

УПРАВЛЕНИЕ ИТ-РАЗРАБОТКОЙ И ВНЕДРЕНИЕМ

Екатеринбург
2021

Министерство науки и высшего образования Российской Федерации
Уральский государственный экономический университет



УПРАВЛЕНИЕ ИТ-РАЗРАБОТКОЙ И ВНЕДРЕНИЕМ

Рекомендовано
Советом по учебно-методическим вопросам и качеству образования
Уральского государственного экономического университета
в качестве учебного пособия

Екатеринбург
2021

УДК 005(075)
ББК 65.290.21я73
У67

Рецензенты:
ученый совет
Института радиоэлектроники и информационных технологий — РтФ
Уральского федерального университета
имени первого Президента России Б. Н. Ельцина
(протокол № 9 от 27 мая 2021 г.)
генеральный директор ООО «Октоника»
К. Г. Ведьманов

Авторский коллектив:
В. П. Часовских, М. П. Воронов, В. Г. Лабунец, Е. Н. Стариков

У67 **Управление ИТ-разработкой и внедрением** : учебное пособие / авторский коллектив : В. П. Часовских, М. П. Воронов, В. Г. Лабунец, Е. Н. Стариков ; Министерство науки и высшего образования Российской Федерации, Уральский государственный экономический университет. — Екатеринбург : УрГЭУ, 2021. — 173 с.

ISBN 978-5-9656-0318-3

Цель учебного пособия — формирование системы теоретических и практических знаний об основах управления ИТ-разработкой и внедрением. Пособие включает общетеоретические и практические аспекты управления разработкой ИТ-проектов: инициация, управление интеграцией, оценка и планирование ИТ-проекта, инструменты управления расписанием, человеческими ресурсами и командой проекта, управление рисками и контроль качества. Структура и содержание пособия полностью соответствуют требованиям федеральных государственных образовательных стандартов высшего образования (ФГОС ВО).

Для студентов очной и заочной форм обучения направления 02.03.03 «Математическое обеспечение и администрирование информационных систем» (бакалавриат). Также может быть полезно студентам направлений 09.03.03 «Прикладная информатика» (бакалавриат), 09.03.01 «Информатика и вычислительная техника» (бакалавриат), 38.04.05 «Бизнес-информатика» (магистратура), 09.04.03 «Прикладная информатика» (магистратура).

УДК 005(075)
ББК 65.290.21я73

ISBN 978-5-9656-0318-3

© Часовских В. П., Воронов М. П.,
Лабунец В. Г., Стариков Е. Н., 2021
© Уральский государственный
экономический университет, 2021

Введение

Управление ИТ-разработкой и внедрением играет важную роль в различных отраслях и областях, таких как разработка программного обеспечения, инжиниринг, строительство, маркетинг, ИТ-операции и др.

Комплекс знаний по управлению ИТ-разработкой и внедрением в настоящее время очень обширен и покрывает все аспекты разработки ИТ-проектов, начиная от генерации идеи и заканчивая тестированием готового продукта и его эксплуатацией. Управление разработкой ИТ-проектов включает в себя все усилия, направленные на управление процессами и деятельностью, связанными с обеспечением успеха ИТ-проектов. Но также может включать в себя управление любым ИТ-проектом за пределами тех типов организаций, которые занимаются ИТ-инфраструктурой, информационными системами или компьютерными технологиями в целом.

ИТ-проект может быть любым типом проекта, который имеет дело с ИТ-инфраструктурой, информационными системами или компьютерными технологиями. Это может быть деятельность по разработке программного обеспечения, такая как программирование простого мобильного приложения или крупномасштабной программной системы, а также веб-разработка, включая обновление веб-страницы, создание сайта онлайн-покупок или даже разработка всей веб-инфраструктуры. Другие распространенные примеры ИТ-проектов — проектирование ИТ-инфраструктуры организации, развертывание систем и программного обеспечения, обеспечение ИТ-безопасности.

Управление ИТ-проектами включает такие типичные виды деятельности, как управление задачами, установление сроков, выполнение заданий и руководство командами для достижения целей компании.

Ранние этапы управления охватывают оценку, определение приоритетов и планирование потенциальных ИТ-проектов. Опре-

деляются объем работ и доступные ресурсы, назначение ресурсов, а также ожидания в отношении этапов проекта. Далее руководитель ИТ-проекта должен быть ориентирован на выполнение других видов деятельности, таких как координация рабочей нагрузки и рабочих процессов проекта, управление фактической работой, содействие коммуникации и сотрудничеству между членами команды, получение результатов и оценка результатов проекта. Таким образом, для ИТ-менеджера управление проектами сводится к определению наилучшего способа их инициирования, планирования и выполнения. Существует множество методологий и инструментов управления разработкой ИТ-проектов, выбор которых зависит от конкретных задач. Поскольку каждая методология имеет свои преимущества, часто руководители ИТ-проектов реализуют гибридный подход, используя ту из них, которая больше подходит для конкретного рассматриваемого проекта или его части.

Отдельные лица или группы, которым необходимо управлять только небольшими и простыми ИТ-проектами, могут использовать исключительно электронные таблицы и электронную почту. Эти простые инструменты обладают определенным потенциалом, но по мере роста проектов они становятся неэффективными, что приводит к недопониманию и другим проблемам.

Решения, разработанные специально для управления ИТ-проектами, могут упростить и активизировать усилия менеджеров команд. Однако многие из этих решений сосредоточены в основном на управлении исполнением проектов, игнорируя другие важные мероприятия, такие как оценка проектов, определение приоритетов, планирование, сотрудничество, отчетность, управление рисками, контроль качества и т. д.

Одни системы поддерживают только определенную методологию управления проектами, другие — множество. Сложность и стоимость определенных программных средств или решений с большим функционалом и набором инструментов для управления проектами зачастую затрудняют их использование.

Лучшая система управления ИТ-проектами — та, которая облегчает все аспекты управления, поддерживает несколько методологий, проста в использовании и относительно недорога. Это также имеет большое значение, если решение централизует

усилия по совместной работе и обеспечивает высокую доступность и совместную работу в режиме реального времени, предлагаемую облаком.

В первой главе пособия подробно рассматриваются стадии и модели жизненного цикла ИТ-проекта, процессы его инициации, управления интеграцией и предварительной оценки.

Вторая глава посвящена вопросам планирования ИТ-проекта, включая методы определения списка и взаимосвязи операций.

Третья глава включает модели и инструменты управления расписанием ИТ-проекта, основные аспекты управления человеческими ресурсами и командой как единым целым.

В четвертой главе дается описание основных инструментов качественного и количественного анализа рисков, приводятся методы идентификации рисков и стратегии реагирования на рисковые события, а также инструменты контроля качества ИТ-проекта и сопровождающей проект документации.

Пятая глава содержит рекомендации по разработке корпоративной информационной системы промышленного предприятия, включая требования к среде проектирования приложений, требования к основным компонентам системы, графическому интерфейсу и отчетным формам.

В конце каждой главы приведены вопросы для самоконтроля. Обучающимся рекомендуется найти ответы на них с целью наилучшего усвоения теоретического материала. Также предлагается практическая работа, которую надо выполнить, используя основной теоретический материал.

Пособие соответствует учебной программе дисциплины «Управление ИТ-разработкой и внедрением» и дисциплины «Управление ИТ-разработкой», изучаемой студентами направления 02.03.03 «Математическое обеспечение и администрирование информационных систем» (бакалавриат) очной и заочной форм обучения. Также может использоваться студентами направлений 09.03.03 «Прикладная информатика» (бакалавриат), 09.03.01 «Информатика и вычислительная техника» (бакалавриат), 38.04.05 «Бизнес-информатика» (магистратура), 09.04.03 «Прикладная информатика» (магистратура).

Глава 1

Жизненный цикл ИТ-проекта

1.1. Жизненный цикл проекта: понятие, основные фазы

Жизненный цикл любого проекта состоит из четырех основных этапов (фаз), в ходе которых менеджер проекта и его команда пытаются достичь поставленных целей. Это: концепция (начало), планирование, выполнение (реализация) и закрытие.

Таким образом, каждый проект имеет начало, центральный период, завершение и заключительную фазу (успешную или нет).

Упомянутые выше этапы являются лишь основными вехами проекта, что означает отправную точку для дальнейшего разделения на подэтапы, мероприятия и задачи, все более базовые и подробные, необходимые для разработки и распределения работы между ресурсами.

На каждом этапе или мероприятии жизненного цикла руководитель проекта должен держать в голове два момента:

1) цели каждого этапа проекта — на основе ограничений компании, начиная от качества и заканчивая сроками и затратами;

2) продукты — каждая деятельность должна приводить к результатам, которые могут быть материальными товарами, документацией, конкретными услугами и т. д.

Рассмотрим подробнее этапы жизненного цикла.

Начальный этап. На первом этапе определяется цель или «потребность» проекта. Это может быть, например, решение бизнес-проблемы или анализ конкретной потребности. Соответству-

ющая разработка в ответ на эту потребность может быть задокументирована в бизнес-кейсе с рекомендуемыми вариантами решения. Затем проводится технико-экономическое обоснование, чтобы проверить, соответствует ли каждый вариант поставленной цели, и определяется окончательное решение. В технико-экономическом анализе также производится обоснование целесообразности проекта. После того как анализ проведен и проект признан осуществимым и необходимым, он официально начинается, и назначается руководитель проекта. Затем определяется и формируется проектная группа.

Этап планирования. На этом этапе в соответствии с целью проекта осуществляется подробное планирование шагов необходимых для достижения окончательного решения. Затем определяются конкретные задачи проекта, а также требования к ресурсам и стратегия. Создается план проекта, который содержит действия, задачи и сроки. Руководитель проекта координирует подготовку бюджета проекта, представляя смету затрат на рабочую силу, оборудование и материалы, если это необходимо. Бюджет используется для мониторинга и контроля расходов, понесенных на протяжении всего проекта. После того как руководитель проекта определил работу, подготовил стратегию, а также показатели и сметные затраты, основные компоненты процесса планирования завершены. Затем выявляются и устраняются факторы, которые могут представлять угрозу для успеха проекта. Эта часть называется управлением рисками. Выявляются потенциальные проблемы, а также действия, которые необходимо предпринять, чтобы избежать их, решить или, по крайней мере, уменьшить их воздействие. Также выявляются заинтересованные стороны и разрабатывается план коммуникации, в котором излагается информация необходимая для информирования всех сторон, участвующих в проекте. Наконец, руководитель проекта разрабатывает план качества, который включает в себя цели в области качества, меры контроля, а также перечень критериев, которым необходимо соответствовать, чтобы получить одобрение клиента, которым, в частности, может быть сама компания.

Этап выполнения. На третьем этапе план проекта приводится в действие, и работа выполняется, следуя шагам, определенным на этапе планирования. Прогресс постоянно отслежива-

ется, и в случае отклонения соответствующие изменения вносятся и документируются как изменения по отношению к первоначальному плану. В ходе проекта люди выполняют поставленные конкретные задачи, обмен информацией о ходе работы производится на регулярных встречах команды, так называемых совещаниях о состоянии выполнения. Руководитель проекта использует эту информацию для поддержания контроля над направлением проекта путем сравнения отчетов о ходе выполнения с планом проекта, для измерения эффективности деятельности и принятия корректирующих мер, если это необходимо. Основная стратегия всегда должна заключаться в том, чтобы при любых отклонениях вернуть проект в его первоначальное русло. Если это невозможно, должны быть произведены изменения в исходном плане, а измененный план должен быть доведен до членов команды. На этом этапе спонсоры проекта и другие заинтересованные стороны должны регулярно информироваться о ходе работы. Каждый результат продукта должен быть проанализирован и принят. Как только будут получены результаты всех этапов и клиент примет окончательное решение, проект будет готов к закрытию.

Закрытие проекта. На заключительном этапе проводится подведение окончательных результатов, поставка проектной документации, расторжение контрактов с поставщиками, высвобождение ресурсов проекта, сообщение о закрытии проекта всем заинтересованным сторонам. Самый последний шаг — анализ того, что прошло хорошо, а что нет. Благодаря этому приобретается опыт и знания, выявляются факторы, которые помогут руководителю проекта и команде в целом в будущих проектах. К сожалению, заключительная фаза часто недооценивается, и во многих компаниях проект реализуется без дополнительной оценки; важно только, был ли проект успешным или нет. На самом деле важно не только успешно завершить проект, но и иметь возможность выполнить его так, как это было предусмотрено в первоначальном плане. Фаза закрытия также служит для того, чтобы избежать повторения ошибок в будущем и неадекватной оценки определенных рисков [1].

Жизненный цикл ИТ-проектов (в частности, направленных на создание программного продукта) имеет ряд отличительных особенностей.

Жизненный цикл в данном случае — это процесс, который определяет различные этапы разработки программного обеспечения для предоставления высококачественного продукта. Этапы охватывают полный жизненный цикл программного обеспечения — от создания до выхода продукта из эксплуатации.

Соблюдение жизненного цикла ИТ-проекта приводит к систематической и дисциплинированной разработке программного обеспечения. Целью инициирования жизненного цикла ИТ-проекта является поставка высококачественного продукта в соответствии с требованиями заказчика.

В рамках жизненного цикла ИТ-проекта основные этапы определяются как: 1) сбор требований; 2) проектирование; 3) кодирование; 4) тестирование; 5) развертывание и 6) техническое обслуживание.

Очень важно придерживаться этих этапов для обеспечения системности при создании продукта. Допустим, должно быть разработано программное обеспечение, при этом команда разделена для работы над функцией продукта и каждому разрешено работать так, как он хочет. Тогда один из разработчиков решает сначала разработать дизайн, в то время как другой решает сначала написать код, а третий — часть документации. Это неминуемо приведет к провалу, а значит, необходимо обеспечить должную сферу понимания и взаимодействия для членов команды, чтобы обеспечить создание ожидаемого продукта.

Ниже приведено схематическое представление жизненного цикла ИТ-проекта (рис. 1).



Рис. 1. Жизненный цикл ИТ-проекта

Приведем более подробное описание фаз жизненного цикла ИТ-проекта.

1. Сбор и анализ требований. На этом этапе вся необходимая информация собирается от клиента для разработки продукта в соответствии с его ожиданиями. Любые неясности должны быть разрешены именно на этом этапе. Бизнес-аналитик и руководитель ИТ-проекта назначают встречу с клиентом, чтобы собрать всю информацию, например, что клиент хочет создать, кто будет конечным пользователем, какова цель продукта. Очень важно базовое понимание или знание продукта.

Например, клиент хочет иметь приложение, которое включает в себя денежные операции. В этом случае требование должно быть ясным, например, какие операции будут выполняться, как это будет сделано, в какой валюте это будет сделано и т. д.

После того как сбор требований завершен, проводится анализ для проверки целесообразности разработки продукта. В случае какой-либо неоднозначности необходимо дальнейшее обсуждение с клиентом.

Как только требование становится четким и понятным, создается документ под названием «Спецификация требований к программному обеспечению». Этот документ должен быть полностью понят разработчиками, а также должен быть рассмотрен заказчиком для дальнейшего использования [2].

2. Проектирование и дизайн. На этом этапе в качестве входных данных используется «Спецификация требований к программному обеспечению» и выводится архитектура программного обеспечения, используемая для реализации разработки системы.

3. Реализация (кодирование). Этап начинается, как только разработчик получает проектный документ. Дизайн программного обеспечения переводится в исходный код. На этом этапе реализуются все компоненты программного обеспечения.

4. Тестирование. Начинается после завершения кодирования и выпуска модулей для тестирования. На этом этапе разработанное программное обеспечение тщательно тестируется, и любые обнаруженные дефекты исправляются разработчиками. Повторное тестирование, регрессионное тестирование проводится до тех пор, пока программное обеспечение не будет соответ-

ствовать ожиданиям клиента. Тестировщики ориентируются на «Спецификацию требований к программному обеспечению», чтобы убедиться, что программное обеспечение соответствует стандарту заказчика.

5. Развертывание. После того как продукт протестирован, он развертывается в производственной среде или выполняется первое приемочное тестирование пользователем. В том случае, когда применяется приемочное тестирование, создается копия производственной среды, и заказчик вместе с разработчиками проводит тестирование. Если клиент находит приложение соответствующим ожиданиям, то производится развертывание.

6. Техническое обслуживание. После развертывания продукта в производственной среде разработчики заботятся о его обслуживании, т. е. решают проблемы, которые необходимо устранить, совершенствуют или модернизируют продукт в соответствии с желаниями клиента.

1.2. Модели жизненного цикла ИТ-проекта

Модель жизненного цикла ИТ-проекта — это описательное представление цикла разработки программного обеспечения. Подходы могут быть различные, но основные фазы и действия остаются одинаковыми для всех моделей [3].

