек планирует встречу и способствует плавному процессу встречи. Этот человек может быть руководителем проекта или ведущим аналитиком, ответственным за текущий этап жизненного цикла.

•Ведущий. Этот человек описывает рабочий продукт группе. Докладчик обычно является аналитиком, который выполнил всю или часть представленной работы.

•Пользователь. Этот человек (или группа) следит за тем, чтобы рабочий продукт соответствовал потребностям клиентов проекта. Этот пользователь обычно не входит в команду проекта.

•Секретарь. Этот человек делает заметки и записывает решения или рекомендации, принятые группой. Это может быть клерк, назначенный в проектную группу, или один из аналитиков в команде.

• Знаменосец. Роль этого человека заключается в том, чтобы гарантировать, что рабочий продукт соответствует организационным техническим стандартам. Во многих крупных организациях есть группы сотрудников в подразделении, ответственные за разработку стандартных процедур, методов и форматов документации. Эти носители стандартов подтверждают работу, чтобы она могла быть использована другими в организации-разработчике.

• Оракул сопровождения. Этот человек рассматривает рабочий продукт с точки зрения будущих действий по техническому обслуживанию. Цель состоит в том, чтобы сделать систему и ее документацию простыми в обслуживании.

После того, как Джим и Джеки завершили свой BPP для CTS, Джим подошел к своему боссу и попросил назначить пошаговую встречу и назначить координатора для проекта. PVF помогает координатору, предоставляя пошаговую форму обзора, показанную на рисунке 5-14. Используя эту форму, координатор может легко убедиться, что на каждую роль пошагового руководства назначен квалифицированный специалист; что каждому члену была предоставлена копия материалов обзора; и что каждый член знает повестку дня, дату, время и место проведения собрания. На встрече Джим представил BPP, а Джеки добавила комментарии с точки зрения пользователя. После того, как пошаговая презентация была завершена, координатор опросил каждого представителя на предмет его или ее рекомендаций относительно результатов работы. Результаты этого голосования могут привести к утверждению рабочего продукта, утверждению ожидающих изменений, предложенных на собрании, или предположению о том, что рабочий продукт требует существенного пересмотра перед представлением на утверждение. В этом последнем случае обычно запрашиваются существенные изменения в рабочем продукте, после чего необходимо запланировать еще одно прохождение, прежде чем проект может быть предложен Совету по системным приоритетам (руководящему комитету). Что касается КТС, то БПП был поддержан сквозной комиссией в ожидании некоторых незначительных изменений в оценках продолжительности в графике. Эти предложенные изменения были занесены секретарем в Пошаговый список действий (см. Рисунок 5-15) и переданы Джиму для включения в окончательную версию базового плана, который будет представлен руководящему комитету.

Как было сказано в предыдущем обсуждении, пошаговые совещания являются обычным явлением в большинстве групп разработчиков систем и могут использоваться для большего количества действий, чем проверка BPP, включая следующие:

• Технические характеристики системы

• Логический и физический дизайн

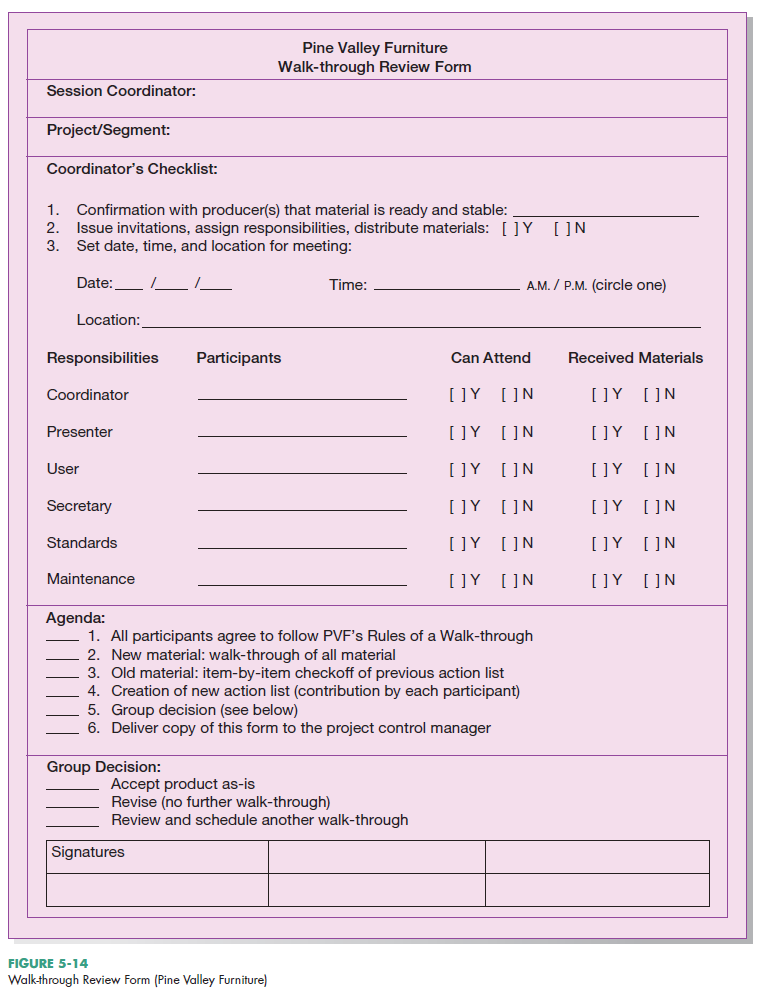
• Сегменты кода или программы

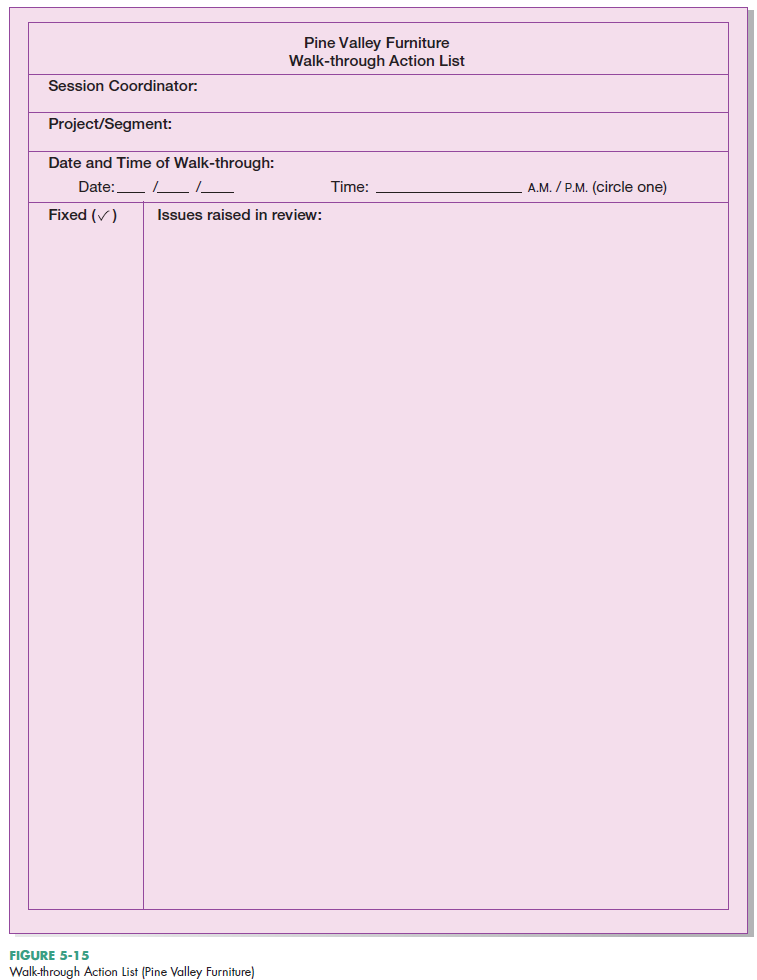
• Процедуры и результаты испытаний

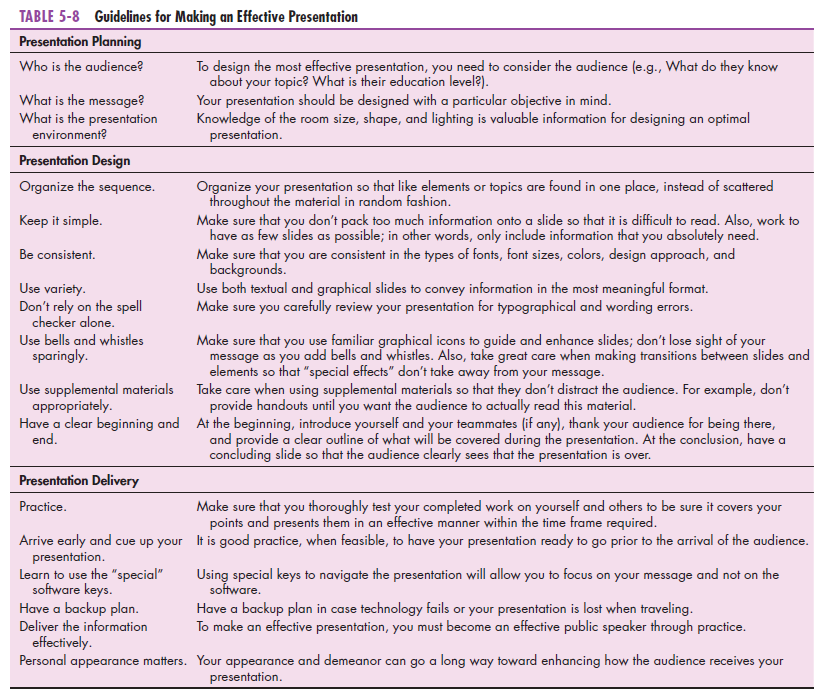
• Руководства и документация

Одним из ключевых преимуществ использования структурированного процесса проверки является то, что он гарантирует, что формальные точки проверки происходят во время проекта. На каждом последующем этапе проекта должна проводиться формальная проверка (и указываться в графике проекта), чтобы убедиться, что все аспекты проекта удовлетворительно выполнены, прежде чем выделять дополнительные ресурсы для проекта. Такой консервативный подход, предусматривающий проведение обзора каждой крупной деятельности по проекту, продолжение которого зависит от успешного завершения предыдущего этапа, называется поэтапным принятием обязательств. При использовании этого подхода гораздо проще остановить или перенаправить проект в любой момент.

Пошаговые инструкции используются на протяжении всего проекта для инструктажа членов команды и внешних заинтересованных сторон. Эти презентации могут принести много пользы команде, но, к сожалению, часто не очень хорошо сделаны. С распространением компьютерных технологий и наличием мощного программного обеспечения, помогающего в разработке и проведении презентаций, создание эффективной презентации никогда не было таким простым. Microsoft PowerPoint стал стандартом де-факто для создания компьютерных презентаций. Хотя эта программа относительно проста в использовании, ею также можно злоупотреблять, так что «навороты», добавленные к компьютерной презентации, фактически отвлекают от презентации. Как и любой проект, чтобы сделать эффективную презентацию, он должен быть хорошо спланирован, хорошо продуман и хорошо поставлен. Планирование и оформление презентации так же важны, как и ее проведение. Если ваши слайды плохо оформлены, плохо читаются или непоследовательны, не имеет значения, насколько хороша ваша подача; Ваша аудитория будет больше думать о низком качестве слайдов, чем о том, что вы говорите. К счастью, приложив немного усилий, легко создать качественную презентацию, если выполнить несколько простых шагов, которые изложены в таблице 5-8.