Модель водопада

Это самая первая модель, которая используется в описании жизненного цикла ИТ-проекта. Она также известна как линейная последовательная модель. В этой модели результат одной фазы является входным сигналом для следующей фазы. Разработка следующей фазы начинается только после завершения предыдущей фазы. Стадии жизненного цикла остаются прежними.

В первую очередь проводится сбор и анализ требований. Как только это будет выполнено, можно приступить к проектированию системы. Здесь созданный документ «Спецификация требований к программному обеспечению» является выходом для фазы требований и входом для проектирования системы.

На стадии проектирования и дизайна программного обеспечения создаются документы, которые выступают в качестве входных данных для следующего этапа.

На этапе реализации выполняется кодирование, и разработанное программное обеспечение является исходным материалом для следующего этапа, т. е. тестирования.

На этапе тестирования разработанный код тщательно тестируется для выявления дефектов в программном обеспечении. Дефекты регистрируются и повторно тестируются после исправления. Регистрация ошибок, повторное тестирование, регрессионное тестирование продолжается до тех пор, пока программное обеспечение не перейдет в рабочее состояние.

На этапе развертывания разработанный код перемещается в рабочую среду после того, как заказчик дает разрешение на выход.



Рис.2. Модель водопада

Любые проблемы в производственной среде решаются разработчиками, которые проводят техническое обслуживание.

Модель водопада может быть представлена в виде схемы (рис. 2).

Преимущества модели водопада:

- это простая модель, которую легко понять и в которой все этапы выполняются шаг за шагом;
- результаты каждого этапа четко определены, и это не приводит к какой-либо сложности и делает проект легко управляемым.

Недостатки модели водопада:

- требует много времени и не может использоваться в краткосрочных проектах, так как новая фаза не может быть начата до завершения текущей фазы;
- не может использоваться для проектов, которые имеют неопределенные требования или в которых требования постоянно меняются, поскольку эта модель предполагает, что требования будут ясными на самой фазе их сбора и анализа, и любое

изменение на более поздних этапах приведет к увеличению затрат, поскольку изменения потребуются на всех этапах.

V-образная модель

Также известна как модель проверки и валидации. В этой модели верификация и валидация идут рука об руку, т. е. разработка и тестирование идут параллельно. Модель V и модель водопада похожи, за исключением того, что планирование и тестирование в V-модели начинаются на ранней стадии (рис. 3).

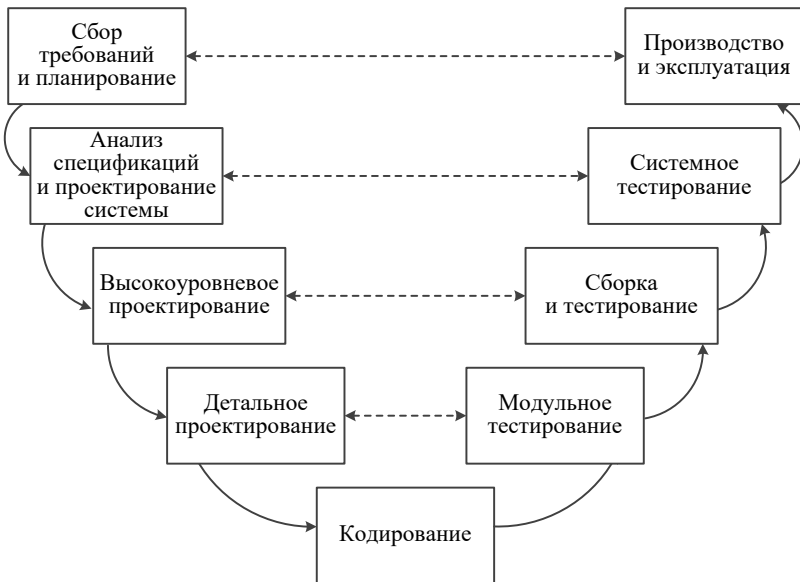


Рис. 3. V-образная модель

Фаза верификации включает этапы (см. рис. 3):

- сбор требований и планирование. На этом этапе собирается и анализируется вся необходимая информация. Мероприятия по проверке включают в себя проверку требований;
- анализ спецификаций и проектирование системы. Результат — требования к системе ясны и система спроектирована, т. е. архитектура и компоненты продукта созданы и зафиксированы в проектной документации;

- высокоуровневое проектирование. Определяет архитектуру и дизайн модулей;
- низкоуровневое проектирование. Определяет архитектуру/дизайн отдельных компонентов;
- кодирование. На этом этапе выполняется разработка кода.

Фаза валидации включает этапы (см. рис. 3):

- модульное тестирование. Выполняется с использованием наборов модульных тестов, которые разработаны на этапе проектирования низкого уровня. Модульное тестирование выполняется самим разработчиком на отдельных компонентах, что приводит к раннему обнаружению дефектов;
 - интеграционное тестирование (сборка и тестирование). Выполняется с использованием интеграционных тестов на этапе проектирования высокого уровня. Это тестирование проводится на интегрированных модулях; его выполняют тестировщики;
 - системное тестирование. Выполняется на этапе проектирования системы. При этом тестируется вся система, вся функциональность системы;
 - итоговые испытания (пуск в производство, эксплуатация).
- Приемочное тестирование связано с этапом анализа требований и проводится в среде заказчика.

Преимущества V-образной модели:

- простая и легко понятная;
- подходит для небольших проектов, в которых требование определено на ранней стадии и не изменяется;
- это систематическая жестко формализованная модель, которая приводит к высококачественному продукту.

Недостатки V-образной модели:

- не подходит для текущих проектов;
- изменение требований на более поздних этапах обходится слишком дорого.

Модель прототипа

Это модель, в которой прототип разрабатывается до фактического программного обеспечения. Прототипные модели имеют ограниченные функциональные возможности и неэффективную производительность по сравнению с реальным программ-

ным обеспечением. Для создания прототипов используются фиктивные функции.

Прототипы программного обеспечения создаются до фактического программного обеспечения, чтобы получить ценную обратную связь от клиента. Когда обратная связь реализована, прототип снова проверяется заказчиком на предмет любых изменений. Этот процесс продолжается до тех пор, пока модель не будет принята заказчиком.

Модель прототипа может быть представлена схематично (рис. 4).



Рис. 4. Модель прототипа

Как только сбор требований завершен, создается дизайн и прототип, который представляется заказчику для оценки.

Отзывы клиентов и уточненные требования используются для модификации прототипа, который снова представляется заказчику для оценки. Как только заказчик утверждает прототип, он используется в качестве технического требования для создания фактического программного обеспечения. В свою очередь, фактическое программное обеспечение строится с использованием водопадной модели.

Преимущества модели прототипа:

- прототип снижает стоимость и время разработки, так как дефекты обнаруживаются гораздо раньше;
- отсутствующая функция, функциональность или изменение требований могут быть выявлены на этапе оценки и реализованы в усовершенствованном прототипе;
- привлечение клиента с начального этапа уменьшает путаницу в требованиях или понимании функциональности продукта.

Недостаток модели прототипа — поскольку клиент участвует в каждом этапе, он может изменить требования к конечному продукту, что усложняет область применения и может увеличить срок поставки продукта.

Спиральная модель

Модель включает в себя итеративный и прототипный подходы. В итерациях следуют фазы спиральной модели. Циклы в модели представляют собой фазу процесса жизненного цикла ИТ-проекта, т. е. самый внутренний цикл — это сбор и анализ требований, который следует за планированием, анализом рисков, разработкой и оценкой. Следующий цикл — проектирование, за которым следует реализация, а затем тестирование.

Спиральная модель состоит из четырех фаз (рис. 5):

1) планирование. Этап включает в себя сбор требований, в ходе которого собирается вся необходимая информация от клиента и документируется. Для следующего этапа создается документ «Спецификация требований к программному обеспечению»;

2) анализ рисков. На этом этапе выбирается наилучшее решение на основе анализа рисков, который проводится путем создания прототипа. Например, риск, связанный с доступом к данным из удаленной базы данных, может заключаться в том, что скорость доступа к данным окажется слишком низкой. Риск может быть устранен путем создания прототипа подсистемы доступа к данным;

3) инжиниринг. После того как анализ рисков завершен, выполняется кодирование и тестирование;

4) оценка. Заказчик оценивает разработанную систему и планирует следующую итерацию.

Преимущества спиральной модели:

- анализ рисков широко проводится с использованием прототипных моделей;
- любое улучшение или изменение функциональности может быть сделано в следующей итерации.

Недостатки спиральной модели:

- подходит только для крупных проектов;
- большое количество итераций потребует дополнительных затрат времени для получения конечного продукта, и стоимость может оказаться высокой.



Рис. 5. Спиральная модель

Итеративная (инкрементная) модель

Модель делит процесс работы над продуктом на короткие составляющие.

Допустим, функция, которая будет разработана в итерации, определена и реализована. Каждая итерация проходит через этапы (анализ требований, проектирование, кодирование и тестирование). Детальное планирование в итерациях не требуется. После завершения итерации продукт проверяется и доставляется клиенту для оценки и обратной связи. Обратная связь с клиентом реализуется в следующей итерации вместе с недавно добавленной функцией. Следовательно, продукт расширяет свою функциональность, и после завершения итераций окончательная сборка содержит все функции продукта (рис. 6).

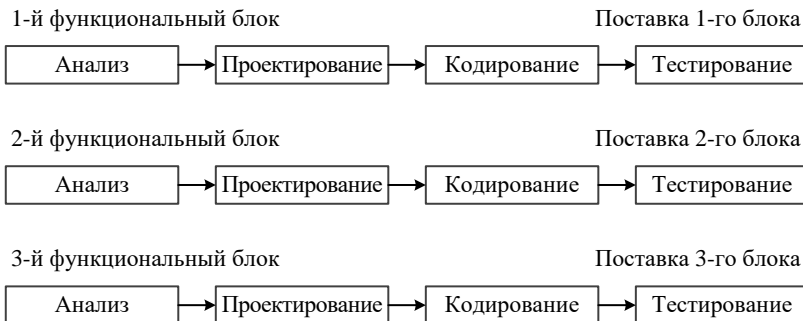


Рис. 6. Итеративная (инкрементная) модель

Этапы итеративной модели разработки:

1) анализ требований. Начальная фаза включает в себя сбор и анализ требований и определение объема проекта;

2) проектирование. На этапе проектирования создается рабочая архитектура продукта, которая покрывает риски, выявленные на начальном этапе, а также удовлетворяет нефункциональным требованиям;

3) кодирование. На данном этапе архитектура заполняется кодом, который готов к развертыванию и создается путем анализа, проектирования, реализации и тестирования функциональных требований;

4) тестирование. На этом этапе продукт развертывается в производственной среде, тестируется, анализируется и клиент дает обратную связь, которая учитывается в следующем функциональном блоке.

Преимущества итеративной (инкрементной) модели:

- любое изменение в требовании может быть легко выполнено без дополнительных затрат, поскольку существует возможность включения нового требования в следующую итерацию;
- риск анализируется и идентифицируется в итерациях;
- дефекты обнаруживаются на ранней стадии;
- поскольку продукт разделен на более мелкие части, им легко управлять.

Недостаток итеративной (инкрементной) модели — как для создания, так и для модификации модели необходим полный перечень требований и понимание продукта.

Модель большого взрыва

Модель не имеет какого-либо определенного процесса. Деньги и усилия объединяются по мере того, как входные и выходные данные поступают в виде разработанного продукта, который может быть или не быть тем, что нужно клиенту.

Модель большого взрыва не требует тщательного и скрупулезного планирования. Разработчик проводит анализ требований и кодирование самостоятельно и разрабатывает продукт в соответствии со своим пониманием. Эта модель используется только для небольших проектов. Нет команды тестирования, формальное тестирование не проводится, и это может стать причиной провала проекта.

Преимущества модели большого взрыва:

- это очень простая модель;
- требуется меньше планирования;
- разработчик обладает гибкостью для создания собственного программного обеспечения.

Недостатки модели большого взрыва:

- не может использоваться для крупных, текущих и сложных проектов;
- высоки риск и неопределенность.

Гибкая модель

Эта модель больше фокусируется на гибкости при разработке продукта, а не на требованиях. Продукт разбивается на небольшие инкрементные сборки. Он не разрабатывается как целостный продукт за один раз. Каждая сборка увеличивается с точки зрения функций. Следующая сборка построена на предыдущей функциональности (рис. 7).



Рис. 7. Гибкая модель¹

В гибкой модели итерации называются спринтами. Каждый спринт длится 2–4 недели. В конце каждого спринта владелец продукта проверяет продукт, и после одобрения он доставляется клиенту. Отзывы клиентов принимаются для улучшения, а предложения и улучшения прорабатываются в следующем спринте. Тестирование проводится в каждом спринте, чтобы свести к минимуму риск любых сбоев.

¹ Сравнение Agile, Scrum, Kanban в качестве эффективных методов управления проектами — блог RDV IT. URL: <https://rdv-it.ru/company/press-center/blog/sravnienie-agile-scrum-kanban/>.

1.3. Инициация проекта

Общеизвестно, что во всех сферах жизни сильный старт может значительно увеличить шансы на достижение успеха. От высоких небоскребов до футбольного матча — всё требует прочного фундамента, и управление ИТ-проектами ничем не отличается. Все проекты начинаются с больших надежд и мотивированной команды. Однако если они не будут инициированы правильно, угроза сбиться с курса и в конечном итоге остаться без направления будет сохраняться на протяжении всего жизненного цикла проекта.

Инициация — это первый этап жизненного цикла управления проектом, возможно, самый важный, и на этом этапе компании решают, нужен ли проект и насколько он будет полезен для них. Для оценки предлагаемого проекта и определения ожиданий от него используются бизнес-кейс и технико-экономическое обоснование.

Данный этап очень важен, поскольку на нем принимаются важные решения, определяющие направление и потребности в ресурсах, такие как устав проекта и выбор заинтересованных сторон. Заинтересованные стороны приходят к четкой цели сформировать единое видение того, как должен развиваться проект.

Во время и после выполнения проекта проводится несколько проверок, чтобы предотвратить недопонимание и обеспечить непрерывность проекта на протяжении всего его хода. Однако драгоценное время и ресурсы могут быть потрачены впустую, что нежелательно.

Эффективное управление проектами требует максимизации выгод и минимизации затрат, обеспечивая при этом «ценность» для клиента.

Основные шаги по иницированию проекта [4]

1. Создание бизнес-кейса. Бизнес-кейс — это важный документ, который объясняет, как цели проекта согласуются с долгосрочными планами компании. В этом документе объясняется, почему компания должна тратить свои технические, финансовые и людские ресурсы на конкретный проект. Идеальный бизнес-

кейс не говорит о каких-либо технических деталях и фокусируется исключительно на бизнес-аспектах. Это делается для того, чтобы убедить высшее руководство одобрить проект и ответить на опасения, связанные с возможными финансовыми и деловыми рисками.

2. Проведение технико-экономического обоснования. Следующим шагом после утверждения бизнес-кейса является определение вероятности успеха проекта после рассмотрения всех факторов. Это исследование определяет ограничения и допущения высокого уровня и решает, стоит ли проект того или нет.

3. Создание устава проекта. Устав проекта, пожалуй, является наиболее всеобъемлющей и важной частью процесса инициирования проекта. Он определяет объем работ, цель, членов команды и возможные временные рамки проекта. В некотором смысле устав является первым документом, в котором определяются необходимые детали проекта, такие как цели и ограничения. В нем также определяется объем проекта и перечисляются необходимые ресурсы для его завершения.

4. Определение заинтересованных сторон и составление реестра заинтересованных сторон. Общение и переговоры являются огромной частью эффективного управления проектами, и большая часть времени руководителя проекта обычно тратится на общение с заинтересованными сторонами. РМВоК [5] определяет как заинтересованных лиц всех, на кого может повлиять или кто имеет влияние на проект. Заинтересованные стороны проекта могут быть как внутренними, так и внешними, и каждый тип имеет свои собственные требования к коммуникации. Общепринятой практикой является ведение реестра или карты заинтересованных сторон для определения частоты и средств связи для каждой заинтересованной стороны в соответствии с их влиянием и интересом к проекту.

5. Сбор команды и создание проектного состава. Ни один проект не может быть начат без проектной команды. Создание рабочей проектной группы и распределение между ее членами ролей и обязанностей является жизненно важной частью этапа инициирования проекта. Распределение ролей и обязанностей на раннем этапе повышает общую подотчетность всей команды и может помочь на более поздних этапах.

6. Окончательный обзор. Это хорошая практика, чтобы просмотреть всю стадию начала проекта и убедиться, что ничего не пропущено. На более поздних этапах вы продолжите пересматривать свою работу, поскольку мониторинг и контроль являются одним из пяти этапов жизненного цикла управления проектом.