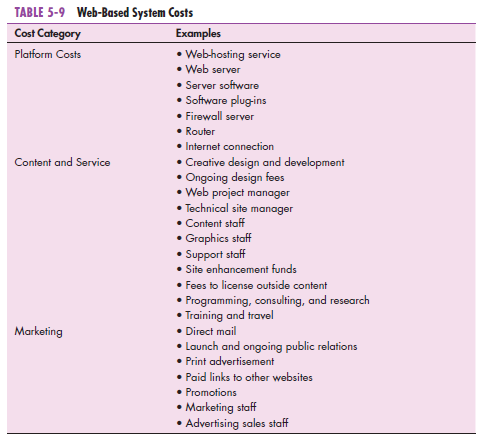
### *ПРИКЛАДНЫЕ ПРОГРАММЫ ЭЛЕКТРОННОЙ ТОРГОВЛИ: ИНИЦИИРОВАНИЕ И ПЛАНИРОВАНИЕ ПРОЕКТОВ ПО РАЗРАБОТКЕ СИСТЕМ*

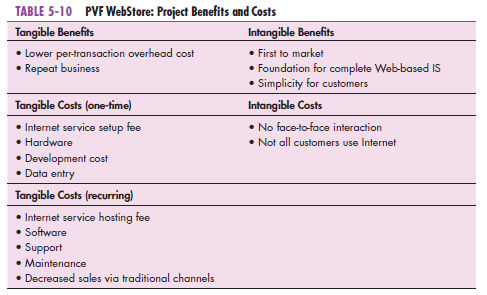
Инициирование и планирование проектов разработки систем для интернет-приложений ЕС очень похоже на процесс, применяемый для более традиционных приложений. В главе 4 вы прочитаете, как руководство PVF начало проект WebStore — продавать мебельную продукцию через Интернет. В этом разделе мы осветим некоторые вопросы, которые имеют непосредственное отношение к процессу определения и отбора проектов разработки систем.

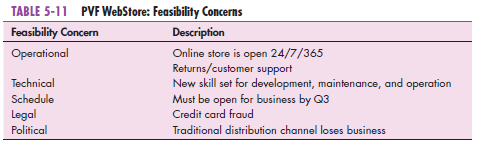
#### Инициирование и планирование проектов разработки систем для интернет-магазина мебели Pine valley

Учитывая высокий приоритет проекта WebStore, для работы над этим проектом были назначены вице-президент по маркетингу Джеки Джадсон и старший системный аналитик Джим Ву (Jim Woo). Как и CTS, описанная ранее в этой главе, их первоначальная деятельность заключалась в том, чтобы начать деятельность по инициированию и планированию проекта.

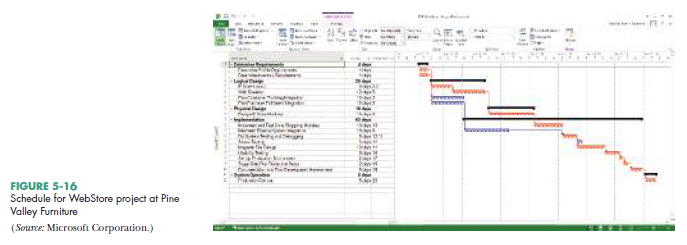
Инициирование и планирование системы электронной коммерции PVF, чтобы начать процесс инициации и планирования, Джим и Джеки провели несколько встреч в течение нескольких дней. На первой встрече они договорились, что «WebStore» будет предложенным названием системного проекта. Затем они работали над выявлением потенциальных выгод, затрат и проблем с осуществимостью. Чтобы помочь в этом процессе, Джим разработал список потенциальных затрат на разработку веб-систем, которым он поделился с Джеки и другими членами проектной группы (см. Таблицу 5-9).







Пошаговое руководство по проекту WebStore После встречи с проектной группой Джим и Джеки составили первоначальный список выгод и затрат (см. Таблицу 5-10), а также несколько вопросов осуществимости (см. Таблицу 5-11). Затем Джим работал с несколькими техническими специалистами PVF, чтобы разработать первоначальный график проекта. На рисунке 5-16 показана диаграмма Ганта для этого 84-дневного графика. Наконец, Джим и Джеки представили свои первоначальные планы проекта совету директоров и высшему руководству PVF. Все были в восторге от плана проекта, и было дано одобрение на перевод проекта WebStore в фазу анализа.



### Резюме

Фаза инициации и планирования проекта (PIP) является критически важным действием в жизни проекта. Именно в этот момент проект принимается к разработке, отклоняется как неосуществимый или перенаправляется. Цель этого процесса состоит в том, чтобы преобразовать расплывчатый системный запрос в осязаемое описание системы, четко излагающее цели, вопросы осуществимости, выгоды, затраты и сроки проекта.

Инициация проекта включает в себя формирование команды инициации проекта, установление отношений с клиентами, разработку плана начала проекта, настройку процедур управления проектом и создание общей среды управления проектом. Ключевым видом деятельности в планировании проекта является оценка многочисленных вопросов осуществимости, связанных с проектом. Типы осуществимости, которые должны быть изучены, включают экономические, технические, эксплуатационные, графические, юридические и договорные, и политические. На эти вопросы влияет размер проекта, тип предлагаемой системы, а также коллективный опыт группы разработчиков и потенциальных заказчиков системы. Высокие затраты и риски проекта не обязательно являются плохими; скорее, более важно, чтобы организация понимала затраты и риски, связанные с проектом и портфелем активных проектов, прежде чем приступить к работе.

После завершения всех анализов может быть создан BPP. BPP включает в себя высокоуровневое описание предлагаемой системы или изменения системы, описание различных технико-экономических возможностей и обзор вопросов управления, характерных для проекта. Прежде чем приступить к разработке информационной системы, пользователи, руководство и группа разработки должны рассмотреть и согласовать эту спецификацию. Основное внимание в этом пошаговом обзоре уделяется оценке достоинств проекта и обеспечению того, чтобы проект, если он будет принят к разработке, соответствовал организационным стандартам и целям. Цель этого процесса также состоит в том, чтобы убедиться в том, что все соответствующие стороны понимают и согласны с информацией, содержащейся в плане, до начала последующей деятельности по развитию.

Инициирование и планирование проекта — сложная и трудоемкая деятельность, требующая активного участия многих участников организации. Конечный успех проекта развития и функционирования информационных систем в целом зависит от эффективного использования дисциплинированных, рациональных подходов, таких как методы, изложенные в этой главе. В последующих главах вы познакомитесь с множеством других инструментов, которые помогут вам стать эффективным проектировщиком и разработчиком информационных систем.

# Часть III. Анализ

## Обзор

Анализ — это первая фаза жизненного цикла разработки системы (SDLC), на которой вы начинаете глубоко понимать необходимость изменений в системе. Системный анализ требует значительных усилий и затрат и поэтому проводится только после того, как руководство решило, что рассматриваемый проект разработки систем имеет смысл и должен быть продолжен на этом этапе. Аналитическая группа не должна воспринимать процесс анализа как нечто само собой разумеющееся или пытаться ускорить его. Большинство наблюдателей согласятся с тем, что многие ошибки в разработанных системах напрямую связаны с недостаточными усилиями на этапах анализа и проектирования жизненного цикла. Поскольку анализ — это большой и сложный процесс, мы разделяем его на два основных действия, чтобы сделать весь процесс более понятным:

1. *Определение требований.* Это, прежде всего, деятельность по установлению фактов.
2. *Структурирование требований.* Эта деятельность создает подробное и четкое описание текущих бизнес-операций и новых услуг по обработке информации

Цель анализа состоит в том, чтобы определить, какая информация и услуги по ее обработке необходимы для поддержки выбранных целей и функций организации. Сбор этой информации называется определением требований, о чем идет речь в главе 6. Методы установления фактов, описанные в главе 6, используются для того, чтобы узнать о текущей системе, организации, которую будет поддерживать заменяющая система, а также о требованиях или ожиданиях пользователей в отношении заменяющей системы.

В главе 6 мы также обсудим основной источник новых систем — реинжиниринг бизнес-процессов (BPR). В отличие от постепенных улучшений, которые лежат в основе многих проектов по разработке систем, BPR приводит к радикальной перестройке процессов, для поддержки которых предназначены информационные системы. Мы покажем, как BPR связан с анализом информационных систем, в главе 7, где мы используем диаграммы потоков данных для поддержки процесса реинжиниринга. В главе 6 вы также узнаете о новых методах определения требований, которые иногда используются в рамках гибких методологий. К ним относятся Planning Game (от eXtreme Programming) и Usage-Centered Design.

Информация о текущих операциях и требованиях к системе замены должна быть организована для анализа и проектирования. Организация или структурирование системных требований приводит к созданию диаграмм и описаний (моделей), которые могут быть проанализированы, чтобы показать недостатки, неэффективность, недостающие элементы и нелогичные компоненты текущих бизнес-операций и информационных систем. Наряду с требованиями пользователя, они используются для определения стратегии замены системы.

Результаты определения требований могут быть структурированы по трем основным видам существующей и заменяемой информационных систем:

1. Логика и тайминг. Правила, по которым данные преобразуются и обрабатываются, а также указание на то, что инициирует преобразование данных.
2. Данные. Неотъемлемая структура данных, не зависящая от того, как и когда они обрабатываются.

Представление процесса системы может быть представлено диаграммами потоков данных, основной темой главы 7. Глава также включает раздел, посвященный таблицам решений — одному из способов описания логики и времени того, что происходит внутри блоков процессов на диаграммах потоков данных. Глава 7 заканчивается четырьмя приложениями. Первые три посвящены одному из трех методов объектно-ориентированной разработки. Первое приложение знакомит вас с моделированием прецедентов — объектно-ориентированным методом, используемым для отображения функциональности системы. Во втором приложении представлены диаграммы активности, а в третьем — диаграммы последовательностей. Эти объектно-ориентированные модели ориентированы на системную логику и синхронизацию. Четвертое приложение посвящено моделированию бизнес-процессов, которое не является частью объектно-ориентированного подхода. Наконец, представление данных системы, обсуждаемое в главе 8, показывает правила, управляющие структурой и целостностью данных, и концентрируется на том, к каким данным о бизнес-объектах и отношениях между этими сущностями должен быть получен доступ в системе. В главе 8 представлены методы определения отношений сущностей в основной части главы и методы построения диаграмм классов для моделирования данных в специальном объектно-ориентированном разделе в конце главы. За главами 7 и 8 следуют статьи Петри, иллюстрирующие процессы, логику и модели данных, описывающие новую систему. Эти примеры также иллюстрируют, как диаграммы и модели для каждого из этих трех представлений системы связаны друг с другом, чтобы сформировать последовательное и тщательное структурированное описание предлагаемой системы.