1.4. Управление интеграцией ИТ-проектов

Управление интеграцией ИТ-проектов включает процессы и мероприятия по выявлению, определению, объединению, унификации и координации различных процессов и мероприятий в рамках групп процессов управления ИТ-проектами [6].

Управление интеграцией ИТ-проектов — это первая область знаний системы управления проектами, которая помогает поддерживать стабильность проекта. Оно затрагивает все этапы жизненного цикла управления проектами — инициирование, планирование, выполнение, контроль и мониторинг, закрытие. Управление интеграцией помогает руководителю проекта внимательно следить за различными процедурами, выполняемыми на протяжении всего жизненного цикла проекта, от начала до закрытия.

Для руководителя проекта наличие надлежащего плана управления интеграцией очень важно, поскольку это означает, что вся команда работает над достижением общей цели, оставаясь в рамках заданных сроков, объема и бюджета. Управление интеграцией гарантирует, что сроки выполнения различных результатов проекта, его жизненный цикл и план управления годами хорошо согласованы.

План идеально синхронизирует различные процессы. Он помогает в управлении и контроле производительности и изменений необходимых в деятельности и задачах управления проектом. Он координирует решения, касающиеся ключевых изменений, влияющих на проект.

Управление интеграцией измеряет и контролирует прогресс проекта, предпринимая необходимые шаги для достижения поставленных целей. Играет жизненно важную роль в сборе дан-

ных о достигнутых результатах, их анализе для дальнейшего понимания, а затем передаче соответствующим заинтересованным сторонам.

При правильном управлении интеграцией можно плавно завершить все задачи, связанные с проектом, официально закрыть каждую фазу, контракт, проект в целом и высвободить ресурсы.

Вся область знаний по управлению интеграцией проектов разделена на более мелкие процессы, которые выступают в качестве точек доступа для руководителя проекта. Каждый процесс является неотъемлемой частью управления интеграцией проекта и способствует его успеху. Эти процессы следующие:

- 1) разработка устава проекта;
- 2) разработка плана управления проектом;
- 3) руководство и управление проектной работой;
- 4) управление знаниями по проекту;
- 5) мониторинг и контроль проектной работы;
- 6) интегрированное управление изменениями;
- 7) закрытие проекта.

Раскроем каждый из этих процессов и объясним различные входы, выходы и инструменты, используемые в соответствующих процессах.

1. *Разработка устава проекта.* В этом процессе формально разрабатывается документ, который санкционирует существование проекта. Очень важно создать устав проекта до того, как начнется реализация: он будет содержать пошаговый план реализации. С разработкой устава руководитель проекта получает полномочия в отношении различных ресурсов, которые применяются к деятельности по проекту. Разработав устав проекта, можно установить прямую связь между целями организации и осуществляемым проектом. Он также будет служить официальной документацией проекта, которую организация может использовать для подтверждения своей приверженности проекту и убеждения заинтересованных сторон поддерживать проект. Процесс разработки устава проекта обычно включает в себя следующие моменты:

– видение проекта в целом определяет общую цель и миссию проекта, влияние проекта на организацию и конечный результат;

– организация проекта необходима для определения ролей и обязанностей внутри команды, участвующей в разработке проекта, начиная с соответствующих заинтересованных сторон и их отношения с проектом и заканчивая внутренними и внешними человеческими ресурсами и клиентами;

– реализация. План реализации будет постоянно информировать клиентов и заинтересованные стороны о ключевых вехах, изменениях или обновлениях в ходе реализации и о возможных результатах завершения проекта;

– управление рисками позволяет выявить любые потенциальные риски или проблемные области, которые могут помешать бесперебойной реализации проекта.

Входные данные, инструменты и результаты, связанные с процессом интеграции ИТ-проекта, перечислены в табл. 1.

Таблица 1

Процесс интеграции ИТ-проекта: входы, инструменты, выходы

Входы	Инструменты	Выходы
1. Деловые документы: – бизнес-кейс; – план управления. 2. Соглашения 3. Факторы окружающей среды предприятия. 4. Активы организационного процесса	1. Экспертное заключение. 2. Сбор данных: – мозговая атака; – фокус-группы; – интервью. 3. Навыки межличностного и командного общения: – управление конфликтами; – управление собраниями. 4. Встречи	1. Устав проекта. 2. Журнал предложений

2. Разработка плана управления проектом. Процесс включает в себя определение, подготовку и координацию компонентов плана для окончательной интеграции их в структуру управления проектом. План управления проектом действует как дорожная карта для всех членов команды. Это дает им направление для движения вперед, к единой цели для успешной реализации проекта.

План управления проектом включает в себя несколько аспектов:

– первоначальная встреча для мозгового штурма. В ходе этой встречи ключевые заинтересованные стороны собираются

вместе, чтобы обсудить протокол проекта. Это оказывается эффективным способом инициирования самого первого процесса жизненного цикла управления проектом, т. е. планирования, при одновременном укреплении доверия между членами команды проекта;

- объяснение общих целей проекта заинтересованным сторонам. Несмотря на наличие плана управления проектом, изменения неизбежны, и руководитель проекта должен признать этот факт. В ходе этого проекта есть некоторые изменения, которые обязательно произойдут, чтобы скорректировать и преодолеть непредсказуемые проблемы;

- обязанности членов команды и заинтересованных сторон. Наряду с запуском проекта очень важно определить среди заинтересованных сторон, кто будет отвечать за утверждение различных аспектов плана проекта;

- оператор области действия. Заявление о сфере охвата помогает обеспечить спонсорство и определить результаты проекта, чтобы предотвратить любое недопонимание и объединить команду;

- разработка базовых показателей. Прежде чем перейти к этапу разработки проекта, очень важно установить базовые показатели для различных аспектов, таких как затраты, ресурсы, графики, результаты и т. д.;

- создание кадрового плана. Кадровый план — это временная шкала, которая указывает время и продолжительность, в течение которых каждый из сотрудников будет задействован в проекте;

- анализ рисков. Необходим в оценке и смягчении потенциальных рисков, гарантирует, что качество проекта останется неизменным;

- разработка плана коммуникации. Правильный план коммуникации обеспечивает сотрудникам структуру, в которой членам команды выделяются надлежащие точки связи для сообщения о своих проблемах и прогрессе.

Входные данные, инструменты и результаты, связанные с этим процессом, перечислены в табл. 2.

**Процесс разработки плана ИТ-проекта:
входы, инструменты, выходы**

Входы	Инструменты	Выходы
1. Устав проекта. 2. Результаты других процессов. 3. Факторы окружающей среды предприятия. 4. Активы организационного процесса	1. Экспертное заключение. 2. Сбор данных: – мозговая атака; – фокус-группы; – интервью. 3. Навыки межличностного и командного общения: – управление конфликтами; – управление собраниями. 4. Встречи	План управления проектом

3. *Руководство и управление проектной работой.* В соответствии с планом управления проектом этот процесс помогает управлять проектной работой и вносить необходимые изменения для достижения намеченной цели. При правильном направлении и управлении вероятность успеха проекта возрастает при одновременном повышении качества результатов.

Этот процесс осуществляется на протяжении всего жизненного цикла проекта и в основном состоит из следующих аспектов:

– утвержденные запросы на изменения. Любые разрешенные изменения, требуемые или запрошенные в плане, объеме, стоимости или графике проекта, систематически документируются;

– факторы окружающей среды предприятия. Отслеживаются любые внутренние или внешние факторы, которые могут положительно или отрицательно повлиять на конечный результат. Это может быть состояние рынка, инфраструктура, организационная культура или план управления проектами;

– активы организационного процесса. Наряду с факторами, организационные активы, такие как политика, процедуры, формальные и неформальные планы, историческая информация и т. д., которые могут повлиять на конечный результат, должны быть отслежены и оценены.

Входные данные, инструменты и результаты, связанные с этим процессом, перечислены в табл. 3.

Таблица 3

**Процесс руководства и управления проектной работой:
входы, инструменты, выходы**

Входы	Инструменты	Выходы
1. План управления проектом, в том числе любые составляющие. 2. Проектные документы: – журнал изменений; – реестр извлеченных уроков; – список вех; – проектные коммуникации; – график проекта; – матрица исполнения требований; – реестр рисков; – отчет о рисках. 3. Утвержденные запросы на изменение. 4. Факторы окружающей среды предприятия. 5. Активы организационного процесса	1. Экспертное заключение. 2. Информационная система управления проектами. 3. Встречи	1. Результаты. 2. Данные о производительности труда. 3. Журнал проблем. 4. Запросы на изменение. 5. Обновление плана управления проектом, в том числе любые составляющие. 6. Обновления проектных документов: – список мероприятий; – журнал предложений; – реестр извлеченных уроков; – документация по требованиям; – реестр рисков; – реестр заинтересованных сторон. 7. Обновление активов организационных процессов

4. *Управление знаниями по проекту.* Необходимо для достижения поставленной цели проекта и дальнейшего содействия будущему обучению. Управление знаниями осуществляется с использованием исторических или существующих организационных данных и новых знаний. Это в значительной степени помогает улучшить результаты проекта за счет использования организационных знаний. Процесс осуществляется на протяжении всего жизненного цикла проекта и включает в себя различные входные данные, инструменты и результаты (табл. 4).

Управление знаниями по проекту: входы, инструменты, выходы

Входы	Инструменты	Выходы
1. План управления проектом, в том числе любые составляющие. 2. Проектные документы: – реестр извлеченных уроков; – задания проектной группы; – структура разбивки ресурсов; – критерии выбора источника; – реестр заинтересованных сторон. 3. Результаты. 4. Факторы окружающей среды предприятия. 5. Активы организационного процесса	1. Экспертное заключение. 2. Управление знаниями. 3. Управление информацией. 4. Навыки межличностного и командного общения: – активное слушание; – упрощение; – руководство; – сетевая работа; – политическая осведомленность	1. Реестр извлеченных уроков. 2. Обновление плана управления проектом, в том числе любые составляющие. 3. Обновление активов организационных процессов

5. Мониторинг и контроль проектной работы. Этот процесс реализуется для эффективного достижения целей, определенных в плане управления проектом. В процессе управления проект отслеживается и анализируется, сообщается о его общем прогрессе, что позволяет заинтересованным сторонам получить точное представление о состоянии проекта. Этот процесс осуществляется постоянно и позволяет руководителю проекта убедиться, что проект выполняется по расписанию. Некоторые из аспектов этого процесса:

– своевременная корректировка. Регулярные отчеты о производительности и обновление статуса проекта необходимы для достижения успеха;

– пересмотр сферы применения. Время от времени пересмотр сферы применения помогает руководителю проекта убедиться в том, что внесенные изменения были эффективными;

– контроль базовых показателей. Исходные показатели, запланированные в начале проекта, должны строго соблюдаться, и, если будут внесены какие-либо изменения, они должны быть

подробно задокументированы. Это поможет держать команду в фокусе и двигаться в правильном направлении;

– контроль качества. Является одним из основных столпов успеха проекта и должен осуществляться с полным вниманием. Для руководителя проекта важно проводить постоянную оценку его различных компонентов, чтобы обеспечить их эффективность;

– мониторинг и контроль рисков. Отдельный процесс мониторинга и контроля рисков необходим, поскольку риски — это то, что может привести к провалу проекта или отклонению от первоначального результата. Таким образом, оценка новых рисков на каждом этапе помогает на ранней стадии выявить потенциальные угрозы проекту и заранее устранить или смягчить их.

Входные данные, инструменты и выходы, связанные с этим процессом, перечислены в табл. 5.

Таблица 5

**Мониторинг и контроль проектной работы:
входы, инструменты, выходы**

Входы	Инструменты	Выходы
1. План управления проектом, в том числе любые составляющие. 2. Проектные документы: – журнал предложений; – основные оценки; – прогнозы затрат; – журнал проблем; – реестр извлеченных уроков; – список вех; – проект; – отчеты о качестве; – реестр рисков; – отчет о рисках; – примерные расписания. 3. Информация о результатах работы. 4. Соглашения. 5. Факторы окружающей среды предприятия. 6. Активы организационного процесса	1. Экспертное заключение. 2. Анализ данных: – анализ альтернатив; – анализ затрат и выгод; – анализ достигнутых результатов; – анализ первопричин; – анализ тенденций; – дисперсионный анализ. 3. Принятие решений. 4. Встречи	1. Отчеты о выполнении работ. 2. Запросы на изменение. 3. Обновление плана управления проектами, в том числе любые составляющие. 4. Обновления проектных документов: – прогнозы затрат; – журналы проблем; – реестр извлеченных уроков; – реестр рисков; – примерные расписания

6. Интегрированное управление изменениями. Процесс осуществляется для управления различными запросами на изменения, получаемыми на протяжении всего жизненного цикла проекта. Здесь рассматриваются все запросы на изменения, утвержденные изменения, изменения конечного результата, проектные документы, план управления проектом и т. д. Процесс помогает сохранить интегрированный документ, содержащий список изменений, при оценке общих рисков, которые могут возникнуть в связи с новыми изменениями.

Входные данные, инструменты и выходы, связанные с этим процессом, перечислены в табл. 6.

Таблица 6

**Интегрированное управление изменениями:
входы, инструменты, выходы**

Входы	Инструменты	Выходы
1. План управления проектом: – план управления изменениями; – план управления конфигурацией; – области применения; – базовый график; – базовый уровень затрат. 2. Проектные документы: – базовые показатели; – матрица выполнения требований; – отчет о рисках. 3. Отчеты о выполнении работ. 4. Запросы на изменение. 5. Факторы окружающей среды предприятия. 6. Активы организационного процесса	1. Экспертное заключение. 2. Инструменты управления изменениями. 3. Анализ данных: – анализ альтернатив; – анализ затрат и выгод; 4. Принятие решений: – голосование; – единоличное принятие решений; – многокритериальный анализ решений. 5. Встречи	1. Утвержденные запросы на изменение. 2. Обновление плана управления проектом, в том числе любые составляющие. 3. Обновления проектных документов — журнал изменений

7. Закрытие проекта. Это заключительный процесс управления интеграцией проекта, в ходе которого завершаются различные проектные мероприятия, этапы и контракты. Он обеспе-

чивает контролируемую среду, в которой проект может быть успешно завершен. Процесс закрытия включает в себя такие действия, как сохранение информации о проекте, завершение запланированных работ, высвобождение задействованных ресурсов и т. д.

Входные данные, инструменты и выходы, связанные с этим процессом, перечислены в табл. 7.

Т а б л и ц а 7

Закрытие проекта: входы, инструменты, выходы

Входы	Инструменты	Выходы
1. Устав проекта. 2. План управления проектом, в том числе любые составляющие. 3. Проектные документы: <ul style="list-style-type: none"> – журнал предложений; – базовые показатели; – журнал изменений; – журнал проблем; – реестр извлеченных уроков; – список вех; – коммуникация по проекту; – измерения контроля качества; – отчеты о качестве; – документация по требованиям; – реестр рисков; – отчет о рисках. 4. Принятые результаты. 5. Деловые документы: <ul style="list-style-type: none"> – бизнес-кейсы; – план управления прибылью. 6. Соглашения. 7. Закупочная документация. 8. Активы организационного процесса	1. Экспертное заключение. 2. Анализ данных: <ul style="list-style-type: none"> – анализ документов; – регрессионный анализ; – анализ тенденций; – дисперсионный анализ. 3. Встречи	1. Обновления проектных документов — реестр извлеченных уроков. 2. Итоговый запуск продукта, услуги или результата. 3. Заключительный отчет. 4. Обновление активов организационных процессов

1.5. Оценка ИТ-проекта

Всесторонняя оценка проекта, политики, программы или инвестиций с учетом всех его этапов: планирования, реализации и мониторинга результатов, предоставляет информацию, используемую в процессе принятия решений.

Оценки можно разделить с точки зрения целей проекта (национальные или общинные) и оперативных аспектов (мониторинг деятельности по проекту).

Существует также разделение в связи с моментом проведения оценки: до внедрения, текущая оценка во время внедрения и оценка после внедрения.

Для достижения цели экономически эффективного распределения капитала инвесторы используют различные методы оценки рациональности инвестиций. С точки зрения фактора времени методы доходности инвестиционных проектов делятся на: статические (также известные как простые) и динамические (методы дисконтирования).

Статические методы оценки проекта

Характерной особенностью статических методов оценки эффективности инвестиционных проектов является определение соотношения годовых (средних или целевых) поступлений от инвестиций и общих номинальных расходов, необходимых для их реализации.

Эти методы не учитывают влияние времени, соответственно, отдельные значения не дифференцируются в последующие годы, и расчет включает сумму ожидаемых затрат и выгод или средние значения за определенный период. Эти методы лишь приблизительно отражают жизненный цикл проекта и уровень капитальных затрат.