## Часть 6. Разделение требований к системе

### Введение

Системный анализ — это часть жизненного цикла разработки системы, в ходе которой вы определяете, как функционирует текущая информационная система, и оцениваете, что пользователи хотели бы видеть в новой системе. Анализ состоит из двух подэтапов: определение требований и структурирование требований. В этой главе вы узнаете об определении системных требований. Методы, используемые при определении требований, со временем эволюционировали, чтобы стать более структурированными и все больше полагаться на компьютерную поддержку. Сначала мы изучим более традиционные методы определения требований, включая интервьюирование, наблюдение за пользователями в их рабочей среде, а также сбор процедур и других письменных документов. Затем мы обсудим более современные методы сбора системных требований. Первый из этих методов — Joint Application Design (JAD). Далее вы прочитаете о том, как аналитики все больше и больше полагаются на информационные системы, помогающие им выполнять анализ. Как вы увидите, CASE-инструменты, рассмотренные в главе 1, полезны при определении требований, а создание прототипов стало ключевым инструментом для некоторых усилий по определению требований. Наконец, вы узнаете, как анализ требований продолжает оставаться важной частью системного анализа и проектирования, независимо от того, включает ли этот подход реконструкцию бизнес-процессов, новые методы Agile (такие как постоянное вовлечение пользователей или дизайн, ориентированный на использование) или фокусируется на разработке интернет-приложений.

#### Подготовка к разделению требований

Как было сказано ранее и показано на рисунке 6.1, системный анализ состоит из двух подэтапов: определение требований и структурирование требований. Мы рассмотрим их как отдельные шаги, но вы должны рассматривать их как параллельные и итеративные. Например, когда вы определяете некоторые аспекты текущей и желаемой системы (систем), вы начинаете структурировать эти требования или создавать прототипы, чтобы показать пользователям, как может вести себя система. Несоответствия и недостатки, обнаруженные при структурировании и создании прототипов, приводят к дальнейшему изучению функционирования текущей системы (систем) и будущих потребностей организации. В конечном счете, ваши идеи и открытия объединяются в подробное и точное описание текущих операций и требований к новой системе. Когда вы думаете о начале этапа анализа, вы, вероятно, задаетесь вопросом, что именно включает в себя определение требований.

Мы обсудим этот процесс в следующем разделе.

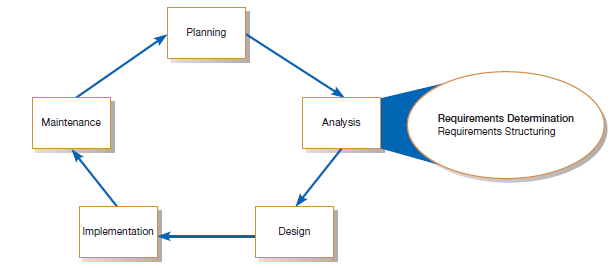


Рисунок 6.1 – Жизненный цикл системной разработки и фаза анализа (подсвечена на рисунке)

##### Процесс разделения требований

После того, как руководство дало разрешение на разработку новой системы (это было сделано в конце этапа определения и выбора проекта SDLC) и проект инициирован и спланирован (см. главу 5), вы начинаете определять, что должна делать новая система. Во время определения требований вы и другие аналитики собираете информацию о том, что должна делать система, из как можно большего количества источников: от пользователей текущей системы; от наблюдающих пользователей; а также из отчетов, форм и процедур. Все системные требования тщательно документированы и подготовлены к структурированию, что является предметом глав 7 и 8.

Во многих отношениях сбор системных требований похож на проведение любого расследования. Читали ли вы какие-нибудь детективные истории о Шерлоке Холмсе или подобные детективы? Вам нравится решать головоломки? Исходя из этого опыта, мы можем обнаружить некоторые схожие характеристики хорошего системного аналитика на подэтапе определения требований. К таким характеристикам относятся следующие:

1. *Дерзость.* Вы должны подвергать все сомнению. Вам нужно задать такие вопросы, как: всё ли транзакции обрабатываются одинаково? Может ли кто-то взимать что-то, кроме стандартной цены? Может быть, когда-нибудь мы захотим разрешить и поощрять сотрудников работать более чем в одном отделе?
2. *Беспристрастность.* Ваша роль заключается в том, чтобы найти наилучшее решение бизнес-проблемы или возможности. Речь идет, например, не о том, чтобы найти способ оправдать покупку нового оборудования или настаивать на включении в новые системные требования того, что, по мнению пользователей, они хотят. Вы должны рассмотреть вопросы, поднятые всеми сторонами, и постараться найти наилучшее организационное решение.
3. *Ослабьте ограничения.* Исходите из того, что все возможно, и исключайте неосуществимое. Например, не принимайте утверждение: «Мы всегда так делали, поэтому мы должны продолжать практику». Традиции отличаются от правил и политики. Традиции, вероятно, возникли по уважительной причине, но по мере того, как организация и окружающая среда меняются, они могут превратиться в привычки, а не в разумные процедуры.
4. *Внимание к деталям.* Каждый факт должен согласовываться со всеми остальными фактами. Один неправильный элемент означает, что даже самая лучшая система в какой-то момент выйдет из строя. Например, неточное определение того, кто является клиентом, может означать, что вы удаляете данные о клиенте, когда у клиента нет активных заказов, но эти прошлые клиенты могут быть жизненно важными контактами для будущих продаж
5. *Рефрейминг.* Анализ – это, в частности, творческий процесс. Вы должны бросить себе вызов, чтобы взглянуть на организацию по-новому. Необходимо учитывать, как каждый пользователь видит свои требования. Вы должны быть осторожны и не делать поспешных выводов: «Однажды я работал над подобной системой — эта новая система должна работать так же, как и та, которую я построил раньше».

##### Конечные результаты и результаты

Основными результатами определения требований являются различные формы информации, собранной в процессе определения: стенограммы собеседований; заметки о наблюдении и анализе документов; комплекты бланков, отчетов, должностных инструкций и других документов; и сгенерированные компьютером выходные данные, такие как прототипы систем. Короче говоря, все, что группа аналитиков собирает в рамках определения требований к системе, включается в конечные результаты, полученные в результате этой подфазы жизненного цикла разработки системы. В таблице 6-1 приведены примеры некоторой специфической информации, которая может быть собрана при определении требований.

Эти результаты содержат информацию, необходимую для системного анализа в рамках разрабатываемой системы. Кроме того, необходимо понимать следующие составляющие организации:

* Бизнес-цели, определяющие, что и как выполняется
* Информация, необходимая людям для выполнения своей работы
* Данные (определение, объем, размер и т. д.), обрабатываемые в организации для поддержки заданий
* Когда, как, кем или чем данные перемещаются, преобразуются и хранятся
* Последовательность и другие зависимости между различными действиями по обработке данных
* Правила, регулирующие обработку и обработку данных
* Политики и руководящие принципы, описывающие характер бизнеса, а также рынок и среду, в которой он функционирует.
* Ключевые события, влияющие на значения данных, и время их возникновения

Как должно быть очевидно, такой большой объем информации должен быть организован для того, чтобы быть полезным. Это и есть цель следующего подэтапа — структурирования требований.

Исходя из этой подфазы анализа, вы, вероятно, уже поняли, что объем информации, которую необходимо собрать, может быть огромным, особенно если область применения разрабатываемой системы широка. Время, необходимое для сбора и структурирования большого количества информации, может быть большим и, поскольку это требует больших человеческих усилий, довольно дорогостоящим. Слишком большой объем анализа непродуктивен, и термин «аналитический паралич» был придуман для описания проекта разработки систем, который увяз в изобилии аналитической работы. Из-за опасностей чрезмерного анализа современные системные аналитики больше сосредотачиваются на разрабатываемой системе, чем на существующей системе. Методы, о которых вы узнаете далее в этой главе, JAD и прототипирование, были разработаны для того, чтобы свести к минимуму усилия по анализу, но при этом сохранить его эффективность. Для обеспечения быстрого и гибкого определения требований также были разработаны новые методы, включая постоянное вовлечение пользователей, проектирование, ориентированное на использование, и игру планирования от eXtreme Programming. Традиционные методы сбора фактов являются темой следующего раздела.

Таблица 6.1 Конечные результаты для определения требований

1. Информация, собранная из разговоров с пользователями или наблюдений за ними: стенограммы интервью, заметки о наблюдениях, протоколы совещаний
2. Существующая письменная информация: заявления о бизнес-миссии и стратегии, образцы бизнес-форм и отчетов, а также компьютерные дисплеи, руководства по процедурам, должностные инструкции, учебные пособия, блок-схемы и документация существующих систем, отчеты консультантов
3. Компьютерная информация: результаты сеансов JAD, содержимое репозитория CASE и отчеты существующих систем, а также дисплеи и отчеты из прототипов систем
4. Индивидуальное собеседование с людьми, информированными о работе и проблемах текущей системы и будущих потребностей в системах
5. Опрашивайте группы людей с различными потребностями, чтобы найти синергию и контраст между системными требованиями
6. Наблюдайте за работниками в выбранное время, чтобы увидеть, как обрабатываются данные и какая информация нужна людям для выполнения своей работы.
7. Изучайте бизнес-документы, чтобы обнаружить проблемы, политики, правила и указания, о которых сообщалось, а также конкретные примеры использования данных и информации в организации.

### Традиционные методы разделения требований

В основе системного анализа лежит сбор информации. На начальном этапе необходимо собрать информацию об информационных системах, которые используются в настоящее время, и о том, как пользователи хотели бы улучшить существующие системы и организационные операции с помощью новых или заменяющих информационных систем. Один из лучших способов получить эту информацию — поговорить с людьми, которые прямо или косвенно вовлечены в различные части организации, затронутые возможными системными изменениями: пользователями, менеджерами, спонсорами и так далее. Еще один способ узнать о текущей системе — собрать копии документации, относящейся к текущим системам и бизнес-процессам. В этой главе вы узнаете о различных способах получения информации непосредственно от заинтересованных сторон: интервью, групповые интервью, метод номинальной группы и непосредственное наблюдение. Вы узнаете о сборе документации по текущей работе системы и организации в виде письменных процедур, форм, отчетов и других бумажных копий. Эти традиционные методы сбора системных требований перечислены в таблице 6-2.

#### Интервьюирование и слушание

Интервьюирование является одним из основных способов сбора аналитиками информации о проекте информационных систем. В начале проекта аналитик может потратить много времени на опрос людей об их работе, информации, которую они используют для ее выполнения, и типах обработки информации, которые могут дополнить их работу. Опрашиваются другие заинтересованные стороны, чтобы понять организационное направление, политику, ожидания менеджеров в отношении подразделений, которыми они руководят, и другие нестандартные аспекты организационной деятельности. Во время интервью вы будете собирать факты, мнения и предположения, а также наблюдать за языком тела, эмоциями и другими признаками того, чего хотят люди и как они оценивают существующие системы.

Таблица 6-3. Рекомендации по эффективному проведению собеседований

Запланируйте собеседование

• Подготовка интервьюируемого: назначение, вопросы по праймингу

• Подготовьте контрольный список, повестку дня и вопросы

Внимательно слушайте и делайте заметки

(запись, если разрешено)

Просматривайте заметки в течение 48 часов

интервью

Будьте нейтральны

Поиск разнообразных мнений

Есть много способов эффективно взять у кого-то интервью, и ни один из них не обязательно лучше другого. Некоторые рекомендации, которые вы должны иметь в виду во время собеседования, обобщенные в таблице 6-3, обсуждаются ниже.