Простые методы рекомендуется использовать в следующих случаях [4]:

- 1) на начальных этапах процесса подготовки проектов, когда отсутствует достаточно подробная и обширная информация об инвестиционном проекте;

- 2) в случае проектов с относительно коротким экономическим жизненным циклом, в которых различные сроки ввода

и эффекты не влияют решающим образом на расчет рентабельности;

3) в случае проектов малого масштаба, когда как затраты, так и эффекты незначительны и не влияют на рыночную позицию и финансовое положение компании.

К наиболее часто упоминаемым и описываемым статическим методам оценки инвестиционных проектов относятся:

- расчет срока окупаемости;
- учет сравнительных затрат;
- счет сравнительной прибыли;
- учет сравнительной рентабельности;
- расчет средней нормы прибыли на инвестиции;
- тест по прошествии года.

Преимущества статических методов:

- доступность для понимания;
- понятная математическая формула;
- простота интерпретации результатов.

Недостатки статических методов:

- не принимается во внимание распределение платежей по времени;
- нет определенности в отношении получения будущих доходов;
- не учитывается стоимость неиспользованных возможностей, связанных с получением будущих доходов;
- не принимается во внимание инфляция, которая снижает реальные доходы населения.

Динамические методы оценки проектов

Для оценки инвестиционных проектов более распространенной является тенденция использования динамических методов, таких как:

- расчет NPV — чистой приведенной стоимости;
- расчет IRR — внутренней нормы прибыли;
- расчет MIRR — модифицированной внутренней нормы доходности;
- метод аннуитета;
- расчет индекса прибыльности;
- расчет NPVR — показателя пересмотренной чистой приведенной стоимости.

Выводы по главе 1

Жизненный цикл проекта включает следующие фазы:

- 1) начальная фаза. Определение цели проекта;
- 2) планирование. Включает подробное планирование шагов необходимых для достижения окончательного решения;
- 3) этап выполнения. План проекта приводится в действие, и работа выполняется, следуя шагам, определенным на этапе планирования;
- 4) закрытие проекта. Подведение окончательных результатов, поставка проектной документации, расторжение контрактов с поставщиками, высвобождение ресурсов проекта.

Основные модели жизненного цикла ИТ-проекта:

- модель водопада, в которой результат одной фазы является входным сигналом для следующей фазы;
- V-образная модель. Верификация и валидация (разработка и тестирование) идут параллельно;
- модель прототипа, в которой прототип разрабатывается до фактического программного обеспечения;
- спиральная модель. Включает итеративный и прототипный подходы. Итеративный подход оперирует фазами спиральной модели. Циклы в модели представляют собой фазу жизненного цикла ИТ-проекта;
- итеративная (инкрементная) модель, делящая процесс работы над продуктом на короткие составляющие;
- модель большого взрыва. Не имеет какого-либо определенного процесса. Деньги и усилия объединяются по мере того, как входные и выходные данные поступают в виде разработанного продукта;
- гибкая модель. При разработке продукта фокусируется на гибкости, а не на требованиях. Продукт разбивается на небольшие инкрементные сборки.

Инициация проекта — это первый этап жизненного цикла управления проектом. На этом этапе компании решают, нужен ли проект и насколько он будет полезен для них.

Основные шаги по инициированию проекта:

- 1) создание бизнес-кейса;
- 2) проведение технико-экономического обоснования;
- 3) создание устава проекта;
- 4) определение заинтересованных сторон и составление реестра заинтересованных сторон;
- 5) сбор команды и создание проектного состава;
- 6) окончательный обзор.

Основные процессы управления интеграцией ИТ-проекта:

- разработка устава проекта;
- разработка плана управления проектом;
- руководство и управление проектной работой;
- управление знаниями по проекту;
- мониторинг и контроль проектной работы;
- выполнение интегрированного управления изменениями;
- закрытие проекта.

При оценке инвестиционной привлекательности ИТ-проектов используются как статические, так и динамические методы.

К статическим методам относятся:

- расчет срока окупаемости;
- учет сравнительных затрат;
- счет сравнительной прибыли;
- учет сравнительной рентабельности;
- расчет средней нормы прибыли на инвестиции;
- тест по прошествии года.

Динамические методы включают:

- расчет NPV — чистой приведенной стоимости;
- расчет IRR — внутренней нормы прибыли;
- расчет MIRR — модифицированной внутренней нормы доходности;
- метод аннуитета;
- расчет индекса прибыльности;
- расчет NPVR — показателя пересмотренной чистой приведенной стоимости.

Вопросы для самоконтроля

1. Назовите и раскройте стадии жизненного цикла ИТ-проекта.
2. Почему понятие жизненного цикла важно при управлении ИТ-разработкой?
3. Какие основные модели жизненного цикла ИТ-проекта применяются? В чем их различие?
4. Какие из моделей жизненного цикла проекта предпочтительнее применять в тех или иных случаях?
5. Что такое инициация ИТ-проекта? Почему эта стадия важна?
6. Назовите и раскройте основные этапы инициации ИТ-проекта.
7. Что понимается под интеграцией ИТ-проекта?
8. Охарактеризуйте основные процессы интеграции ИТ-проекта.
9. Назовите и дайте краткую характеристику основных инструментов, используемых в рамках каждого процесса интеграции.
10. В чем заключаются преимущества и недостатки статических и динамических методов оценки ИТ-проекта?

Глава 2

Планирование ИТ-проекта

2.1. Основные аспекты планирования ИТ-проекта

Планирование — третий этап жизненного цикла управления ИТ-проектами. На данном этапе проводятся различные мероприятия, которые охватывают работу, график, бюджет, ресурсы, потребности в персонале и т. д. Это делается для того, чтобы проект был завершен вовремя и в рамках бюджета. Планируемый проект переходит к фазе выполнения и контроля.

В табл. 8 перечислены основные мероприятия этого этапа и их результаты (документы процесса) [7].

Т а б л и ц а 8

Мероприятия и результаты фазы планирования

Мероприятие	Описание	Результат
Инициирование планирования. Стартовая встреча	Проводится стартовая встреча со всеми участниками этого этапа для определения ожиданий от проекта	
Определение проектных работ	Определяются проектные работы с использованием иерархической структуры проекта	Иерархическая структура проекта
Планирование расписания	Определяется расписание каждого рабочего пакета и их взаимозависимости	График проекта
Планирование ресурсов	Определяется тип и объем ресурсов необходимых для завершения проекта	План ресурсов
Кадровое планирование	Документируется кадровый план, который обеспечивает привлечение соответствующих людских ресурсов с необходимыми навыками	Кадровый план

Мероприятие	Описание	Результат
Бюджетное планирование	Определяются расходы и источники финансирования проекта в течение срока действия проекта	Бюджетный план
Планирование эффективности проекта	Определяется показатель эффективности для каждой бизнес-цели	План выполнения проекта
Планирование управления рисками	Определяется, как можно выявлять, смягчать, управлять и контролировать риски	План управления рисками
Идентификация и учет рисков проекта	Определяются риски проекта, следуя подходу, описанному в плане управления рисками	Реестр рисков
Планирование управления изменениями и конфигурациями	Определяются компоненты проекта, охватываемые процессом управления изменениями, и документация процесса изменений	План управления изменениями и конфигурациями
Планирование закупок	Планируется, как будут управляться различные аспекты закупок с начала и до конца проекта	План закупок
Планирование коммуникаций	Определяются информационные потребности заинтересованных сторон и способы их удовлетворения	План коммуникаций
Планирование управления качеством	Планируются мероприятия по обеспечению качества продукции и проекты, связанные с обеспечением качества	План качества
Разработка плана проекта	Все результаты объединяются в отдельный документ планирования	План проекта

Инициирование планирования. Стартовая встреча

Прежде чем вы начнете какое-либо фактическое планирование, будет проведена стартовая встреча со всеми участниками этого этапа, чтобы установить, чего ожидать от проекта, сформулировать вероятные риски и прояснить неопределенности, которые могут возникнуть у проектной команды.

Стартовая встреча включает:

- 1) краткое представление команды проекта и сторон, участвующих в проекте; описание их роли и обязанностей;
- 2) четкую формулировку продолжительности этапа планирования; определение дат начала и окончания этапа планирования;

3) установку правил и руководящих принципов для этапа планирования. Сообщите команде об ожиданиях. Например, ожидается, что руководитель проекта должен участвовать в любых официальных и неофициальных встречах с заинтересованными сторонами;

4) изучение и обсуждение устава проекта. Убедитесь, что всем ясно видение проекта, основные этапы и т. д.;

5) обсуждение подхода к проекту и сроки. В зависимости от продолжительности обсуждения и необходимости участия заинтересованных сторон, не входящих в проектную группу, оно может быть проведено на отдельной встрече;

6) обсуждение рисков, ограничений и допущений. Обсудите вероятные выявленные риски, которые могут возникнуть на этом этапе, и сообщите команде о стратегиях разрешения. Допустим, если существует риск того, что суровые погодные условия могут повлиять на ход планирования мероприятий, то следует обсудить последствия, прогнозируемые на данном этапе, а также организацию работы. Помимо рисков, важно убедиться, что все участники четко понимают ограничения и допущения планирования проекта;

7) определение ресурсов проекта и способов доступа к ним. Обсудите ресурсы и инструменты необходимые для завершения этой фазы и данного проекта. При необходимости сообщите команде, как можно получить эти ресурсы;

8) вопросы и ответы. Ответьте на любые вопросы, которые могут возникнуть у участников.

Определение проектных работ

Первым шагом является определение проектных мероприятий с использованием иерархической структуры. Цель этой деятельности состоит в том, чтобы установить общее понимание объема проекта.

Пример определения проектных мероприятий с использованием иерархической структуры показан на рис. 8. Это описание работы, которая должна быть выполнена для получения результатов проекта. Корневой элемент — это элемент контекста, который обычно является именем проекта. Каждый нисходящий уровень в иерархии представляет собой все более подробное описание деятельности по проекту. Разработка структуры раз-



Рис. 8. Пример определения проектных мероприятий с использованием иерархической структуры¹

¹ Презентация на тему «Управление содержанием проекта». URL: <https://infourok.ru/prezentaciya-na-temu-upravlenie-soderzhaniem-proekta-2196090.html>.

бивки работ включает разделение основных видов деятельности по проекту или подвидов деятельности на более мелкие, более управляемые, до тех пор, пока эти виды деятельности не будут определены достаточно подробно для поддержки управления и развития проектных работ. Элементы на самом низком уровне ветви называются рабочими пакетами.

Вообще говоря, существует три типичных способа структурирования работ с помощью иерархической структуры. Это структуры, основанные на фазах, структуры, основанные на результатах, и структуры, основанные на ответственности, определяющие деятельность по проекту на основе этапов проекта, согласованных результатов и организационных подразделений, которые будут работать над проектом, соответственно.

Планирование расписания

Включает в себя определение расписания каждого рабочего пакета, как указано в предыдущем действии, и их взаимосвязей с использованием диаграммы PERT. Менеджер проекта может использовать расписание для планирования и реализации задач проекта, а также для мониторинга хода выполнения проекта.

Диаграмма PERT есть стандартный метод анализа оценки программы, который помогает создавать рабочие процессы, контрольные точки, графики, расписания, критические пути и другие требования к планированию проекта. Связи между задачами представляют их зависимости. Вот свойства задач PERT:

- имя — имя задачи, совпадающее с именем соответствующего рабочего пакета;
- идентификатор — однозначно идентифицирует задачу;
- планируемая дата начала и окончания. Задача может быть запущена только в том случае, если предыдущая задача завершена. Другими словами, дата начала задачи должна быть больше даты окончания всех зависимых задач;
- продолжительность — количество дней между датой начала и датой окончания;
- ответственное лицо — лицо, отвечающее за выполнение этой задачи.

Планирование ресурсов

Планирование ресурсов включает в себя определение типа и объема ресурсов необходимых для завершения проекта. При

разработке устава проекта проводится ранняя оценка потребностей в ресурсах. На этом этапе необходимо предоставить подробную разбивку ресурсов.

Типичные ресурсы могут включать материалы, оборудование и аппаратное обеспечение, программное обеспечение и т. д. Кадровые ресурсы будут запланированы в следующем мероприятии, поэтому здесь их можно пропустить. Для каждого из идентифицированных ресурсов описываются следующие атрибуты:

- необходимое количество и качество. Скажем, если принтер является необходимым ресурсом, сколько принтеров вам нужно? Требуется ли какая-либо конкретная функция? Например, цветная печать, двусторонняя печать и т. д.;

- статус распределения. Был ли выделен ресурс? Был ли он зарезервирован? Или просто планировался;

- проектные мероприятия, которые имеют потребность в данных ресурсах;

- время, в течение которого будет потребляться ресурс;

- планируемая дата и время, когда будет выделен ресурс;

- стоимость покупки или аренды этого ресурса;

- единицы измерения стоимости ресурса (за час, за день, за месяц и т. д.);

- уровень риска невозможности приобретения этого ресурса в соответствии с планом.

Кадровое планирование

Кадровый план иногда называют планом людских ресурсов. Цель кадрового плана состоит в том, чтобы добиться успеха проекта путем обеспечения соответствующих людских ресурсов, обладающих необходимыми навыками.

Часть этой деятельности включает в себя определение ролей и обязанностей заинтересованных сторон проекта. Некоторые заинтересованные стороны были определены при разработке устава проекта. На этом этапе необходимо уточнить роли для членов проектной команды. Программист, менеджер по качеству — вот некоторые типичные роли членов проектной команды.

Как только роли заинтересованных сторон определены, необходимо представить их обязанности, связанные с задачами проекта, используя матрицу (диаграмму) RACI. В матрице заинтересованные стороны проекта перечислены в виде строк, а па-

кеты работ, определенные при разработке иерархической структуры проектных мероприятий, в виде столбцов. Подробный разбор матрицы RACI представлен в главе 3.

Бюджетное планирование

Включает в себя обобщение расходов и источников финансирования проекта в течение всего срока его действия путем предоставления сметы расходов или затрат по каждому пакету работ по финансовым годам.

Планирование эффективности проекта

На начальном этапе при разработке устава проекта определяется набор бизнес-целей.

Эффективные цели — это стандарты производительности, которых необходимо достичь. Они помогают узнать, чего ожидает конечный пользователь. Неясный, хаотичный список целей может привести проект к провалу. В управлении проектами используется аббревиатура SMART.

SMART представляет пять качеств, которыми должны обладать цели:

- конкретные — цели должны быть четко сформулированы;
- измеримые — должны существовать конкретные критерии и единицы для измерения прогресса в достижении цели;
- достижимые — реалистичные. Не нужно ставить цели, которые слишком трудны для достижения или выходят за рамки возможностей команды. Если такая цель предложена пользователем и не может быть изменена, вам, возможно, потребуется переоценить другую часть планирования, такую как кадровый план, чтобы увидеть, может ли эта цель быть достигнута при привлечении большего количества ресурсов;
- релевантные;
- ориентированные по времени — необходимо для каждой цели определить крайний срок ее достижения.

В табл. 9 приведены некоторые примеры эффективных целей, соответствующих критериям SMART.

Помимо постановки эффективных целей, данный этап включает в себя детализацию критериев приемлемости для каждого результата. В уставе проекта был определен перечень конечных результатов. Критерии приемлемости для этих результа-

тов будут использоваться для определения или оценки завершения проекта.

Таблица 9

Критерии SMART

Эффективная цель	Соответствие критериям SMART
Страница магазина должна одинаково отображаться в IE 9+, Firefox 40+, Chrome 50+	Цель конкретна, потому что в ней четко указаны типы браузеров, а также их версии, которые должны правильно отображать страницу
Увеличить среднее ежедневное количество просмотров страницы магазина на 20 %	Цель измерима, поскольку требуется, чтобы количество ежедневных просмотров увеличилось на 20 %, что является показателем, который можно использовать для измерения уровня достижения целей. Цель достижима, потому что стремление к улучшению на 20 % — это то, чего может достичь команда
Завершить операцию «Добавить товар в корзину» к полудню 13.06.2021	Цель сориентирована по времени, потому что установлен крайний срок

Планирование управления рисками

Риски оказывают негативное влияние на проект. Поэтому важно принять соответствующие меры для их выявления, смягчения, управления ими и контроля. Планирование управления рисками заключается в выявлении и детализации этих аспектов. Вот мероприятия по управлению рисками, которые необходимо запланировать:

- идентификация — описывает процесс, с помощью которого будут идентифицированы риски. Он должен включать методы, которые команда проекта будет использовать для выявления рисков, а также способ и формат, в котором будут регистрироваться риски;

- приоритизация и категоризация — определяет, как квалифицируются и приоритизируются риски. После выявления рисков важно оценить вероятность и влияние каждого риска, чтобы определить приоритетность риска и спланировать стратегию смягчения последствий. Риски, которые с большей вероятностью возникнут и окажут значительное влияние на проект, будут самыми приоритетными;

- планирование реагирования — описывает, как проектная группа должна реагировать на каждый выявленный риск;
- мониторинг, контроль и отчетность — описывает, как будут контролироваться риски в проекте. Важно обеспечить непрерывный мониторинг рисков на протяжении всего жизненного цикла проекта и определить условия возникновения каждого риска.