Во-первых, следует основательно подготовиться к собеседованию. Назначьте встречу в удобное для собеседника время и на удобное для него время. Общий характер интервью должен быть заранее разъяснен интервьюируемому. Вы можете попросить интервьюируемого подумать над конкретными вопросами или проблемами или просмотреть определенную документацию, чтобы подготовиться к собеседованию. Вам следует потратить некоторое время на обдумывание того, что вам нужно выяснить, и записать свои вопросы. Не думайте, что вы можете предугадать все возможные вопросы. Вы хотите, чтобы собеседование было естественным, и, в какой-то степени, вы хотите спонтанно направлять интервью, когда вы узнаете, какой опыт интервьюируемый привносит в сессию.

Вы должны подготовить руководство по собеседованию или контрольный список, чтобы знать, в какой последовательности вы намерены задавать свои вопросы и сколько времени вы хотите потратить на каждую область собеседования. Контрольный список может включать в себя несколько наводящих вопросов, которые следует задать в качестве последующих действий, если вы получите определенные ожидаемые ответы. Вы можете, в некоторой степени, интегрировать свое руководство по собеседованию с заметками, которые вы делаете во время собеседования, как показано в образце руководства на рисунке 6-2. Это же руководство может служить кратким изложением того, что вы обнаружите во время собеседования.

Первая страница примера руководства по проведению собеседования содержит общий план интервью. Помимо основной информации о том, с кем и когда проводится собеседование, вы перечисляете основные цели собеседования. Эти цели, как правило, охватывают наиболее важные данные, которые необходимо собрать, список вопросов, по которым необходимо прийти к согласию (например, содержание определенных системных отчетов), и области, которые необходимо исследовать, не обязательно с помощью конкретных вопросов. Вы также включаете напоминания для себя о ключевой информации о интервьюируемом (например, история работы, известные должности, занимаемые по вопросам, и роль в текущей системе). Эта информация поможет вам перейти на личности, покажет, что вы считаете интервьюируемого важным, и может помочь вам интерпретировать некоторые ответы. Также включена повестка дня собеседования с примерными временными рамками для разных разделов интервью. Возможно, вы не соблюдаете в точности временные рамки, но график поможет вам охватить все области за то время, пока интервьюируемый доступен. Место также отводится для общих замечаний, которые не вписываются в конкретные вопросы, и для заметок, сделанных во время интервью, о пропущенных темах или поднятых проблемах, которые не удалось решить.

На последующих страницах вы перечисляете конкретные вопросы; Образец формы на рисунке 6-2 содержит место для заметок по этим вопросам. Из-за того, что возникает непредвиденная информация, вы не будете строго следовать руководству по порядку. Тем не менее, вы можете отметить вопросы, которые вы задали, и написать напоминания себе, чтобы вернуться к определенным вопросам или пропустить их по мере развития динамики собеседования.

**Открытые вопросы** - вопросы на собеседованиях, на которые нет заранее оговоренных ответов

**Закрытые вопросы** - вопросы в интервью, в которых респондентам предлагается выбрать один из заданных ответов.

**Выбор вопросов на интервью** Вам нужно решить, какое сочетание и последовательность открытых и закрытых вопросов вы будете использовать. **Открытые вопросы** обычно используются для поиска информации, на которую вы не можете предугадать все возможные ответы или на которую вы не знаете точного вопроса, который задать. Опрашиваемому предлагается говорить о том, что его интересует, в общих рамках вопроса. Например: «Что, по вашему мнению, является лучшим в информационной системе, которую вы в настоящее время используете для выполнения своей работы?» или «Перечислите три наиболее часто используемых пункта меню». Вы должны быстро реагировать на ответы и определять, нужны ли какие-либо дополнительные вопросы для уточнения или уточнения. Иногда язык тела говорит о том, что пользователь дал неполный ответ или не хочет разглашать какую-либо информацию; Дополнительный вопрос может дать дополнительную информацию. Одно из преимуществ открытых вопросов заключается в том, что на поверхность может всплыть ранее неизвестная информация. Затем вы можете продолжить исследование по неожиданным направлениям, чтобы получить еще больше новой информации. Открытые вопросы также часто успокаивают интервьюируемых, потому что они могут отвечать своими словами, используя свою собственную структуру; Открытые вопросы дают интервьюируемым больше ощущения вовлеченности и контроля над интервью. Основным недостатком открытых вопросов является время, которое может потребоваться для получения ответов на вопросы. Кроме того, открытые вопросы может быть трудно резюмировать.

**Закрытые вопросы** предоставляют ряд ответов, из которых интервьюируемый может выбрать. Вот пример:

*Что из перечисленного ниже вы считаете самым лучшим в информационной системе, которую вы в настоящее время используете для выполнения своей работы (выберите только один)?*

*a. Легкий доступ ко всем необходимым данным*

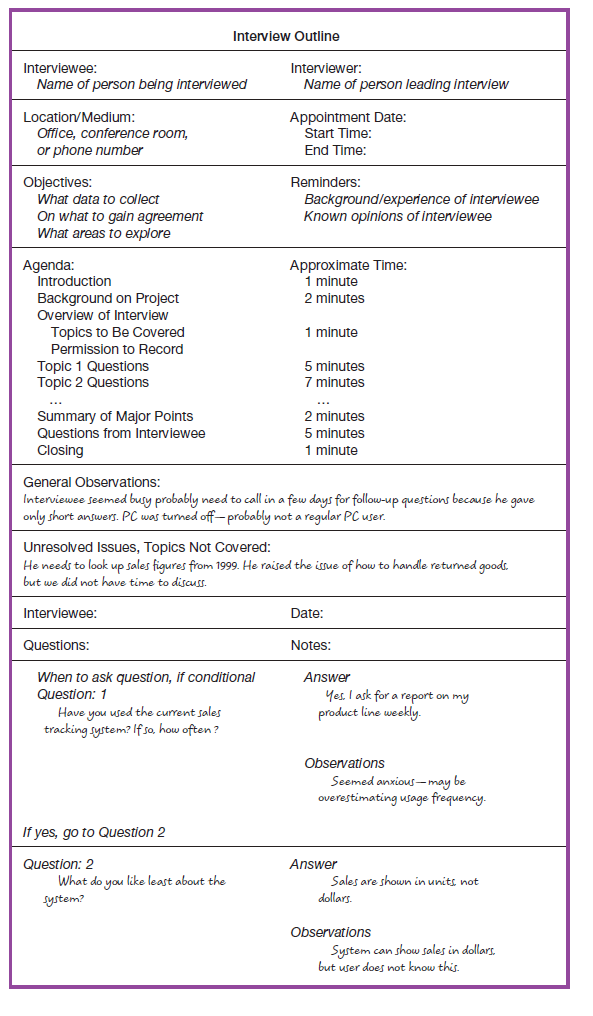
*b. Время отклика системы*

*c. Возможность удаленного доступа к системе*

Закрытые вопросы хорошо работают, когда основные ответы на вопросы хорошо известны. Еще один плюс заключается в том, что собеседования, основанные на закрытых вопросах, не обязательно требуют больших временных затрат — можно охватить больше тем. Вы можете видеть язык тела и слышать тон голоса, что может помочь в интерпретации ответов интервьюируемого. Закрытые вопросы также могут быть простым способом начать собеседование и определить, какую линию открытых вопросов следует использовать. Вы можете включить опцию «Другое», чтобы побудить интервьюируемого добавить непредвиденные ответы. Основной недостаток закрытых вопросов заключается в том, что полезная информация, которая не совсем вписывается в определенные ответы, может быть упущена из виду, поскольку респондент пытается сделать выбор вместо того, чтобы дать свой лучший ответ.

Закрытые вопросы, как и объективные вопросы на экзамене, могут иметь несколько форм, включая следующие варианты:

1. Правда или ложь
2. Множественный выбор (только с одним ответом или с выбором всех подходящих вариантов).
3. Оценивайте ответ или идею по шкале, скажем, от плохого к хорошему или категорически не согласен. Каждая точка на шкале должна иметь четкое и последовательное значение для каждого человека, и обычно в середине шкалы есть нейтральная точка.
4. Ранжирование элементов в порядке важности



**Правила проведения собеседований**

Во-первых, как с открытыми, так и с закрытыми вопросами не формулируйте вопрос так, чтобы подразумевать правильный или неправильный ответ. Респондент должен чувствовать, что он или она может высказать свое истинное мнение и точку зрения и что его или ее идея будет принята во внимание наравне с идеями других. Такие вопросы, как «Должна ли система продолжать предоставлять возможность переопределить значение по умолчанию, даже если большинству пользователей теперь эта функция не нравится?» следует избегать, поскольку такая формулировка предопределяет социально приемлемый ответ.

Второе правило, которое следует помнить при проведении собеседований, — очень внимательно слушать то, что говорят. Делайте тщательные записи или, если возможно, записывайте интервью (обязательно сначала спросите разрешения!). Ответы могут содержать чрезвычайно важную информацию для проекта. Кроме того, это может быть ваш единственный шанс получить информацию от этого конкретного человека. Если у вас мало времени, а вам все еще нужно получить информацию от человека, с которым вы разговариваете, попросите назначить последующее собеседование.

В-третьих, после окончания собеседования вернитесь в свой офис и напечатайте свои записи в течение 48 часов. Если вы записали интервью, используйте запись для проверки материала в своих заметках. Через 48 часов ваши воспоминания об интервью быстро исчезнут. По мере того, как вы печатаете и систематизируете свои заметки, записывайте любые дополнительные вопросы, которые могут возникнуть из-за ошибок в ваших заметках или неоднозначной информации. Отделяйте факты от своих мнений и интерпретаций. Составьте список неясных моментов, которые нуждаются в разъяснении. Позвоните человеку, у которого вы брали интервью, и получите ответы на эти новые вопросы. Используйте телефонный звонок как возможность проверить точность своих записей. Вы также можете отправить письменную копию своих записей человеку, у которого вы проводили собеседование, чтобы он мог проверить точность ваших записей. Наконец, обязательно поблагодарите человека за уделенное ему время. Возможно, вам придется еще раз поговорить с вашим респондентом. Если собеседник будет пользователем вашей системы или иным образом причастен к ее успеху, вы хотите произвести хорошее впечатление.

В-четвертых, будьте осторожны во время собеседования и не высказывайте ожиданий относительно новой или заменяющей системы, если вы не уверены, что эти функции будут частью поставляемой системы. Дайте интервьюируемому понять, что проект состоит из множества этапов и необходимо учитывать мнения многих людей, а также то, что технически возможно. Дайте респондентам знать, что их идеи будут тщательно рассмотрены, но из-за итеративного характера процесса разработки систем сейчас преждевременно говорить точно, что будет или не будет делать конечная система.

В-пятых, ищите в интервью различные точки зрения. Узнайте, что потенциальные пользователи системы, пользователи других систем, на которых могут повлиять изменения, менеджеры и руководители, сотрудники информационных систем, имеющие опыт работы с текущей системой, и другие думают о текущих проблемах и возможностях и какие новые информационные услуги могут возникнуть. лучше служить организации. Вы хотите понять все возможные перспективы, чтобы на более позднем этапе утверждения у вас была информация, на которой можно основывать рекомендации или проектные решения, которые смогут принять все заинтересованные стороны.