Идентификация и учет рисков проекта

Необходимо определить риски проекта, следуя плану управления рисками. Информация о каждом выявленном риске хранится в реестре рисков.

Реестр рисков — это журнал всех выявленных рисков. Он показывает вероятность и влияние рисков на проект, стратегию смягчения последствий и то, когда, по оценкам, риск может возникнуть. Реестр рисков создается на этапе планирования проекта и обновляется на этапе исполнения и контроля с учетом рисков, выявленных в ходе проектной деятельности. Основные элементы реестра рисков:

- уязвимые моменты плана — те, которые будут затронуты в случае возникновения риска;
- вероятность того, что определенный риск возникнет, по шкале от 0 до 10 (10 — самая высокая);
- влияние риска на проект (воздействие) в случае его возникновения — от 0 до 10 (10 — самое сильное);
- оценка риска — определяется путем умножения вероятности на воздействие;
- действия, которые должны быть предприняты в случае возникновения риска, для минимизации последствий;
- признаки раннего предупреждения риска. Определение индикаторов или триггеров, которые указывают на то, что риск вот-вот произойдет или уже произошел;
- ответственный по рискам. Человек, которого руководитель проекта назначает следить за индикаторами или триггерами и управлять реакцией на риск, если риск возникает;
- дата последнего просмотра и обновления конкретного журнала рисков.

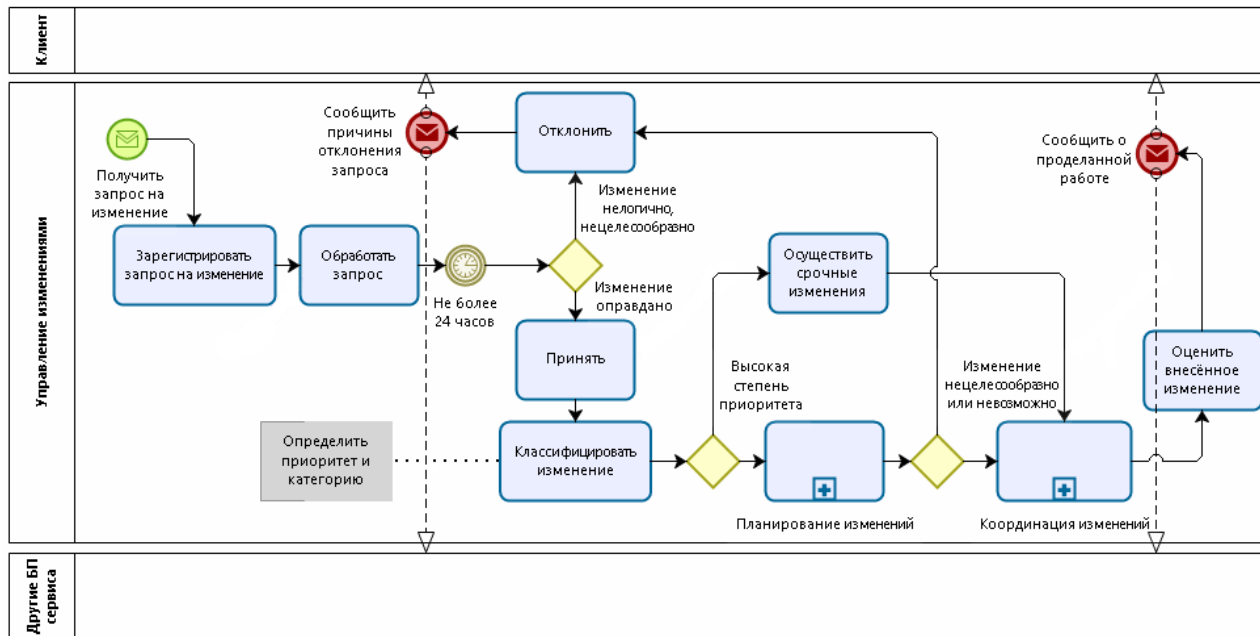


Рис. 9. Схема бизнес-процесса, визуализирующая процесс управления изменениями¹

¹ Дизайн сервиса. URL: <https://is59-2015.susu.ru/terzeman/2018/04/09/2-servisnyiy-dizayn/>.

Планирование управления изменениями и конфигурациями

Управление изменениями — важная часть управления проектами. Изменения должны быть проверяемы и управляемы, чтобы гарантировать, что они находятся в рамках проекта и доведены до сведения всех заинтересованных сторон. Деятельность по управлению изменениями включает в себя идентификацию компонентов проекта, которые регулируются процессом управления изменениями, и документирование процесса изменений.

В планировании управления изменениями и конфигурацией существуют две основные концепции: контроль изменений и контроль управления конфигурацией. Хотя обе они связаны с «управлением изменениями», они представляют собой разные виды деятельности и имеют разную направленность.

Контроль изменений — это процесс выявления, документирования, утверждения (или отклонения) и управления изменениями в таких элементах, как область применения, расписание, бюджет, планы выполнения. Другими словами, управление изменениями действует в базовой линии проекта.

Управление конфигурацией — это процесс управления изменениями конечных результатов проекта и связанных с ними документов на протяжении всего жизненного цикла проекта.

Таким образом, управление изменениями фокусируется на изменении проекта, в то время как управление конфигурацией — на изменении продукта. На рис. 9 показана схема бизнес-процесса, которая визуализирует процесс управления изменениями.

Планирование закупок

Планирование закупок — определение того, как потребности проекта могут быть наилучшим образом удовлетворены путем закупки продуктов и услуг за пределами организации. Включает в себя описание того, как будет организовано управление различными аспектами закупок от начала и до конца проекта. Аспекты:

- процедуры документирования — описывают, как будут задокументированы решения о закупках в рамках проекта;
- проведение закупок — описывает необходимые шаги и обязанности по закупкам в ходе проекта;

– управление поставщиками — определяет роли и действия, которые будут предприняты проектной группой и отделом закупок и контрактов для обеспечения того, чтобы выбранные поставщики предоставляли все согласованные продукты и услуги и поддерживали соответствующий уровень качества.

Планирование коммуникаций

В управлении проектами коммуникация — это способ доставки необходимой информации. У разных заинтересованных сторон проекта разные потребности в информации. Например, исполнительный спонсор может захотеть узнать о том, как продвигается проект. Важно, чтобы коммуникация осуществлялась гладко, чтобы информация могла быть доставлена заинтересованным сторонам по мере необходимости.

Для того чтобы участники проекта, клиенты и другие заинтересованные стороны имели информацию необходимую им для выполнения своих обязанностей, и могли управлять ожиданиями заинтересованных сторон относительно того, как продвигается проект, важно документировать информационные потребности и способы предоставления информации. Этот процесс состоит из трех основных этапов:

1) определение информационных потребностей заинтересованных сторон. Необходимо определить людей и целевую группу, которым нужно получить определенный вид информации, обозначить информационные потребности каждого из участников и заинтересованных сторон;

2) определение способов сбора и оформления информации. Для каждой потребности в информации, определенной на первом этапе, необходимо определить ее цель, поставщика информации, когда и как она будет собираться, как будет доставлена;

3) определение средств доставки информации. Для каждого отчета или документа необходимого для передачи информации нужно определить методы доставки. Типичные примеры включают электронную почту, устную или конференц-связь, встречу, записку, информационный бюллетень, веб-сайт, официальную презентацию, отчет о состоянии и т. д. Также надо предусмотреть способы распространения для каждого отчета или документа.

Планирование управления качеством

Планирование мероприятий по обеспечению качества продукции и проектов включает три основные части:

1) планирование мероприятий по обеспечению качества продукции. Определяются планы управления в процессе производства, которые касаются областей контроля качества;

2) планирование мероприятий по обеспечению качества, связанных с проектом. Определяется, как часто план проекта будет пересматриваться для проверки степени выполнения задач и любого влияния извне;

3) планирование мероприятий по независимой проверке и валидации. Эти действия описываются как «независимые», поскольку будут выполняться незаинтересованной третьей стороной.

Разработка плана проекта

План проекта представляет собой комбинацию отдельных документов планирования, полученных в результате выполнения всех предусмотренных на этом этапе мероприятий. Документ плана проекта должен быть утвержден соответствующим органом.

2.2. Определение содержания ИТ-проекта

Разработка ИТ-проекта требует огромных усилий и является очень сложной. Многие организации приходят к такому выводу, когда они находятся на полпути к осуществлению, поскольку у них отсутствует содержание (программа), обеспечивающее основу, где можно использовать и управлять выявленными преимуществами и следить за ними. Управление содержанием сосредоточивает внимание на следующем:

- выбор набора мероприятий, каждое из которых необходимо для достижения конечного состояния;
- определение контрольных параметров мероприятий;
- обеспечение инфраструктуры, в которой обозначенные мероприятия могут быть успешно выполнены.

Программа ИТ-проекта успешна, когда она реализует преимущества, определенные организацией, в сроки и в рамках

бюджета. Успех программы в целом в значительной степени зависит от успешного выполнения каждого из мероприятий в рамках программы. Мероприятия можно сравнить со строительными блоками: если они не будут выполнены в срок (блоки не будут поставлены в срок), весь проект в конечном итоге потерпит неудачу [8].

Основными инструментами и технологиями определения содержания ИТ-проекта являются [5]:

1) экспертные заключения. Должны рассматриваться отдельными лицами или группами, имеющими специальные знания или подготовку в отрасли ИТ и в области применения разрабатываемого ИТ-проекта, а также имеющими опыт управления или ключевого участия в аналогичных проектах;

2) анализ данных. Метод анализа данных, который может быть использован для этого процесса, включает и анализ альтернатив. Оцениваются различные способы сбора требований, разработки проекта и объема продукта, создания продукта, проверки объема и контроля объема;

3) рабочие встречи. В число участников могут входить руководитель проекта, спонсоры проекта, члены проектной группы, выбранные представители заинтересованных сторон, а также все, кто несет ответственность за любой из процессов управления ИТ-проектом.

Сбор требований к ИТ-проекту — это процесс определения, документирования и управления потребностями и требованиями заинтересованных сторон для достижения целей. Ключевой характеристикой этого процесса является то, что он обеспечивает основу для определения объема продукта и объема проекта. Этот процесс выполняется один раз или в заранее определенных точках проекта. Он является начальным шагом к формированию содержания ИТ-проекта.

Основные инструменты и технологии сбора требований к проекту:

1. Экспертные знания. Необходимо опросить отдельных лиц или группы, имеющие специальные знания или подготовку по таким темам, как:

- бизнес-анализ;
- выявление требований;

- анализ требований;
- документация требований;
- требования к проектам;
- методы построения диаграмм;
- управление конфликтами;
- прочие смежные.

2. Сбор данных. Методы сбора данных, которые могут быть использованы для этого процесса, включают:

- мозговой штурм — метод, используемый для генерации и сбора множества идей, связанных с требованиями к проекту и продукту;

- интервью — формальный или неформальный подход к получению информации от заинтересованных сторон путем прямого общения с ними. Обычно это делается путем постановки подготовленных и спонтанных вопросов и записи ответов. Опрос опытных участников проекта, спонсоров, других руководителей и экспертов по предмету может помочь в выявлении и определении особенностей и функций желаемых результатов продукта;

- фокус-группы, объединяющие квалифицированных заинтересованных лиц и экспертов по предмету, чтобы узнать об их ожиданиях и отношении к предлагаемому продукту, услуге или результату. Обученный модератор проводит группу через интерактивную дискуссию, которая должна быть более разговорной, чем интервью один на один;

- анкеты и опросы — письменные наборы вопросов, предназначенные для быстрого сбора информации от большого числа респондентов. Наиболее подходят для различных аудиторий, когда требуется быстрое изменение, когда респонденты географически рассредоточены и когда может быть уместен статистический анализ;

- бенчмаркинг — сравнение фактических или запланированных продуктов, процессов и практик с аналогичными в других организациях, чтобы выявить лучшие практики, генерировать идеи для улучшения и обеспечить основу для оценки эффективности. Организации, сравниваемые в ходе бенчмаркинга, могут быть внутренними или внешними.

3. Анализ данных, в том числе анализ документов. Используется для выявления требований путем анализа существующей документации и выявления информации, относящейся к требованиям. Примеры документов, которые могут быть проанализированы:

- соглашения;
- бизнес-планы;
- документация по бизнес-процессам или интерфейсам;
- текущие потоки процессов;
- маркетинговая литература;
- нормативная документация;
- запросы предложений;
- примеры использования.

4. Принятие решений. Методы принятия решений, которые могут быть использованы в процессе сбора требований, включают:

– голосование — метод коллективного принятия решений и процесс оценки, имеющий множество альтернатив с ожидаемым результатом в виде будущих действий. Можно использовать для создания, классификации и приоритизации требований к продукту;

– автократическое принятие решений. В этом методе один человек берет на себя ответственность за принятие решения за группу;

– многокритериальный анализ решений — метод, который использует матрицу решений для обеспечения систематического аналитического подхода к установлению критериев, таких как уровни риска, неопределенность и оценка, для оценки и ранжирования многих идей.

5. Представление данных. Методы представления данных, которые могут быть использованы для этого процесса, включают:

– диаграммы сходства — позволяют классифицировать большое количество идей по группам для обзора и анализа;

– ментальное картирование — объединяет идеи, созданные в ходе индивидуальных сеансов мозгового штурма, в единую карту, чтобы отразить общность и различия в понимании и генерировать новые идеи.

6. Межличностное и командное общение. Приемы межличностного общения и работы в команде включают:

- метод номинальной группы. Усиливает мозговой штурм с помощью процесса голосования, используемого для ранжирования наиболее полезных идей для дальнейшего мозгового штурма или для определения приоритетов. Перед группой ставится вопрос или проблема. Каждый человек молча генерирует и записывает свои идеи. Каждая записанная идея обсуждается до тех пор, пока все члены группы не получают ее четкого понимания. Люди голосуют в частном порядке, чтобы определить приоритетность идей, обычно используя шкалу от 1 до 5. Голосование может проходить в несколько раундов, чтобы сократить количество идей и сосредоточиться на них. После каждого раунда подсчитываются голоса и отбираются идеи, набравшие наибольшее количество очков;

- наблюдение или беседа. Обеспечивают прямой способ наблюдения за людьми в их окружении и за тем, как они выполняют свою работу, задачи, процессы. Это особенно полезно для детальных процессов, когда люди, которые используют продукт, неохотно формулируют свои требования. Обычно осуществляется путем внешнего наблюдения за бизнес-экспертом, выполняющим работу. Это также может быть сделано «участником-наблюдателем», который фактически выполняет процесс или процедуру, чтобы испытать, как это делается, и выявить скрытые требования;

- фасилитация. Используется в ходе целенаправленных сессий, где ключевые заинтересованные стороны собираются вместе для определения требований к продукту. Благодаря своей интерактивной групповой природе хорошо организованные сессии могут укрепить доверие, улучшить отношения и коммуникацию между участниками, что может привести к консенсусу заинтересованных сторон. Кроме того, проблемы могут быть обнаружены раньше и решены быстрее, чем в отдельных сеансах.

7. Контекстная диаграмма. Является примером модели области. Контекстные диаграммы визуально отображают область применения продукта, показывая бизнес-систему (процесс, оборудование, компьютерную систему и т. д.) и то, как люди и другие системы (субъекты) взаимодействуют с ней. Контекстные

диаграммы показывают входы в бизнес-систему, субъекта (субъектов), предоставляющего входные данные, выходы из бизнес-системы и субъекта (субъектов), получающего выходные данные.

8. Прототипы. Прототипирование — это метод получения ранней обратной связи по требованиям путем предоставления модели ожидаемого продукта до его фактического создания. Примерами прототипов являются мелкомасштабные продукты, компьютерные 2D- и 3D-модели, макеты или симуляторы. Прототипы позволяют заинтересованным сторонам экспериментировать с моделью конечного продукта, а не ограничиваться обсуждением абстрактных представлений. Прототипы поддерживают концепцию прогрессивной разработки в итеративных циклах создания макета, экспериментирования с пользователями, генерации обратной связи и пересмотра прототипа. Если было выполнено определенное количество циклов обратной связи, требования, полученные от прототипа, являются достаточно полными, чтобы перейти к этапу проектирования или сборки.

В процессе определения содержания ИТ-проекта выбираются окончательные собранные требования. Затем разрабатывается подробное описание проекта и продукта, услуги или результата. Подготовка подробного заявления о масштабах проекта базируется на основных результатах, допущениях и ограничениях, которые документируются во время начала проекта. В ходе планирования определяется объем проекта и описывается с большей конкретностью по мере того, как становится известно больше информации о проекте. Существующие риски, допущения и ограничения анализируются и при необходимости добавляются или обновляются. Процесс определения области действия может быть очень итеративным. В итеративных проектах жизненного цикла разрабатывается видение высокого уровня для всего проекта, но детальный объем определяется по одной итерации за раз, и детальное планирование следующей итерации осуществляется по мере продвижения работы над текущим объемом проекта. Основными инструментами и технологиями здесь являются экспертное заключение, анализ данных, принятие решений, межличностное и командное общение (в той же мере, что и на этапе сбора требований), а также такой инструмент, как анализ ИТ-продукта.

Анализ ИТ-продукта включает в себя вопросы о будущем продукте или услуге и формирование ответов для описания использования, характеристик и других соответствующих аспектов того, что будет поставлено. Примеры методов анализа продукта включают: разбор продукта; анализ требований; системный анализ; проектирование систем; анализ стоимости; ценностный инжиниринг.

В итоге в качестве содержания проекта мы получаем описание объема проекта, основных результатов, допущений и ограничений, которые сводятся в документ под названием «Требования к содержанию ИТ-проекта». В нем подробно описываются результаты проекта. Он также обеспечивает общее понимание содержания проекта среди заинтересованных сторон. С его помощью осуществляется руководство работой проектной группы. Он обеспечивает базовую основу для оценки того, содержатся ли запросы на изменения или дополнительные работы в пределах или за пределами границ проекта.

Требования к содержанию ИТ-проекта включает следующие составляющие:

- описание области применения продукта. Уточняет и дополняет характеристики продукта, услуги или результата, изложенного в уставе проекта;
- результаты. Любые уникальные и ценные находки, методы, способы или возможности, которые должны быть реализованы в рамках определенного процесса, этапа или проекта;
- критерии принятия. Набор условий, которые должны быть выполнены для достижения результатов;
- исключения из проекта. Определяют, что исключается из проекта. Явное указание того, что выходит за рамки ИТ-проекта, помогает управлять ожиданиями заинтересованных сторон и может уменьшить размытие срока завершения.

Устав проекта и требования к содержанию ИТ-проекта различаются по уровню детализации. Устав проекта содержит информацию высокого уровня, в то время как требования к содержанию ИТ-проекта дают подробное описание компонентов области. Эти компоненты постепенно разрабатываются на протяжении всего проекта.

2.3. Методы формирования списка операций проекта

Для определения списка операций ИТ-проекта помимо иерархической структуры используют следующие инструменты и методы:

- декомпозиция;
- шаблоны;
- планирование методом набегающей волны;
- экспертная оценка.

Метод декомпозиции

Декомпозиция, также известная как факторинг, относится к процессу, с помощью которого сложная проблема или система разбивается на части, которые легче понять, запрограммировать и поддерживать [9].

Существуют различные типы декомпозиции, определенные в компьютерных науках:

- алгоритмическая декомпозиция (в структурном программировании) разбивает процесс на четко определенные этапы;
- структурированный анализ разбивает программную систему с уровня системного контекста на системные функции и объекты данных (в соответствии с описанием Тома ДеМарко);
- объектно-ориентированная декомпозиция разбивает большую систему на постепенно уменьшающиеся классы или объекты, которые отвечают за некоторую часть проблемной области;
- алгоритмическая декомпозиция является необходимой частью объектно-ориентированного анализа и проектирования, однако объектно-ориентированные системы следует начинать разрабатывать с декомпозиции на классы.

Планирование методом набегающей волны

Метод управления проектами, который включает в себя постепенную доработку и добавление деталей в иерархическую структуру заданий проекта на постоянной основе. В самом начале проекта краткосрочные результаты раскладываются на отдельные компоненты (рабочие пакеты) и определяются с наибольшей степенью детализации. Результаты и график мероприятий, которые будут проводиться в течение нескольких отчетных

периодов в будущем, определяются менее детально. Например, фазы 1–2 могут быть полностью отражены в иерархической структуре, а фазы 3–6 могут быть очерчены только на уровне подпроектов. Затем, пока ведутся запланированные мероприятия для этапа 1, начинается детальное планирование этапа 3. Как только фаза 2 будет запущена, начнется планирование фазы 4 и т. д. (рис. 10).

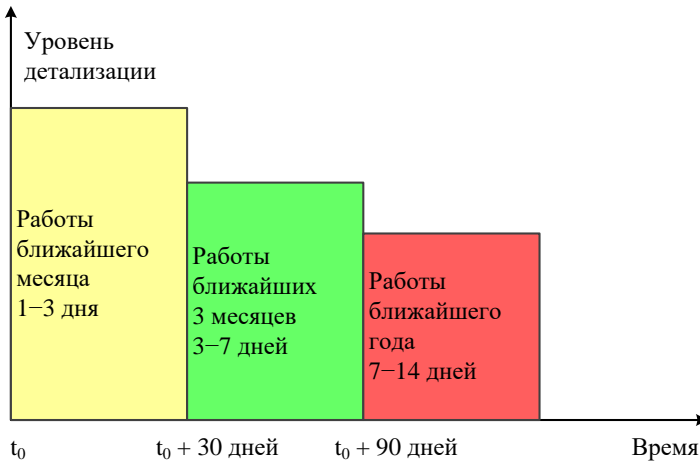


Рис. 10. Планирование методом набегающей волны

Таким образом, планирование методом набегающей волны позволяет рабочим мероприятиям продвигаться вперед по текущим и краткосрочным результатам, в то время как планирование будущих пакетов работ все еще продолжается. Такой подход к управлению проектами особенно полезен, когда наличие информации необходимой для детального планирования будущих пакетов работ зависит от успешного завершения предыдущих этапов проекта. Этот метод также может помочь сократить время завершения проекта за счет двух аспектов:

- возможно начать продуктивную деятельность, не дожидаясь, пока каждая деталь проектной работы будет определена;
- устраняется время простоя при дополнительном планировании в середине проекта, поскольку планирование выполняется непрерывно.

Управление проектами, осуществляемое с использованием метода набегавшей волны, должно с самого начала иметь четко определенный объем проекта. В противном случае, по мере разработки планирования, вероятно, произойдет размытие сроков выполнения проекта.

2.4. Метод предшествования — определение взаимосвязи операций

Метод предшествования (Precedence Diagramming Method—PDM) — это стратегия разработки сетевой диаграммы расписания проекта, которая использует узлы для представления действий и связывает их с элементами, иллюстрирующими зависимости. Метод также называется activity-on-node (AON) [10].

Схема сети с использованием метода предшествования показана на рис. 11.

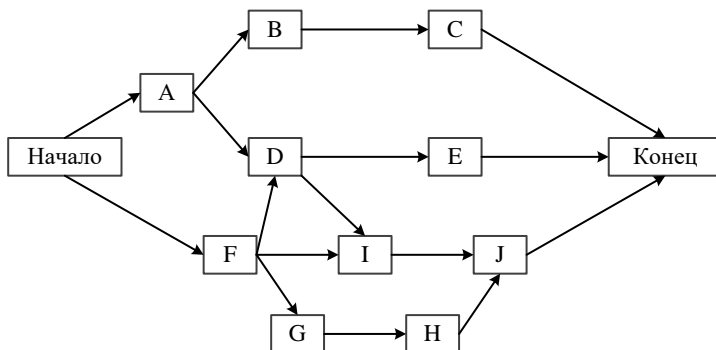


Рис. 11. Схема сети с использованием метода предшествования

Ссылаясь на конкретный метод управления проектом, проектная группа использует процедуру построения графиков сети для графического представления любых признанных и ранее существовавших действий по расписанию с использованием узлов.

Наиболее существенным преимуществом использования метода предшествования является то, что он позволяет проект-

ной команде быстро освоить все запланированные мероприятия и порядок взаимодействия членов команды друг с другом.

Для построения диаграмм по методу предшествования предварительным этапом планирования является работа над иерархической структурой заданий. Зная, насколько важно выполнить задание вовремя, также необходимо знать, в каком порядке должен быть завершен проект. Для этого нам нужно создать диаграмму по методу предшествования.

Руководитель проекта должен убедиться, что он уделяет особое внимание схеме сети проекта. По мере того как выполняются действия, могут возникнуть дополнительные задачи, которые необходимо выполнить, и так мы можем включить их эквиваленты в диаграмму предшествования и иерархическую структуру задач. Далее требуется соединить все задачи (узлы) проекта и убедиться, что не осталось никаких изолированных задач (рис. 12).



Рис. 12. Схема сети процесса кладки кирпича

Важным результатом последовательных действий является схема сети. В последовательности те действия, которые объяс-

няются в процессе планирования, упорядочиваются как конкретные действия, зависящие от других. В то время как схема сети показывает деятельность по проекту и представляет взаимосвязи, построение диаграмм по методу предшествования является наиболее широко признанной стратегией построения сетевых диаграмм.

При использовании диаграмм по методу предшествования в действиях для завершения сетевой диаграммы проекта используются четыре типа связей:

- «от конца к началу» (Finish-to-Start, FS) — наиболее распространенный тип зависимости, используемый между действиями. Если определенные действия не могут начаться до завершения предшествующего действия, то между ними должна присутствовать зависимость «от конца к началу»;

- «от начала к началу» (Start-to-Start, SS) — это своего рода зависимость, показывающая, что два действия решено начать вместе;

- «от конца к концу» (Finish-to-Finish, FF) — в проекте иллюстрирует, что два действия решено завершить вместе;

- «от начала к концу» (Start-to-Finish, SF) — уникальный тип зависимости, он может быть использован мгновенно, например, вместе с материалами цепочки поставок. В этом типе зависимости действие Б может завершиться только после начала действия А.

Также при построении диаграмм по методу предшествования используются следующие типы зависимости:

- 1) обязательная зависимость, известная как «жесткая логика», является неотъемлемой частью метода предшествования. Например, вы работаете над проектом, но не можете протестировать интерфейс, который еще не полностью разработан. Следовательно, в этом случае существует обязательная зависимость между разработкой и тестированием интерфейса;

- 2) дискреционная зависимость, также называемая преференциальной или «мягкой логикой», которая контролируется командой проекта и может быть изменена для сокращения проекта. В этой технике действия остаются прежними, но порядок меняется;

3) внешняя зависимость исходит извне проекта, например, правительственное регулирование, которое, как ожидается, завершится прежде, чем организация приступит к незавершенным мероприятиям по проекту;

4) внутренняя зависимость включает в себя отношение приоритета между действиями проекта. Например, если есть действия, назначенные члену проектной группы, он не может начать работать над этим, пока не завершится предыдущий проект, даже если нет приоритетной обязательной связи.

Метод визуального представления, используемый для подготовки сетевых диаграмм расписания проекта, представляет собой пример метода построения диаграмм по методу предшествования, который анализирует базисные характеристики (рис. 13).

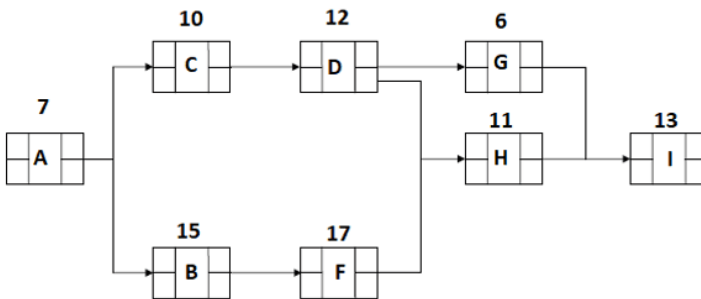


Рис. 13. Метод визуального представления¹

В приведенном выше примере действия обозначаются как A, B, C, D, F, G, H, I. Продолжительность — это числа, указанные выше в графах. Действие B — это предшествующее действие, которое логически предвещает зависимое действие F в этой сетевой системе. Действие F является последующим зависимым действием, которое неизбежно возникает после действия B. Таким образом, между действиями B и F существует связь типа «от конца к началу».

¹ Источники (рис. 13–16): *Precedence Diagramming Method in Project Management?* URL: <https://www.invensislearning.com/articles/pmp/precedence-diagramming-method>.

Для построения диаграммы по методу предшествования необходимо выполнить последовательность действий.

Шаг 1. Расчет времени прямого прохода — необходимо указать минимальные даты, в которые может быть выполнено каждое действие, и, в конечном счете, минимальную продолжительность проекта (рис. 14).

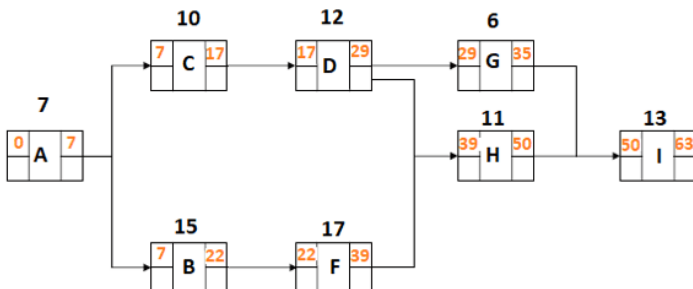


Рис. 14. Построение диаграммы по методу предшествования: шаг 1

Шаг 2. Расчет времени обратного прохода — необходимо определить последние даты, к которым каждое действие может быть выполнено без увеличения минимальной продолжительности проекта (рис. 15).

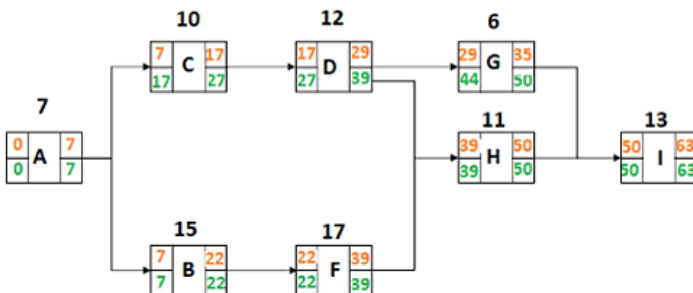


Рис. 15. Построение диаграммы по методу предшествования: шаг 2

Шаг 3. Расчет плавающих сроков для каждого мероприятия.

Общая плавающая величина срока мероприятия — это количество времени, в течение которого действие может быть от-

ложено без задержки даты завершения проекта. Суммарная величина плавающих сроков должна остаться равной 0 на критическом пути.

Например, пользуясь диаграммой на рис. 15, мы можем рассчитать плавающие сроки для мероприятия С следующим образом.

Плавающий срок начала: раннее начало (LS) – позднее начало (ES) = $17 - 7 = 10$.

Плавающий срок конца: ранний конец (LF) – поздний конец (EF) = $27 - 17 = 10$.

Плавающий срок может быть рассчитан путем вычитания даты раннего начала деятельности из даты ее позднего начала или даты раннего завершения из даты ее позднего завершения.

Шаг 4. Определение оптимального пути.

Проанализировав сетевую диаграмму, мы увидим, что проект возможно реализовать с использованием различных путей, и каждый из этих путей имеет свою продолжительность (рис. 16).

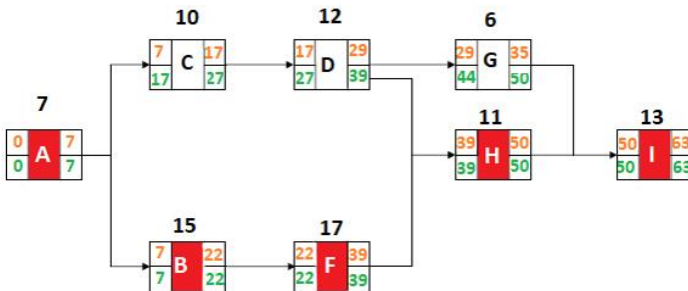


Рис. 16. Построение диаграммы по методу предшествования: шаг 4

Таким образом, построение диаграмм проекта по методу предшествования с его сетью активности на узле и дополнительными отношениями приоритета предлагает более простой и достаточно адаптируемый системный способ представления крупных проектов. В то время как метод предшествования повышает способность планировщика проектов в сроки выполнять крупные проекты, клиенты также должны знать о его основных моментах и быть готовыми адекватно управлять ими.