#### Интервьюирование групп

Одним из недостатков использования интервью для сбора системных требований является необходимость для аналитика согласовывать очевидные противоречия в собранной информации. Серия интервью может дать противоречивую информацию о существующей системе или ее замене. Вы должны проработать все эти несоответствия, чтобы выяснить, что может быть наиболее точным представлением текущих и будущих систем. Такой процесс требует нескольких последующих телефонных звонков и дополнительных собеседований. Поймать важных людей в их офисах зачастую сложно и неприятно, а планирование новых собеседований может занять очень много времени. Кроме того, новые интервью могут выявить новые вопросы, которые, в свою очередь, потребуют дополнительных интервью с теми, кто давал интервью ранее. Очевидно, что сбор информации об информационной системе посредством серии индивидуальных интервью и последующих звонков не является эффективным процессом.

Еще один доступный вам вариант — групповое собеседование. При групповом интервью опрашиваются сразу несколько ключевых людей. Чтобы убедиться, что вся важная информация собрана, вы можете провести интервью с одним или несколькими аналитиками. В случае нескольких интервьюеров один аналитик может задавать вопросы, в то время как другой делает записи, или разные аналитики могут сосредоточиться на разных видах информации. Например, один аналитик может отслеживать требования к данным, в то время как другой отмечает время и начало ключевых событий. Число интервьюируемых, участвующих в процессе, может варьироваться от двух до любого количества, которое, по вашему мнению, может разместиться с комфортом. время, чем серия интервью с отдельными людьми (хотя временные затраты интервьюируемых могут вызывать большее беспокойство). Во-вторых, совместное собеседование с несколькими людьми позволяет им услышать мнения других ключевых людей и дает им возможность согласиться или не согласиться со своими коллегами. Синергия также часто имеет место. Например, комментарии одного человека могут заставить другого сказать: «Это мне напоминает» или «Я не знал, что это проблема». Вы можете получить пользу от такого обсуждения, поскольку оно поможет вам выявить проблемы, по которым существует общее согласие, и области, в которых мнения сильно расходятся.

Основным недостатком группового собеседования является сложность его планирования. Чем больше людей будет участвовать, тем сложнее будет найти удобное время и место для всех. Современные технологии видеоконференций могут свести к минимуму факторы географической разбросанности, которые так затрудняют планирование встреч. Групповые интервью лежат в основе процесса JAD, о котором мы поговорим в следующем разделе этой главы. Далее обсуждается конкретная техника работы с группами — техника номинальной группы.

**Техника номинальной группы -** *Облегченный процесс, поддерживающий генерацию идей группами. В начале процесса члены группы работают в одиночку, генерируя идеи. Затем идеи объединяются под руководством обученного координатора.*

За прошедшие годы было разработано множество различных методик для улучшения процесса работы с группами. Один из наиболее популярных методов генерации идей среди членов группы называется методом номинальной группы (NGT). NGT — это именно то, на что указывает название: люди, работающие вместе над решением проблемы, представляют собой группу только по названию или номинально. Члены группы могут собираться в одной комнате на NGT, но какое-то время все они работают в одиночку. Обычно члены группы составляют письменный список своих идей. По окончании времени генерации идей члены группы объединяют свои индивидуальные идеи под руководством обученного ведущего. Объединение обычно предполагает, что ведущий по очереди спрашивает каждого человека об идее, которая ранее не была представлена. Пока человек читает идею вслух, кто-то другой записывает ее на доске или флипчарте. После того, как все идеи будут представлены, ведущий попросит группу открыто обсудить каждую идею, в первую очередь для разъяснения.

Как только все идеи будут понятны всем участникам, ведущий постарается сократить количество идей, которые группа будет выдвигать на дополнительное рассмотрение. Есть много способов уменьшить количество идей. Фасилитатор может попросить участников выбрать только ту часть идей, которые, по их мнению, важны. Затем ведущий обходит комнату, прося каждого человека прочитать вслух важную для него идею, которая еще не была выявлена кем-то другим. Либо ведущий может работать с группой над выявлением и устранением или объединением идей, очень похожих на другие. В какой-то момент ведущий и группа получают понятный набор идей, которым можно дополнительно расставить приоритеты.

В контексте определения требований идеи, которые ищутся в ходе NGT, обычно применимы к проблемам существующей системы или идеям новых функций в разрабатываемой системе. Конечным результатом будет список проблем или функций, которые сами члены группы создали и расставили по приоритетам. Должен быть высокий уровень владения таким списком, по крайней мере, для группы, принимавшей участие в учениях NGT.

Существуют некоторые свидетельства в пользу использования NGT для помощи в фокусировке и совершенствовании работы группы, поскольку количество и качество идей, возникающих в результате NGT, могут быть выше, чем то, что обычно можно получить на групповом собрании без сопровождения. Упражнение NGT можно использовать в качестве дополнения к работе, выполняемой в ходе типичного группового собеседования или в рамках совместной разработки приложения, более подробно описанной далее в этой главе.

#### Непосредственное наблюдение за пользователями

Все методы сбора информации, которые мы обсуждали до сих пор, предполагают, что люди вспоминают и передают имеющуюся у них информацию об организационной сфере и информационных системах, которые поддерживают эти процессы. Однако люди не всегда являются очень надежными информаторами, даже когда они стараются быть надежными и говорить то, что, по их мнению, является правдой. Как бы странно это ни звучало, люди часто не имеют абсолютно точного представления о том, что они делают и как они это делают. Это особенно верно в отношении редких событий, проблем из прошлого или вопросов, к которым люди испытывают сильную страсть. Поскольку людям не всегда можно доверять в том, что они достоверно интерпретируют и сообщают о своих действиях, вы можете дополнить и подтвердить то, что говорят вам люди, наблюдая за тем, что они делают, или получая относительно объективные показатели того, как люди ведут себя в рабочих ситуациях. (См. рамку «Потерянные продажи безалкогольных напитков», где приведен пример того, как важно, чтобы системные аналитики узнавали из первых рук о бизнесе, для которого они разрабатывают системы.)

Например, один из возможных взглядов на то, как гипотетический менеджер выполняет свою работу, состоит в том, что менеджер тщательно планирует свою деятельность, работает в течение длительных периодов времени и последовательно над решением проблем и контролирует темп своей работы. Менеджер может сказать вам, как она проводит свой день. Однако, когда Минцберг (1973) наблюдал за тем, как работают менеджеры, он обнаружил, что рабочий день менеджера на самом деле прерывается множеством перерывов. Менеджеры работают фрагментарно, сосредотачиваясь на проблеме или общении лишь на короткое время, прежде чем их прерывают телефонные звонки или визиты своих подчиненных и других менеджеров. Информационная система, спроектированная так, чтобы соответствовать рабочей среде, описанной нашим гипотетическим менеджером, не будет эффективно поддерживать реальную рабочую среду, в которой находится этот менеджер.

В качестве другого примера рассмотрим разницу между тем, что другой сотрудник может рассказать вам о том, как часто он пользуется электронной почтой, и тем, как часто вы пользуетесь электронной почтой, которую вы можете обнаружить более объективными методами. Сотрудник может сказать вам, что он завален сообщениями электронной почты и тратит значительную часть своего времени на ответы на электронные письма. Однако, если бы вы смогли проверить записи электронной почты, вы могли бы обнаружить, что этот сотрудник получает в среднем только 3 сообщения электронной почты в день, и что наибольшее количество сообщений, которые он когда-либо получал за один восьмичасовой период, составляет 10. В этом случае. вы смогли получить точную поведенческую оценку того, с каким объемом электронной почты справляется этот сотрудник, не наблюдая, как он читает свою электронную почту.

Однако цель получения системных записей и прямого наблюдения одна и та же: получить больше информации из первых рук и объективных показателей взаимодействия сотрудников с информационными системами. В некоторых случаях поведенческие показатели будут более точным отражением реальности, чем думают сотрудники. В других случаях поведенческая информация подтвердит то, что сотрудники сказали вам напрямую. Хотя наблюдение и получение объективных показателей являются желательными способами сбора соответствующей информации, такие методы не всегда возможны в реальных организационных условиях. Таким образом, эти методы не являются полностью беспристрастными, как и ни один другой метод сбора данных не является беспристрастным.

Например, наблюдение может заставить людей изменить свое обычное рабочее поведение. Сотрудники, которые знают, что за ними наблюдают, могут нервничать и совершать больше ошибок, чем обычно, могут стараться следовать точным процедурам, которым они обычно не следуют, и могут работать быстрее или медленнее, чем обычно. Более того, поскольку наблюдение обычно не может быть непрерывным, вы получаете только моментальный снимок человека или задачи, которую вы наблюдаете, который может не включать важные события или действия. Поскольку наблюдение требует очень много времени, вы будете наблюдать не только в течение ограниченного времени, но и за ограниченным количеством людей и ограниченным количеством мест. Опять же, наблюдение дает лишь небольшой сегмент данных из, возможно, огромного разнообразия источников данных. Трудная задача выбора — за какими именно людьми или местами наблюдать. Вы хотите выбрать как типичных, так и нетипичных людей и места и наблюдать в нормальных и аномальных условиях и в разное время, чтобы получить как можно более полные данные в результате наблюдения.

**Lost Soft Drink Sales**

***Системный аналитик был весьма удивлен, узнав, что после установки новой системы маршрутизации грузовиков продажи всех безалкогольных напитков упали, а не выросли. Программное обеспечение было разработано для сокращения дефицита товаров на объектах клиентов, позволяя водителям чаще посещать каждого клиента, используя более эффективные маршруты доставки.***

***Смущенное результатами, руководство попросило аналитика отложить запланированный отпуск, но он настоял на том, что сможет заново взглянуть на систему только после нескольких просроченных дней отдыха и релаксации.***

***Однако вместо того, чтобы взять отпуск, аналитик позвонил диспетчеру доставки, с которым беседовал во время проектирования системы, и попросил указать маршрут на несколько дней. Аналитик проложил маршрут (для обычного водителя, фактически находившегося в отпуске), по графику, разработанному из новой системы. Аналитик обнаружил, что маршрут оказался очень эффективным, как и ожидалось; Итак, поначалу аналитик не видел никаких причин для снижения продаж.***

***В третий и последний день своего «отпуска» аналитик задержался в одном магазине сверхурочно, чтобы спросить менеджера, есть ли у нее какие-либо идеи, почему продажи могли упасть в последние недели.***

***Менеджер не дал никаких объяснений, но сделал, казалось бы, не связанное с этим наблюдение, что у обычного водителя, похоже, было меньше времени, чтобы проводить в магазине. Похоже, его не особо интересовало, где выставлена продукция, и он не просил размещать рекламные вывески, как он часто делал в прошлом.***

***Из этого разговора аналитик пришел к выводу, что новая система маршрутизации грузовиков в каком-то смысле слишком хороша. Из-за этого у водителя был такой плотный график, что у него не оставалось времени на «болтовню», необходимую для получения специального лечения, что давало продукции компании преимущество над конкурентами.***

***Без непосредственного наблюдения за системой в действии, полученного в результате участия в качестве пользователя системы, аналитик, возможно, никогда не обнаружил бы истинную проблему в конструкции системы. Как только было выделено время не только для снабжения новыми продуктами, но и для необходимой маркетинговой работы, продажи продукции вернулись к уровням, достигнутым до внедрения новой системы, и превысили их.***

#### Анализ процедур и других документов

Как отмечалось ранее, задавать вопросы людям, которые используют систему каждый день или интересуются ею, является эффективным способом сбора информации о текущих и будущих системах. Наблюдение за текущими пользователями системы — это более прямой способ увидеть, как работает существующая система, но даже этот метод дает ограниченное представление обо всех аспектах текущих операций. Эти методы определения системных требований можно усовершенствовать, исследуя системную и организационную документацию, чтобы узнать больше о текущих системах и организации, которую эти системы поддерживают.