Выводы по главе 2

Этап планирования — третий этап жизненного цикла управления ИТ-проектами. Он включает мероприятия:

- инициирование планирования;
- определение проектных работ;
- планирование расписания;
- планирование ресурсов;
- кадровое планирование;
- бюджетное планирование;
- планирование эффективности проекта;
- планирование управления рисками;
- идентификация и учет рисков проекта;
- планирование управления изменениями и конфигурациями;
- планирование закупок;
- планирование коммуникаций;
- планирование управления качеством;
- разработка плана проекта.

Инициирование планирования (стартовая встреча со всеми участниками) данного процесса производится с целью установить, чего ожидать от проекта, определить риски и неопределенности, связанные с проектом.

На этапе определения проектных мероприятий формируется иерархическая структура проекта. Цель этой деятельности — установить общее понимание объема проекта.

Планирование расписания включает в себя определение расписания каждого рабочего пакета, на которые был разбит проект, как правило с использованием диаграммы PERT.

Планирование ресурсов включает определение типа и объема материальных, финансовых и прочих ресурсов необходимых для завершения проекта.

Кадровое планирование осуществляется с целью обеспечения каждого рабочего пакета соответствующими человеческими ресурсами, обладающими необходимыми навыками. Данная стадия также включает определение ролей и обязанностей заинтересованных сторон проекта, как правило с использованием матрицы RACI.

Бюджетное планирование включает в себя обобщение расходов и источников финансирования проекта.

На этапе планирования эффективности ИТ-проекта происходит постановка эффективных целей, а также детализация критериев приемлемости для каждого результата.

Планирование управления рисками заключается в выявлении и детализации рисков, определении степени их влияния на проект и планировании мер по смягчению, управлению и контролю рисков.

Этап планирования управления изменениями и конфигурациями включает в себя идентификацию компонентов проекта, которые регулируются процессом управления изменениями, и документирование процесса изменений.

Этап планирования закупок включает в себя описание того, как будет организовано управление различными аспектами закупок от начала и до конца проекта.

Планирование управления качеством предусматривает определение мероприятий по обеспечению качества продукции и проектов.

Разработка плана проекта является заключительной стадией процесса планирования. План представляет собой комбинацию отдельных документов планирования, полученных в результате выполнения каждого этапа.

Управление содержанием ИТ-проекта включает следующие аспекты:

- выбор необходимого набора мероприятий, каждое из которых необходимо для достижения конечного состояния;
- определение контрольных параметров мероприятий;
- обеспечение инфраструктуры, в которой обозначенные мероприятия могут быть успешно выполнены.

Для определения списка операций ИТ-проекта используются следующие инструменты и методы: иерархическая структура; декомпозиция; шаблоны; метод набегающей волны; экспертные оценки.

Основным методом определения взаимосвязи операций является метод предшествования — стратегия разработки сетевой диаграммы расписания проекта.

Вопросы для самоконтроля

1. Назовите основные цели планирования ИТ-проекта.
2. Охарактеризуйте мероприятие «Инициирование планирования» ИТ-проекта.
3. Охарактеризуйте мероприятие «Определение проектных работ» ИТ-проекта.
4. Охарактеризуйте мероприятие «Планирование расписания» ИТ-проекта.
5. Охарактеризуйте мероприятие «Планирование ресурсов» ИТ-проекта.
6. Охарактеризуйте мероприятие «Кадровое планирование» ИТ-проекта.
7. Охарактеризуйте мероприятие «Бюджетное планирование» ИТ-проекта.
8. Охарактеризуйте мероприятие «Планирование эффективности ИТ-проекта».
9. Охарактеризуйте мероприятие «Планирование управления рисками» ИТ-проекта.
10. Охарактеризуйте мероприятие «Идентификация и учет рисков ИТ-проекта».
11. Охарактеризуйте мероприятие «Планирование управления изменениями и конфигурациями» ИТ-проекта.
12. Охарактеризуйте мероприятие «Планирование закупок» ИТ-проекта.
13. Охарактеризуйте мероприятие «Планирование коммуникаций» ИТ-проекта.
14. Охарактеризуйте мероприятие «Планирование управления качеством» ИТ-проекта.
15. Охарактеризуйте мероприятие «Разработка плана ИТ-проекта».
16. В чем суть метода планирования расписания с использованием диаграммы PERT?
17. Каково назначение матрицы RACI?
18. Какие основные составляющие включает управление содержанием ИТ-проекта?
19. Каковы основные аспекты управления содержанием ИТ-проекта?
20. Назовите основные инструменты для определения списка операций проекта.
21. Опишите суть метода предшествования.

Глава 3

Управление расписанием и командой ИТ-проекта

3.1. Инструменты и методы составления расписания ИТ-проекта

Руководителю необходимо использовать различные методы планирования в проекте, чтобы согласовать все его аспекты. Итоговое расписание должно быть соразмерно срокам, установленным для проекта, и все его ресурсы должны использоваться оптимальным образом. Учитывая переменный характер проекта и его масштабы, его трудно спланировать, но сделать это необходимо [10].

Расписание включает все мероприятия, входящие в реализацию и выполнение проекта в заранее определенных временных рамках. Расписание помогает расставить приоритеты в работе, связанной с проектом, и завершить ее упорядоченным образом. Это также помогает в назначении подходящих людей и в правильном распределении имеющихся ресурсов. Управление временем и корректировка объема проекта возможны только при наличии надлежащего расписания.

Основные методы планирования ИТ-проектов

1. Математический анализ. Метод критического пути (CPM) и метод оценки и анализа программ (PERT) наиболее часто используются для планирования расписания ИТ-проектов. Эти методы применяются для расчета временных интервалов в рамках проекта.

Древовидная диаграмма каждого проекта имеет критический путь. Метод критического пути оценивает максимальное и минимальное время необходимое для завершения проекта. СРМ также помогает определить критические задачи, которые должны быть включены в проект. Изменения времени доставки не влияют на график. Объем проекта и перечень мероприятий необходимы для использования СРМ. Затем вычисляется время, затраченное на каждое действие. Идентифицируются все зависимые переменные. Это помогает идентифицировать и разделять независимые переменные.

PERT — это способ планирования потока задач в проекте и оценки общего времени, затраченного на его выполнение. Этот метод помогает представить, как каждая задача зависит от другой. Чтобы запланировать проект с использованием PERT, необходимо определить мероприятия, упорядочить их и определить основные этапы. Вы можете рассчитать сроки для проекта на основе уровня доверия:

- оптимистичное время;
- наиболее вероятное время;
- пессимистичный выбор времени.

Средневзвешенная продолжительность, а не оценки используются PERT для расчета различных временных рамок.

2. Сжатие продолжительности помогает сократить график, если это необходимо. Можно корректировать установленное расписание без изменения области в случае, если проект опаздывает. Две методологии, которые могут быть применены: быстрое отслеживание и сбой.

Быстрое отслеживание — это еще один способ использования СРМ. Быстрое отслеживание находит способы ускорить темпы реализации проекта, либо одновременно выполняя множество задач, либо накладывая многие задачи друг на друга. СРМ помогает определить мероприятия, которые могут быть использованы для ускорения темпов проекта. Хотя это привлекательная техника, у нее тоже есть своя доля рисков. Поскольку многие мероприятия будут осуществляться одновременно, крайне важно не допускать ошибки и идти на компромисс в отношении качества.

Сбой связан с привлечением большего количества ресурсов для своевременного завершения проекта. Для этого вам нужны свободные ресурсы, которые будут в вашем распоряжении. Более того, все задачи не могут быть выполнены путем привлечения дополнительных ресурсов. Необходимость добавления новых членов команды в проект и ограниченная делимость задач приводят к увеличению коммуникации. Техника сбоя также может быть использована путем добавления времени, оплачиваемого сверхурочно, но оно должно оставаться в пределах установленного срока. Это, к сожалению, приводит к удорожанию проекта.

3. Моделирование. Ожидаемая продолжительность проекта рассчитывается с использованием другого набора задач в моделировании. Расписание создается на основе предположений, поэтому его можно использовать, даже если область действия изменена или задачи недостаточно ясны.

4. Эвристика выравнивания ресурсов. Сокращение сроков доставки или предотвращение недостаточного или чрезмерного использования ресурсов путем внесения корректировок в график или ресурсы называется эвристикой выравнивания ресурсов. Разделение задач осуществляется в соответствии с имеющимися ресурсами, чтобы ни один ресурс не был использован недостаточно или чрезмерно. Единственным недостатком этой методики является то, что она может увеличить стоимость и время проекта.

5. Список задач. Это самый простой и популярный метод планирования проекта. Представляет документированный в электронной таблице или текстовом процессоре список всех возможных задач, связанных с проектом. Это очень полезно при реализации небольших проектов. Но для крупных проектов с многочисленными аспектами рассматривать список задач нецелесообразно.

6. Диаграмма Ганта. Это метод визуализации, используемый в управлении проектами для отслеживания прогресса и отчетности. Он используется менеджерами проектов, чтобы получить представление о среднем времени необходимом для завершения проекта. Диаграмма Ганта — это гистограмма, которая представляет ключевые действия в последовательности слева направо по времени. Каждая задача представлена полосой, кото-

рая отражает начало и дату действия, а следовательно, и его продолжительность.

7. Календарь. Большинство календарей могут быть кураторами с собственными именами. В этом случае вы можете создать один календарь для каждого проекта и запланированных событий для этого проекта. В календаре отображается временная шкала для всего проекта. Главное преимущество заключается в том, что он может быть подвергнут изменениям, поскольку доступен для совместного использования. Хотя это кажется отличным методом для отслеживания проекта, у него есть определенные ограничения: вы не можете назначать задачи определенным людям и не можете видеть зависимости задач.

Инструменты планирования могут помочь руководителю проекта правильно распределить ресурсы и завершить проект в заданное время. Они также помогают в подготовке графиков для всех проектов, которые должны быть приняты в будущем. Изучение прошлых результатов работы вашей команды, препятствий, с которыми вы столкнулись при реализации проекта, и того, как эти препятствия были преодолены, очень пригодится. Планирование должно быть неотъемлемой частью управления проектами. Оно помогает не просто сэкономить время, а организовать проект.

3.2. Диаграмма Ганта

Управление почти всегда направлено в будущее. Его задача состоит в том, чтобы выбрать политику и принять меры в соответствии с той политикой, которая приведет к желаемому состоянию. Решения, влияющие на будущее, должны основываться на знании того, что произошло в прошлом. И хотя запись о том, что произошли определенные события или был проделан определенный объем работы, имеет ценность для принятия таких решений, она не дает нам достаточного представления о будущем. Мы должны знать, когда произошли эти события или с какой скоростью работа была сделана. Другими словами, необходимо прояснить отношение фактов ко времени.

Если руководство хочет удовлетворительно управлять ИТ-проектами в условиях постоянно возрастающих трудностей, его решения и действия должны основываться не только на тщательно доказанных фактах, но и на полном осознании значимости этих фактов.

Диаграмма Ганта, благодаря представлению фактов в их связи со временем, является одним из наиболее заметных вкладов в искусство управления.

Использование диаграммы Ганта предполагает наличие плана. Запись этого плана в виде диаграммы, которую могут увидеть другие, делает его более определенным, а также способствует назначению точных задач отдельным людям. План представлен на этих диаграммах настолько четко, что его может понять в деталях и в целом не только сам руководитель проекта, но и те, кто выше его, и его подчиненные.

Диаграмма Ганта информирует руководителя о прогрессе, достигнутом при выполнении плана, и если прогресс не является удовлетворительным, то указывает причины. Время руководителя экономится, потому что каждый раз, когда он получает цифру, ему не нужно сравнивать ее с прошлыми записями и решать, хороша она или плоха. Он раз и навсегда определил, какие цифры будут удовлетворительными, и записал их на диаграмме. Сравнение выполнения с планом, таким образом, становится просто формальной задачей, и руководитель остается свободным для изучения тенденций и принятия мер [11].

Диаграмму Ганта легко нарисовать. Опыт черчения не требуется, так как используются только прямые линии. Она настолько проста, что любой человек со средним интеллектом может быть обучен составлять эти диаграммы. Диаграммы Ганта легко читаются; линии не пересекаются друг с другом, и все записи перемещаются со временем по листу слева направо.

В диаграмме Ганта деление пространства представляется как количество времени и объем работы, которую нужно выполнить за это время. Линии, проведенные горизонтально через это пространство, показывают отношение объема работы, фактически выполненной за это время, к запланированному объему.

Это особенность, которая отличает диаграмму Ганта от всех других диаграмм. Равные деления пространства на одной горизонтальной линии представляют одновременно:

- 1) равные отрезки времени;
- 2) различные объемы запланированных работ;
- 3) различные объемы выполненных работ.

Таким образом, она показывает отношение затраченного времени к проделанной работе. Кроме того, поскольку знание того, что произошло и когда это произошло, вызывает действие, прошлое проецирует себя в будущее, и записи, построенные таким образом, становятся динамичными. Приведем пример.

Допустим, недельная работа запланирована следующим образом:

понедельник — 100 чел.-ч;
вторник — 125 чел.-ч;
среда — 150 чел.-ч;
четверг — 150 чел.-ч;
пятница — 150 чел.-ч.

Лист размечается промежутками, представляющими дни, а объем запланированной работы характеризуют цифры в левой части дневного пространства. Таким образом, график показывает расписание и его связь со временем.

Допустим, фактически выполненная работа:

понедельник — 75 чел.-ч;
вторник — 100 чел.-ч;
среда — 150 чел.-ч;
четверг — 180 чел.-ч;
пятница — 75 чел.-ч.

Составим диаграмму, как показано на рис. 17.

В понедельник промежуток составляет 100 (оранжевая линия); было сделано только 75, поэтому синяя линия проходит через 75 % оранжевой линии. Во вторник было запланировано 125, а сделано 100, таким образом, линия проходит через 80 % и т. д.

Для отображения суммарного фактически выполненного объема работ добавим итоговые линии (рис. 18).

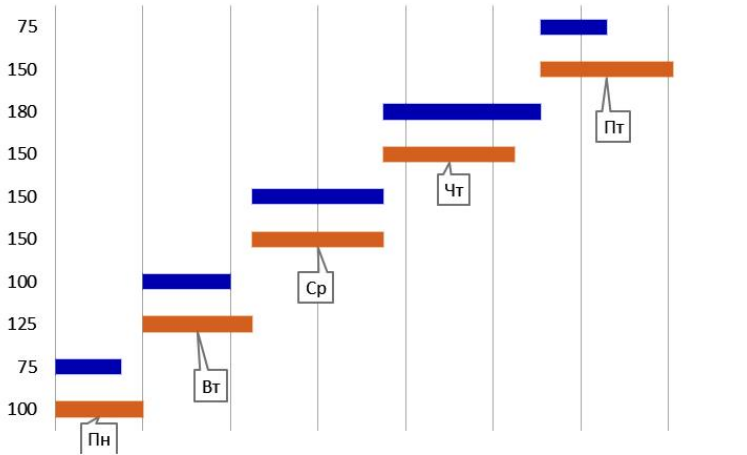


Рис. 17. Диаграмма Ганта, показывающая фактически выполненные объемы работ

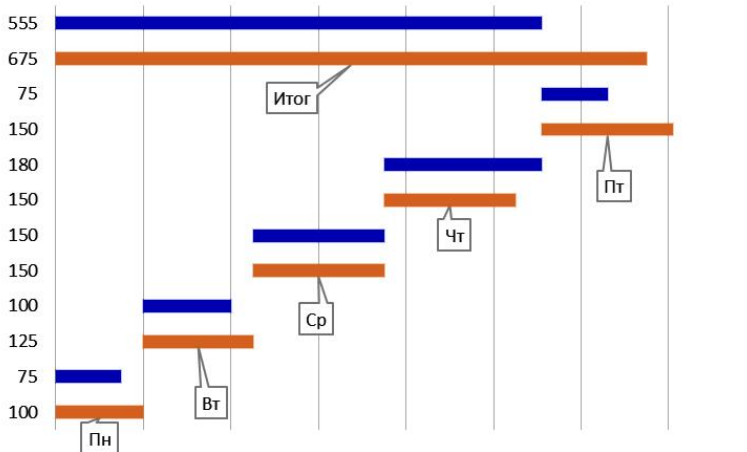


Рис. 18. Диаграмма Ганта, показывающая суммарные фактически выполненные объемы работ

Принцип диаграммы Ганта может быть применен к любой человеческой деятельности. Наиболее широко он использовался в промышленном производстве, но может с успехом применяться в отношении управления ИТ-проектами.

Диаграммы Ганта делятся на три основных класса [10]:

- 1) диаграммы записей человеческого и машинного времени;
- 2) диаграммы компоновки и загрузки;
- 3) диаграммы прогресса.

Диаграмма записей человеческого и машинного времени представляет механизм, показывающий связь между тем, что сделано, и тем, что может быть сделано человеком или машиной.