Хотя мы обсуждаем здесь несколько важных типов документов, которые полезны для понимания возможных будущих системных требований, наше обсуждение не исчерпывает всех возможностей. Вам следует попытаться найти все письменные документы по организационным областям, имеющим отношение к реконструируемым системам. Помимо нескольких конкретных документов, которые мы обсуждаем, ценную информацию могут дать заявления о миссии организации, бизнес-планы, организационные схемы, руководства по бизнес-политике, должностные инструкции, внутренняя и внешняя переписка, а также отчеты о предыдущих организационных исследованиях.

Что анализ документов может сказать вам о требованиях к новой системе? В документах можно найти информацию о следующем:

* + Проблемы с существующими системами (например, недостающая информация или лишние шаги)
  + Возможности удовлетворения новых потребностей, если бы была доступна только определенная информация или обработка информации (например, анализ продаж на основе типа клиента)
  + Организационное направление, которое может повлиять на требования к информационной системе (например, попытка более тесно связать клиентов и поставщиков с организацией)
  + Титулы и имена ключевых лиц, заинтересованных в соответствующих существующих системах (например, имя менеджера по продажам, который проводил исследование покупательского поведения ключевых клиентов).
  + Ценности организации или отдельных лиц, которые могут помочь определить приоритеты для различных возможностей, желаемых разными пользователями (например, сохранение доли рынка, даже если это означает снижение краткосрочной прибыли).
  + Особые обстоятельства обработки информации, которые происходят нерегулярно и которые не могут быть выявлены никаким другим методом определения требований (например, специальная обработка, необходимая для нескольких крупных клиентов, которая требует использования индивидуальных процедур заказа клиентов).
  + Причина, по которой существующие системы спроектированы такими, какие они есть, что может указывать на функции, отсутствующие в текущем программном обеспечении, которые теперь могут быть осуществимы и более желательны (например, данные о покупке клиентом продуктов конкурентов не были доступны, когда текущая система была доступна). разработан; эти данные теперь доступны из нескольких источников)
  + Данные, правила обработки данных и принципы работы организации, которые должны соблюдаться информационной системой (например, каждому клиенту назначается ровно один сотрудник отдела продаж в качестве основного контактного лица, если у клиента есть какие-либо вопросы).

Одним из типов полезного документа является письменная рабочая процедура для отдельного человека или рабочей группы. Процедура описывает, как выполняется конкретная работа или задача, включая данные и информацию, которые используются и создаются в процессе выполнения работы. Например, процедура, показанная на рисунке 6-3, включает данные (перечень особенностей и преимуществ, чертежи, имя изобретателя и имена свидетелей), которые необходимы для подготовки раскрытия изобретения. В нем также указано, что помимо изобретателя, вице-президент по исследованиям, заведующий кафедрой и декан должны просматривать материал, и что для любой подачи заявления о раскрытии изобретения требуется свидетель. Эти идеи явно влияют на то, какие данные необходимо хранить, кому информацию следует отправлять, а также на правила, регулирующие действительные формы.

Однако процедуры не являются надежными источниками информации. Иногда ваш анализ нескольких письменных процедур выявляет дублирование усилий на двух или более работах. Вам следует обратить внимание руководства на такое дублирование как на проблему, которую необходимо решить до того, как можно будет приступить к проектированию системы. То есть может потребоваться перепроектирование организации, прежде чем перепроектирование информационной системы сможет принести все свои преимущества. Другая проблема, с которой вы можете столкнуться при использовании процедуры, возникает, когда процедура отсутствует. Опять же, это не ваша работа — создавать документ для отсутствующей процедуры — это дело руководства. Третья и распространенная проблема с письменной процедурой возникает, когда процедура устарела. Вы можете понять, что процедура устарела, когда проведете собеседование с лицом, ответственным за выполнение задачи, описанной в процедуре. Опять же, решение переписать процедуру так, чтобы она соответствовала действительности, принимается руководством, но вы можете вносить предложения, основанные на вашем понимании организации. Четвертая проблема, часто встречающаяся при использовании письменных процедур, заключается в том, что формальные процедуры могут противоречить информации, полученной вами в ходе собеседований и наблюдений о том, как работает организация и какая информация требуется. Как и в других случаях, решение остается за руководством.

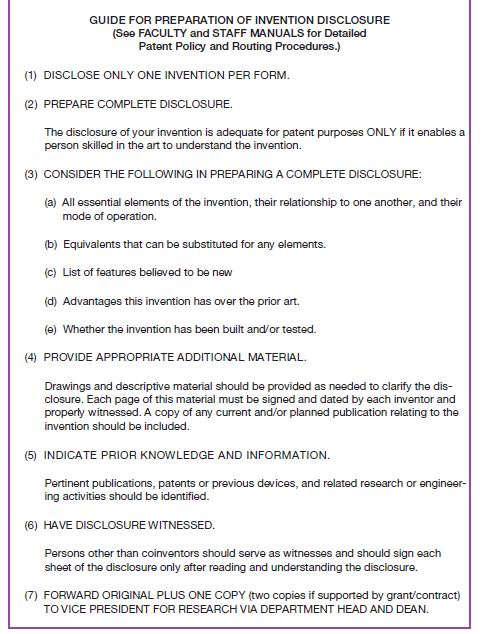


Рисунок 6 -3. Пример процесса

**Formal system -** Официальный способ работы системы, описанный в организационной документации.

**informal system -** Как на самом деле работает система.

Все эти проблемы иллюстрируют разницу между формальными и неформальными системами. Формальные системы признаются в официальной документации организации; неформальные системы – это способ, которым организация фактически работает. Неформальные системы развиваются из-за неадекватности формальных процедур, индивидуальных рабочих привычек и предпочтений, сопротивления контролю и других факторов. Важно понимать как формальные, так и неформальные системы, поскольку каждая из них дает представление о требованиях к информации и о том, что потребуется для перехода от нынешних информационных услуг к будущим.

Второй тип документов, полезный для системных аналитиков, — это бизнес-форма (см. рис. 6-4). Формы используются для всех типов бизнес-функций: от записи заказа, подтверждающего оплату счета, до указания того, какие товары были отгружены. Формы важны для понимания системы, поскольку они явно указывают, какие данные входят в систему или выходят из нее и какие необходимы для функционирования системы. В образце формы счета на рис. 6-4 мы видим места для таких данных, как имя и адрес доставки счета клиента, номер счета, данные (количество, описание, сумма) о каждой позиции в счете и расчетные данные, такие как общее количество.

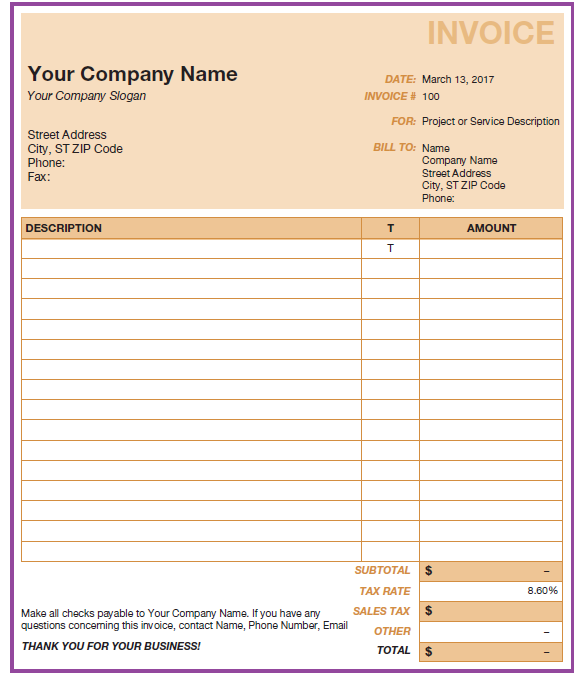


Рисунок 6 – 4. Форма счета-фактуры из Microsoft Exel

Форма дает нам важную информацию о характере организации. Например, компания может отправлять товары и выставлять счета на разные адреса; клиентам могут быть предоставлены скидки; и расходы на перевозку взимаются с клиента. Печатная форма может соответствовать компьютерному дисплею, который система будет генерировать для ввода и сохранения данных или для отображения данных онлайн-пользователям. Формы наиболее полезны для вас, когда они содержат фактические данные организации, поскольку это позволяет вам определить характеристики данных, которые фактически используются приложением. Способы использования форм людьми со временем меняются, и данные, которые были необходимы при разработке формы, могут больше не потребоваться. Вы можете использовать методы системного анализа, представленные в главах 7 и 8, чтобы определить, какие данные больше не требуются.

Третий тип полезного документа — это отчет, созданный современными системами. В качестве основного вывода для некоторых типов систем отчет позволяет вам перейти от информации в отчете к данным, которые должны были быть необходимы для его создания. На рисунке 6-5 представлен пример типичного финансового отчета – отчета о движении денежных средств. Вы анализируете такие отчеты, чтобы определить, какие данные необходимо собрать за какой период времени и какие манипуляции с этими необработанными данными потребуются для создания каждого поля отчета.

Если текущая система основана на компьютере, четвертый набор полезных документов — это те, которые описывают текущие информационные системы — как они были спроектированы и как они работают. Под это описание подходит множество различных типов документов: от блок-схем до словарей данных, от отчетов инструментов CASE до руководств пользователя. Аналитику, имеющему доступ к таким документам, повезло; многие информационные системы, разработанные собственными силами, не имеют полной документации (если не использовался инструмент CASE).

Анализ организационной документации и наблюдение, наряду с интервьюированием, являются наиболее часто используемыми методами сбора системных требований. В Таблице 6-4 обобщены сравнительные особенности наблюдения и анализа организационных документов

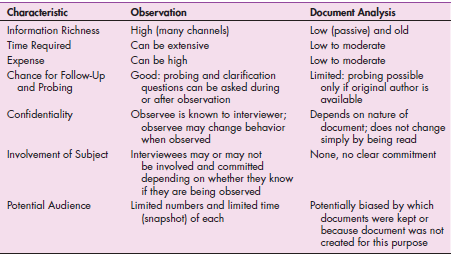
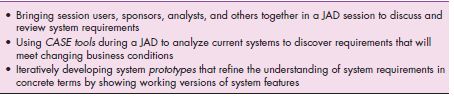


Таблица 6-4. Сравнение наблюдений и анализа документов

### Сравнение наблюдений и анализа документов

Несмотря на то, что мы назвали интервью, наблюдение и анализ документов традиционными методами определения требований к системе, все эти методы по-прежнему широко используются аналитиками для сбора важной информации. Однако сегодня существуют дополнительные методы сбора информации о текущей системе, организационной области, запрашивающей новую систему, и о том, какой должна быть новая система. В этом разделе вы узнаете о нескольких современных методах сбора информации для анализа (перечисленных в таблице 6-5): JAD, инструментах CASE для поддержки JAD и прототипировании. Как мы говорили ранее, эти методы могут способствовать эффективному сбору и структурированию информации, одновременно сокращая время, необходимое для анализа.

Таблица 6-5. Современные методы сбора системных требований



#### Совместная разработка приложения

Совместное проектирование приложений (JAD) началось в конце 1970-х годов в IBM, и с тех пор практика JAD распространилась во многих компаниях и отраслях. Например, он довольно популярен в страховой отрасли Коннектикута, где сформировалась группа пользователей JAD. Фактически, было задокументировано и популяризировано несколько общих подходов к JAD (см., например, Wood and Silver, 1995). Основная идея JAD — объединить ключевых пользователей, менеджеров и системных аналитиков, участвующих в анализе текущей системы. В этом отношении JAD похож на групповое собеседование; JAD, однако, следует определенной структуре ролей и повестки дня, которая сильно отличается от группового интервью, во время которого аналитики контролируют последовательность вопросов, на которые отвечают пользователи. Основная цель использования JAD на этапе анализа — одновременный сбор требований к системе от ключевых людей, связанных с системой. В результате получается интенсивный и структурированный, но очень эффективный процесс. Как и в случае с групповым интервью, собрание всех ключевых людей в одном месте в одно и то же время позволяет аналитикам увидеть, где есть области согласия, а где — конфликты. Встреча со всеми этими важными людьми в течение более чем недели интенсивных занятий дает вам возможность разрешить конфликты или, по крайней мере, понять, почему конфликт может быть нелегко разрешить.

Сеансы JAD обычно проводятся в другом месте, а не в том месте, где обычно работают участвующие в них люди. Идея такой практики состоит в том, чтобы удержать участников от как можно большего количества отвлекающих факторов, чтобы они могли сосредоточиться на системном анализе. JAD может длиться от четырех часов до целой недели и может состоять из нескольких сеансов. JAD использует корпоративные ресурсы стоимостью тысячи долларов, самым дорогим из которых является время задействованных людей. Другие расходы включают расходы, связанные с доставкой людей в отдаленные места, размещением их в гостиницах и кормлением в течение нескольких дней.

**Joint Application Design (JAD)** *Структурированный процесс, в котором пользователи, менеджеры и аналитики работают вместе в течение нескольких дней в ходе серии интенсивных совещаний, чтобы определить или просмотреть системные требования.*

**JAD session leader** *Обученный специалист, который планирует и проводит сеансы совместной разработки приложений.*

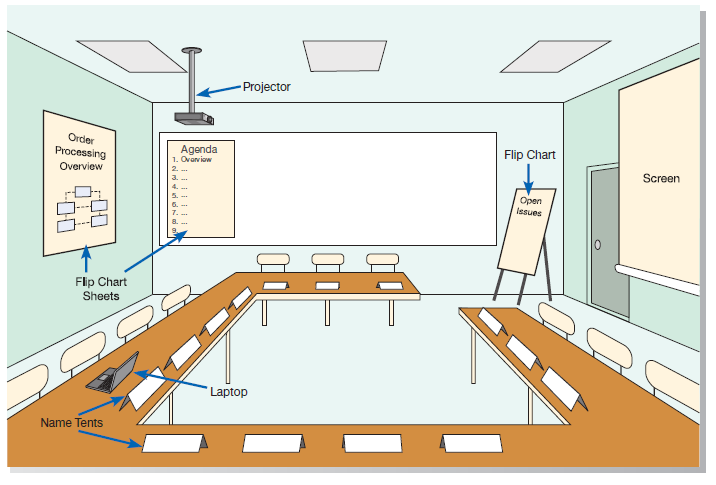
**Scribe** *Человек, который подробно записывает происходящее на сессии совместной разработки приложений.*

Типичные участники JAD перечислены ниже:

* Руководитель сессии JAD. Руководитель сессии JAD организует и управляет JAD. Этот человек прошел обучение групповому управлению и фасилитации, а также системному анализу. Лидер JAD устанавливает повестку дня и следит за ее выполнением; он или она остается нейтральным по вопросам и не вносит идеи или мнения, а скорее концентрируется на том, чтобы держать группу в повестке дня, разрешать конфликты и разногласия и выявлять все идеи.
* Пользователи. Ключевые пользователи рассматриваемой системы являются важными участниками JAD. Они единственные, кто имеет четкое представление о том, что значит использовать систему ежедневно.
* Менеджеры. Менеджеры рабочих групп, использующих рассматриваемую систему, дают представление о новых организационных направлениях, мотивах и организационном влиянии систем, а также поддерживают требования, определенные в ходе JAD.
* Спонсор. Поскольку JAD является крупным мероприятием из-за его затрат, он должен спонсироваться кем-то на относительно высоком уровне в компании. Если спонсор и посещает какие-либо занятия, то обычно только в самом начале или в конце.
* Системные аналитики. Члены группы системного анализа присутствуют на JAD, хотя их фактическое участие может быть ограничено. Аналитики созданы для того, чтобы учиться у пользователей и менеджеров, а не для того, чтобы управлять процессом или доминировать над ним.
* Технический писатель. Писец делает заметки во время сеансов JAD. Обычно это делается на ноутбуке. Примечания можно делать с помощью текстового процессора, а примечания и диаграммы можно вводить непосредственно в инструмент CASE.
* Сотрудники ИС. Помимо системных аналитиков, другие сотрудники информационных систем (ИС), такие как программисты, аналитики баз данных, планировщики ИБ и персонал центров обработки данных, могут присутствовать, чтобы извлечь уроки из обсуждения и, возможно, поделиться своими идеями о технической осуществимости предлагаемых идей или технических ограничениях. современных систем.

Сеансы JAD обычно проводятся в специальных комнатах, где участники сидят за столами в форме подковы, как показано на рисунке 6-6. Эти комнаты обычно оборудованы досками. Могут использоваться и другие аудиовизуальные инструменты, такие как магнитные символы, которые можно легко переставлять на доске, флип-чартах и компьютерных дисплеях. Флип-чарт обычно используется для отслеживания вопросов, которые не могут быть решены в ходе JAD, или для тех вопросов, которые требуют дополнительной информации, которую можно собрать во время перерывов в разбирательстве. Компьютеры можно использовать для создания и отображения проектов форм или отчетов, построения диаграмм существующих или заменяющих систем или создания прототипов.

Когда JAD завершен, конечным результатом является набор документов, в которых подробно описывается работа существующей системы, связанная с изучением системы-замены. В зависимости от точной цели JAD, аналитики также могут отказаться от JAD с некоторой подробной информацией о том, что требуется от системы замены.



**Figure 6-6**

Иллюстрация типичной планировки помещения JAD

**Участие в JAD** Представьте, что вы системный аналитик и участвуете в своем первом JAD. Каким может быть участие в JAD? Обычно JAD проводятся за пределами офиса в комфортабельных конференц-залах. В первое утро JAD вы и ваши коллеги-аналитики входите в комнату, очень похожую на ту, что изображена на рис. 6-6. Фасилитатор JAD уже здесь; она заканчивает писать повестку дня на флип-чарте. Писец сидит в углу со своим ноутбуком и готовится делать заметки о событиях дня. Пользователи и менеджеры начинают объединяться в группы и рассаживаются вокруг U-образного стола. Вы и другие аналитики просматриваете свои записи, в которых описывается то, что вы уже узнали об информационной системе, которую вы все здесь обсуждаете. Ведущий сессии открывает собрание приветствием и кратким изложением повестки дня. Первый день будет посвящен общему обзору действующей системы и основным проблемам, связанным с ней. Следующие два дня будут посвящены анализу текущих экранов системы. Последние два дня будут посвящены анализу отчетов**.**

Руководитель сессии представляет корпоративного спонсора, который рассказывает об организационном подразделении и текущей системе, связанной с исследованием системного анализа, а также о важности обновления существующей системы для удовлетворения меняющихся условий бизнеса. Он уходит, и его место занимает руководитель сессии JAD. Она уступает слово старшему аналитику, который начинает презентацию о ключевых проблемах системы, которые уже выявлены. После презентации ведущий сессии открывает обсуждение для пользователей и менеджеров в комнате.

После нескольких минут разговора начинается острая дискуссия между двумя пользователями из разных корпоративных офисов. Один пользователь, представляющий офис, который послужил моделью для первоначального проектирования системы, утверждает, что воспринимаемая нехватка гибкости системы на самом деле является преимуществом, а не проблемой. Другой пользователь, представляющий офис, который до слияния был частью другой компании, утверждает, что нынешняя система настолько негибкая, что ее практически невозможно использовать. Руководитель сеанса вмешивается и пытается помочь пользователям изолировать определенные аспекты системы, которые могут способствовать ощущению недостаточной гибкости системы.

Возникают вопросы о намерениях первоначальных разработчиков. Руководитель сессии спрашивает группу аналитиков об их впечатлениях от первоначального проекта системы. Поскольку на эти вопросы невозможно ответить во время встречи (никто из первоначальных проектировщиков не присутствует и ни одна из исходных проектных документов не доступна), ведущий сессии помещает вопрос о намерениях в список «дела». Этот вопрос становится первым на флип-чарте с пунктами «что сделать», и ведущий сессии дает вам задание выяснить намерения первоначальных дизайнеров. Она записывает ваше имя рядом с пунктом «Что нужно сделать» в списке и продолжает сеанс. До окончания JAD вы должны получить ответ на этот вопрос.

JAD будет продолжать действовать в том же духе до конца своего существования. Аналитики будут выступать с презентациями, помогать вести дискуссии по дизайну форм и отчетов, отвечать на вопросы пользователей и менеджеров, а также делать заметки по сказанному. После каждой встречи аналитическая группа встречается, обычно неформально, чтобы обсудить то, что произошло за день, и закрепить полученные знания. Пользователи будут продолжать вносить свой вклад во время встреч, а лидер сессии будет содействовать, вмешиваясь в конфликты и следя за тем, чтобы группа следовала повестке дня. По завершении JAD руководитель сессии и ее помощники должны подготовить отчет, документирующий результаты JAD, и распространить его среди пользователей и аналитиков.

**CASE средства в процессе JAD** Для определения и структурирования требований наиболее полезными инструментами CASE являются диаграммы, формы и отчеты. Чем больше аналитики взаимодействуют с пользователями на этом этапе, тем полезнее становится этот набор инструментов. Аналитик может использовать инструменты построения диаграмм и прототипирования, чтобы придать графическую форму системным требованиям, показать инструменты пользователям и внести изменения на основе реакции пользователей. Эти же инструменты очень ценны и для структурирования требований. Использование общих инструментов CASE во время определения и структурирования требований упрощает переход между этими двумя подэтапами и сокращает общие затраты времени. При структурировании также полезны инструменты CASE, которые анализируют информацию о требованиях на предмет правильности, полноты и непротиворечивости. Наконец, при создании и выборе альтернатив инструменты построения диаграмм и прототипирования являются ключевыми для предоставления пользователям графических иллюстраций того, как будут выглядеть альтернативные системы. Такая практика предоставляет пользователям и аналитикам более качественную информацию для выбора наиболее подходящей альтернативной системы.

Некоторые наблюдатели выступают за использование инструментов CASE во время JAD (Lucas, 1993). Запуск инструмента CASE вовремя JAD позволяет аналитикам вводить модели системы непосредственно в инструмент CASE, обеспечивая согласованность и надежность совместного процесса построения моделей. Инструмент CASE фиксирует системные требования более гибким и полезным способом, чем это может сделать писатель или группа аналитиков, делающая заметки. Кроме того, инструмент CASE можно использовать для проектирования меню, отображения и составления отчетов, поэтому пользователи могут напрямую наблюдать за старыми и новыми конструкциями и оценивать их полезность для группы анализа.

#### Использование прототипирования в течение разделения требований

**Prototyping** - Итеративный процесс разработки систем, в ходе которого требования преобразуются в работающую систему, которая постоянно пересматривается посредством тесного сотрудничества между аналитиком и пользователями.

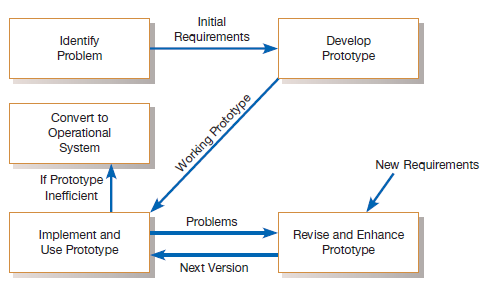
**Прототипирование —** это итеративный процесс с участием аналитиков и пользователей, в ходе которого элементарная версия информационной системы создается и перестраивается в соответствии с отзывами пользователей. Прототипирование может заменить жизненный цикл разработки системы или дополнить его. Нас интересует, как прототипирование может улучшить процесс определения требований**.**

Чтобы собрать первоначальный базовый набор требований, вам все равно придется опросить пользователей и собрать документацию. Однако прототипирование позволит вам быстро преобразовать основные требования в работающую, хотя и ограниченную, версию желаемой информационной системы. Затем прототип будет просмотрен и протестирован пользователем. Обычно вид словесных описаний требований, преобразованных в физическую систему, побуждает пользователя изменить существующие требования и создать новые. Например, в ходе первоначальных интервью пользователь мог сказать, что он хочет, чтобы вся соответствующая информация о выставлении счетов за коммунальные услуги (например, имя и адрес клиента, запись об оказании услуг и история платежей) отображалась на единой компьютерной форме. Как только тот же пользователь увидит, насколько перегруженным и запутанным будет такой дизайн в прототипе, он может изменить свое мнение и вместо этого попросить организовать информацию на нескольких экранах, но с легкими переходами от одного экрана к другому. Ему также можно напомнить о некоторых важных требованиях (данные, расчеты и т. д.), которые не были выявлены во время первоначальных собеседований.

Затем вы перепроектируете прототип, включив в него предложенные изменения (рис. 6-7). После внесения изменений пользователи снова смогут просмотреть и протестировать прототип. И, еще раз, вы учтете их предложения по изменениям. Благодаря такому итеративному процессу высока вероятность того, что вы сможете лучше уловить требования системы.

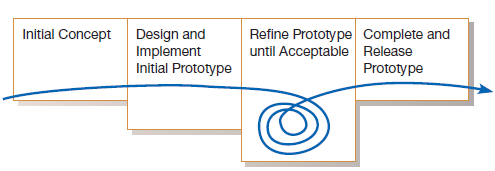
По мере того как прототип меняется на каждой итерации, в него включается все больше и больше проектных характеристик системы. Прототип может затем служить основой производственной системы в процессе, называемом эволюционным прототипированием. Альтернативно прототип может служить только моделью, которая затем используется в качестве эталона для построения реальной системы. В этом процессе, называемом одноразовым прототипированием, прототип выбрасывается после его использования.

**Эволюционное прототипированне** При эволюционном прототипировании вы начинаете с моделирования частей целевой системы и, если процесс прототипирования успешен, вы развиваете остальную часть системы из этих частей (McConnell, 1996). Модель жизненного цикла эволюционного прототипирования иллюстрирует итерационный характер процесса и тенденцию усовершенствовать прототип до тех пор, пока он не будет готов к выпуску (рис. 6-8). Одним из ключевых аспектов этого подхода является то, что прототип становится реальной производственной системой. Из-за этого вы часто начинаете с тех частей системы, которые наиболее сложны и неопределенны.



**Figure 6-7**

The prototyping methodology



**Figure 6-8**

McConnell’s evolutionary prototyping

Model

Хотя система-прототип может отлично представлять легко видимые аспекты системы, такие как пользовательский интерфейс, сама производственная система будет выполнять гораздо больше функций, некоторые из которых прозрачны или невидимы для пользователей. Любая система должна быть спроектирована так, чтобы облегчить доступ к базе данных, целостность базы данных, безопасность системы и работу в сети. Системы также должны быть спроектированы так, чтобы поддерживать масштабируемость, многопользовательскую поддержку и поддержку нескольких платформ. Некоторые из этих проектных спецификаций будут закодированы в прототипе. Более того, до 90 процентов функционирования системы посвящено обработке исключительных случаев (McConnell, 1996). Прототипы предназначены для обработки только типичных случаев, поэтому обработка исключений должна быть добавлена к прототипу при его преобразовании в производственную систему. Понятно, что прототип отражает лишь часть системных требований.

**Одноразовое прототипирование** В отличие от эволюционного прототипирования, одноразовое прототипирование не сохраняет разработанный прототип. При одноразовом прототипировании никогда не возникает намерения превратить прототип в работающую систему. Вместо этого прототип разрабатывается быстро, чтобы продемонстрировать некоторые неясные аспекты конструкции системы или помочь пользователям сделать выбор между различными функциями или характеристиками интерфейса. Как только неопределенность, для решения которой был создан прототип, уменьшена, от прототипа можно отказаться, а принципы, извлеченные из его создания и тестирования, затем могут стать частью определения требований.

Прототипирование наиболее полезно для определения требований, когда:

* требования пользователей не ясны или недостаточно понятны, что часто бывает в случае совершенно новых систем или систем, поддерживающих принятие решений;
* в системе задействованы один или несколько пользователей и других заинтересованных сторон;
* возможные проекты сложны и требуют конкретной формы для полной оценки;
* В прошлом между пользователями и аналитиками существовали проблемы со связью, и обе стороны хотят быть уверены, что системные требования являются как можно более конкретными; и
* инструменты (такие как генераторы форм и отчетов) и данные легко доступны для быстрого создания рабочих систем

Прототипирование как инструмент определения требований также имеет некоторые недостатки.

К ним относятся следующие:

* Прототипы имеют тенденцию избегать создания формальной документации системных требований, что затем может затруднить разработку полностью работающей системы.
* Прототипы могут стать очень уникальными для первоначального пользователя, и их будет трудно распространить или адаптировать к другим потенциальным пользователям.
* Прототипы часто создаются как автономные системы, игнорируя таким образом вопросы совместного использования данных и взаимодействия с другими существующими системами, а также проблемы масштабирования приложений.
* Проверки в SDLC обойдены, поэтому могут быть забыты некоторые более тонкие, но все же важные системные требования (например, безопасность, некоторые элементы управления вводом данных или стандартизация данных в системах).

#### Радикальные методы разделения системных требований

Традиционные или современные методы определения системных требований, о которых вы прочитали в этой главе, применимы к любым усилиям по определению требований, независимо от их мотивации. Но большая часть того, что вы узнали, традиционно применялась в проектах разработки систем, предполагающих автоматизацию существующих процессов. Аналитики используют определение системных требований, чтобы понять текущие проблемы и возможности, а также определить, что необходимо и желательно в будущих системах. Как правило, нынешний способ ведения дел оказывает большое влияние на новую систему. Однако в некоторых организациях руководство ищет новые способы выполнения текущих задач. Эти новые способы могут радикально отличаться от того, как все делается сейчас, но выгоды могут быть огромными: для выполнения той же работы может потребоваться меньше людей, отношения с клиентами могут значительно улучшиться, а процессы могут стать намного более эффективными и действенными, и все это из которых может привести к увеличению прибыли. Общий процесс замены существующих методов радикально новыми обычно называют **реинжинирингом бизнес-процессов** (BPR). Хотя термин BPR обычно ассоциируется с увлечением менеджментом, произошедшим в 1990-х годах, предприятия по-прежнему жизненно заинтересованы в бизнес-процессах и способах их улучшения (Sharp and McDermott, 2001). Даже если кому-то термин «реинжиниринг бизнес-процессов» может показаться устаревшим, ориентация на процесс остается неизгладимым наследием движения BPR.

**Business process reengineering (BPR)** Поиск и внедрение радикальных изменений в бизнес-процессах для достижения прорывных улучшений в продуктах и услугах.

Чтобы лучше понять BPR, рассмотрим следующую аналогию. Предположим, вы успешный европейский игрок в гольф, который настроил свою игру так, чтобы она соответствовала стилю полей для гольфа и погоде в Европе. Вы научились контролировать полет мяча при сильном ветре, катать мяч по широко открытым полям, бить по большим и волнистым полям и нацеливаться на цель без помощи ландшафтного дизайна, распространенного на полях в Северной Америке. Когда вы приезжаете в Соединенные Штаты, чтобы заработать состояние на туре по США, вы обнаруживаете, что постепенное улучшение вашей игры в мяч, точности вождения и ударов по песку поможет, но новая конкурентная среда просто не подходит для вашего стиля игры. Вам необходимо перепроектировать весь свой подход, научившись прицеливаться в мишени, вращать и останавливать мяч на траве, а также управлять отвлекающими факторами толпы и прессы. Если вы достаточно хороши, вы можете выжить, но без реинжиниринга вы никогда не станете победителем.

Точно так же, как конкурентоспособность гольфа вынуждает хороших игроков адаптировать свои игры к изменяющимся условиям, конкурентоспособность нашей глобальной экономики привела большинство компаний к постоянному улучшению качества своих продуктов и услуг (Добинс и Кроуфорд-Мейсон, 1991). Организации осознают, что творческое использование информационных технологий может привести к значительным улучшениям в большинстве бизнес-процессов. Идея BPR заключается не просто в улучшении каждого бизнес-процесса, но, в смысле системного моделирования, в реорганизации полного потока данных в основных подразделениях организации, чтобы исключить ненужные шаги, добиться синергии между ранее отдельными шагами и повысить оперативность реагирования. к будущим изменениям. Такие компании, как IBM, Procter & Gamble, Walmart и Ford, активно реализуют BPR и добились больших успехов. Однако многие другие компании столкнулись с трудностями в применении принципов BPR (Moad, 1994). Тем не менее, концепции BPR активно применяются как в корпоративном стратегическом планировании, так и в планировании информационных систем как способ радикально улучшить бизнес-процессы (как описано в главе 4).

Сторонники BPR предполагают, что радикального повышения качества бизнес-процессов можно добиться за счет творческого применения информационных технологий. Сторонники BPR также предполагают, что радикальных улучшений нельзя добиться путем корректировки существующих процессов, а скорее использовать чистый лист бумаги и задаваться вопросом: «Если бы мы были новой организацией, как бы мы выполнили эту деятельность?» Изменение способа выполнения работы также меняет способы совместного использования и хранения информации, а это означает, что результатом многих усилий BPR является разработка запросов на обслуживание информационных систем или запросов на замену системы. Вполне вероятно, что вы столкнетесь или уже сталкивались с инициативами BPR в своей организации.