Разрыв между реальным и возможным достижением показывает трату рабочего времени или его правильное использование. На диаграмме машинного времени показано, когда машина не используется и по какой причине. Диаграмма человеческого времени показывает, правильно ли человек использует свое рабочее время, и если нет, то указывает причину этого. Это позволяет принять меры для предотвращения трат времени.

Диаграмма компоновки — это механизм Ганта для планирования работы таким образом, чтобы избежать трат времени и простоя оборудования и выполнять работу в порядке ее важности.

Диаграмма загрузки показывает объем работы, запланированной к выполнению предприятием или его подразделением. Выполнение плана имеет такое же значение, как и составление этого плана.

Диаграмма прогресса — это механизм Ганта для выполнения работы, показывающий сравнение достижений с планом и причины невыполнения плана.

Диаграмма Ганта упрощает сложную ситуацию или проблему и указывает на действия, которые следует предпринять. Неудивительно, что она широко применяется и в сфере управления ИТ-проектами.

Основная идея управления ИТ-проектами — это успешная сдача проекта в срок. Для достижения этой цели команда применяет соответствующие методы, знания и навыки.

Компании часто используют программное обеспечение для управления проектами, чтобы облегчить их реализацию и сделать этот процесс более прозрачным и эффективным. Это помогает в планировании проектов, отслеживании времени, составлении отчетности, оценке и многом другом.

Одной из самых популярных функций в управлении ИТ-проектами является диаграмма Ганта. По данным SaasList (независимого ресурса по обнаружению и сравнению программного обеспечения), 45 % респондентов утверждают, что диаграммы Ганта являются их наиболее используемым инструментом управления проектами [12].

Они помогают отслеживать даты начала и окончания ИТ-проектов, задачи, которые включают в себя проекты, продолжительности задач, способы объединения задач, сколько времени занимает каждая задача и кто работает над конкретной задачей.

Основные преимущества диаграмм Ганта в управлении ИТ-проектами:

- позволяют быть 24 ч в сутки в курсе того, что происходит в проекте;
- повышают коммуникацию и сплоченность команды;
- показывают прогресс в реализации проекта;
- помогают избежать перерасхода ресурсов;
- демонстрируют перекрывающиеся действия и зависимости задач;
- обеспечивают лучшее управление временем;
- добавляют ясность;
- способствуют мотивации удаленных членов команды;
- повышают гибкость и прозрачность проекта.

Применительно к управлению ИТ-проектами диаграмма Ганта состоит из двух осей и горизонтальных полос, расположенных между ними. Вертикальная ось — список задач. Горизонтальная ось — временная шкала проекта.

Панель представляет проект, задачу или подзадачу, которые должны быть завершены в течение определенного периода времени. Временная шкала составлена в хронологическом порядке, чтобы помочь отслеживать крайние сроки и последовательность выполнения задач. Время от начала первой задачи до завершения последней — это продолжительность всего проекта. Каждая задача назначается одному или нескольким людям. Каждый, у кого есть доступ к проекту, может проверить состояние каждой видимой задачи на временной шкале [12].

Таким образом, диаграмма Ганта является важным инструментом для управления ИТ-проектами. Она обеспечивает

визуальное отображение плана работы. Кроме того, она удобна для мониторинга выполнения задач.

Существует множество различных инструментов для работы с диаграммами Ганта. Онлайн или локально они позволяют сделать рабочий процесс более плавным, а процесс управления — более надежным. Один из них, GanttPRO, является облачным программным обеспечением управления проектами.

Управлять ИТ-проектами с помощью GanttPRO достаточно просто. Программное обеспечение позволяет создавать планы, настраивать временную шкалу, устанавливать зависимости, контролировать рабочую нагрузку и сроки, эффективно распределять ресурсы и взаимодействовать с командой (рис. 19).

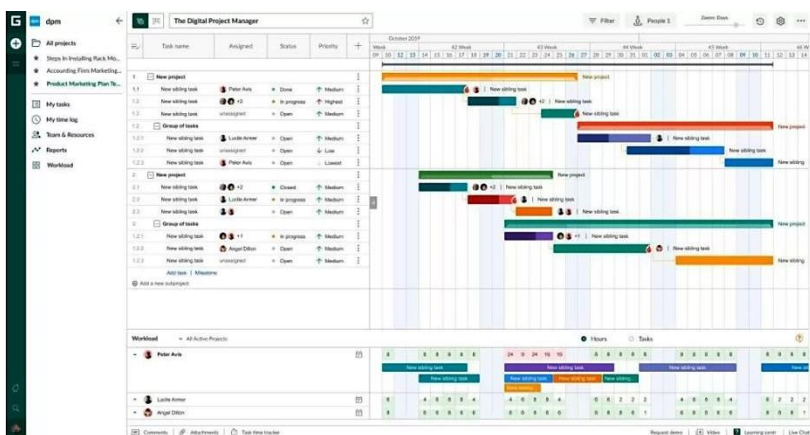


Рис. 19. Облачное приложение GanttPRO¹

На временной шкале можно найти основную и необходимую информацию о нужном проекте:

- название и продолжительность проекта;
- список задач и подзадач;
- имя каждой задачи, приоритет, назначение, ход выполнения;
- крайние сроки, зависимости и вехи;
- загруженность.

¹Источник (рис. 19–21): *Simple Gantt Chart Examples in Project Management*. URL: <https://blog.ganttpro.com/en/gantt-chart-example/>.

В GanttPRO предусмотрены шаблоны диаграммы Ганта для проектов разработки программного обеспечения. Чтобы начать использовать готовый шаблон, требуется выбрать опцию «Создать новый проект», назвать его и выбрать среди множества шаблонов «Разработка программного обеспечения».

После этого проект разработки программного обеспечения будет представлен разделенным на рабочие этапы (рис. 20):

- деятельность по управлению проектами;
- инициация/планирование;
- анализ требований;
- дизайн;
- развитие;
- контроль;
- реализация;
- эксплуатация и техническое обслуживание;
- расположение.



Рис. 20. Построение диаграммы Ганта для проекта разработки программного обеспечения в GanttPRO

Можно настроить шаблон и сделать его более удобным для команды. Можно добавлять или удалять столбцы, изменять

масштаб и задавать цвета для задач, подзадач и панелей проектов. Также можно сохранить свой проект в качестве шаблона и использовать его в будущем.

В GanttPRO имеется шаблон диаграммы Ганта для веб-дизайнерских проектов. По умолчанию он состоит из нескольких рабочих этапов (рис. 21):

- исследование для основного макета;
- генерация концепции;
- цвет и графика;
- презентация контента;
- мультимедиа;
- функциональность;
- прототип 1;
- прототип 2;
- совместимость с браузером;
- доступность;
- доработка и отчеты об ошибках.

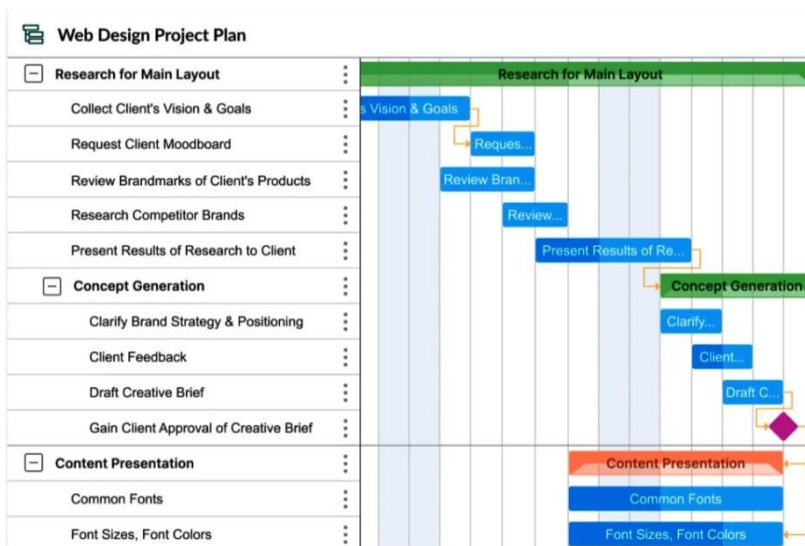


Рис. 21. Построение диаграммы Ганта для веб-дизайнерского проекта в GanttPRO

3.3. Метод критического пути

Метод критического пути — это алгоритм использования ресурсов для планирования набора мероприятий проекта. Морган Р. Уокер и Джеймс Э. Келли разработали его в конце 1950-х годов. Планировщики используют этот метод при разработке графиков для самых разных проектов, включая ИТ, исследования и строительство.

Основой метода критического пути (МКП) является построение модели проекта, которая включает в себя:

- список всех задач необходимых для завершения проекта;
- зависимости между задачами;
- оценку времени (продолжительности), которое потребуется для завершения каждого действия.

С помощью этой информации можно найти критический путь, определив самый длинный отрезок зависимых действий и измерив их от начала до конца.

Критические задачи имеют нулевой резерв времени выполнения. Если продолжительность этих задач изменится, сроки всего проекта будут «сдвинуты». Именно поэтому важнейшие задачи в управлении проектами требуют особого контроля и своевременного выявления рисков.

Анализ критического пути необходим для прогнозирования сроков завершения проекта.

Основные преимущества МКП:

- визуализирует проекты в четкой графической форме;
- определяет наиболее важные задачи;
- экономит время и помогает в управлении сроками;
- помогает сравнить запланированное с реальным статусом;
- определяет все важнейшие виды деятельности, требующие внимания;
- делает зависимости ясными и прозрачными.

Считается, что методология была разработана для рутинных и сложных проектов с возможностью обеспечения минимального изменения времени выполнения задач. МКП теряет свою полезность в более хаотичных проектах.

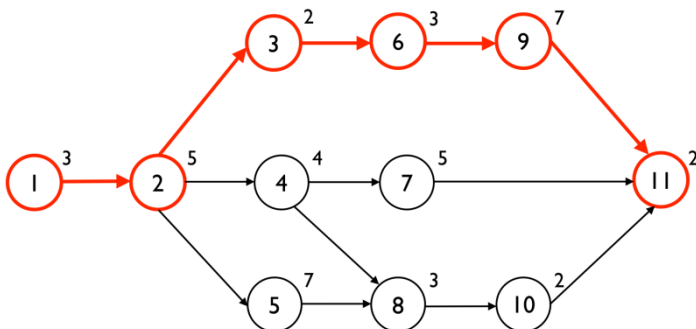


Рис. 22. Нахождение критического пути

Рассмотрим простейший пример применения технологии критического пути к проекту с короткими сроками. Представьте, что наша цель — организовать парковку на пустой асфальтированной площадке рядом с офисом. Для этого нужно выполнить следующие действия:

- 1) выбрать место;
- 2) очистить территорию от мусора;
- 3) купить краску для маркировки;
- 4) измерить площадку для определенного количества автомобилей;
- 5) отметить и покрасить все элементы парковки;
- 6) установить ворота.

Очевидно, что некоторые этапы этого проекта не могут начаться, пока не будут завершены другие. Они зависимы. Шаги 4, 5, 6 являются последовательными действиями, поскольку они должны выполняться в определенном порядке. В данном примере эти этапы являются наиболее важными критическими задачами для решения проблемы. Мы размещаем их на критическом пути проекта, потому что не можем начать какие-либо этапы, пока не будут завершены другие [13].

Основываясь на этом плане, можно определить продолжительность каждого этапа и всего проекта:

- 15 минут на выбор места;
- 90 минут, чтобы очистить территорию;
- 30 минут на покупку краски;
- 45 минут для измерения площади;
- 60 минут для маркировки;
- 60 минут на установку ворот.

Таким образом, ключевые этапы метода критического пути следующие [14].

1. Определение видов деятельности / задач. Зная объем проекта, вы можете разделить структуру работы на список видов деятельности, дав им названия или коды (рис. 23). Все мероприятия в рамках проекта должны иметь продолжительность и конкретную дату.

Имя проекта	Задача 1	Подзадача 1.1	Комплекс работ 1.1.1
			Комплекс работ 1.1.2
		Подзадача 1.2	Комплекс работ 1.2.1
			Комплекс работ 1.2.2
	Задача 2	Подзадача 2.1	
			Комплекс работ 2.1.1
			Комплекс работ 2.1.2

Рис. 23. Метод критического пути: определение видов деятельности¹

2. Определение последовательности. Это самый важный шаг, он дает четкое представление о связях между действиями и помогает установить зависимости, поскольку некоторые действия будут зависеть от завершения других.

3. Создание сетевой диаграммы действий. Определив, какие действия зависят друг от друга, можно создать сетевую диа-

¹ Источник (рис. 23–29): *Beginner's Guide to Critical Path Method (CPM)* / Smartsheet. URL: <https://www.smartsheet.com/critical-path-method/>.

грамму или диаграмму анализа путей. С помощью стрелок вы можете легко подключать действия на основе их зависимостей (рис. 24).

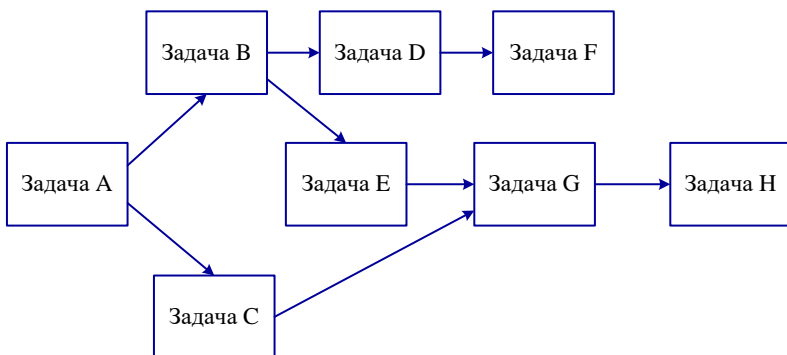


Рис. 24. Метод критического пути: сетевая диаграмма действий

4. Определение временных интервалов для выполнения каждого действия. Оценив, сколько времени будет потрачено на каждое действие, вы сможете определить время необходимое для завершения всего проекта (небольшие проекты можно оценить за несколько дней; более сложные требуют длительной оценки).

5. Нахождение критического пути (рис. 25). Сеть действий поможет вам создать самую длинную последовательность на пути или критическом пути, используя следующие параметры:

- ранний старт — время, когда все предыдущие задачи будут выполнены;

- раннее завершение — ближайшее время начала и время необходимое для выполнения задачи;

- позднее завершение — завершение всех мероприятий без переноса сроков;

- позднее начало — последнее время окончания за вычетом времени необходимого для выполнения задачи.

6. Обновление диаграммы критического пути, чтобы показать прогресс. По мере продвижения проекта вы узнаете фактическое время завершения работы. Затем сетевая диаграмма может быть обновлена, чтобы включить эту информацию (вместо

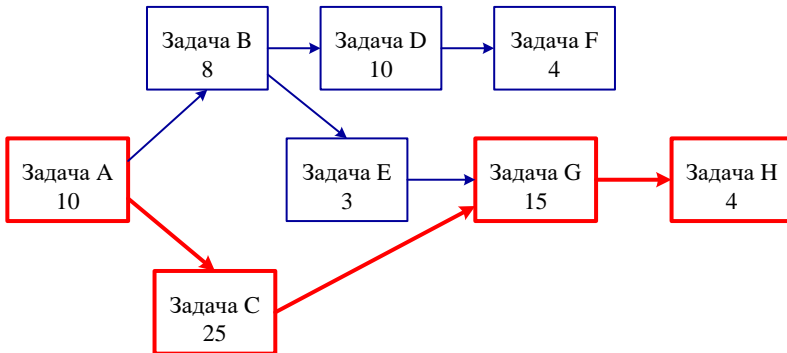


Рис. 25. Метод критического пути: критический путь

того чтобы продолжать использовать оценки). Обновляя сетевую диаграмму по мере появления новой информации, вы можете рассчитать другой критический путь. Вы также будете иметь более реалистичное представление о сроках завершения проекта и сможете определить, находитесь ли вы на верном пути или отстаете.

В настоящее время существует множество автоматизированных средств нахождения критического пути. Одно из таких средств — Microsoft Project.

Например, критический путь в Microsoft Project можно отобразить посредством диаграммы Ганта.

Нужно ввести все свои задачи, даты их начала и окончания, продолжительность каждой задачи и определить предшественников. Затем можно настроить вид файла MS Project для отображения критического пути (рис. 26):

- Выберите *Вид > Диаграмма Ганта*;
- Нажмите кнопку *Формат*, а затем установите флажок

Критические задачи.

Критический путь можно увидеть в любом представлении задачи, выделив его. Перейдите на вкладку *Представление* и выберите нужное в группе *Представление задач*.

Оставаясь на вкладке *Вид*, нажмите *Критический* в списке *Выделение*. Критический путь отображается желтым цветом (рис. 27).

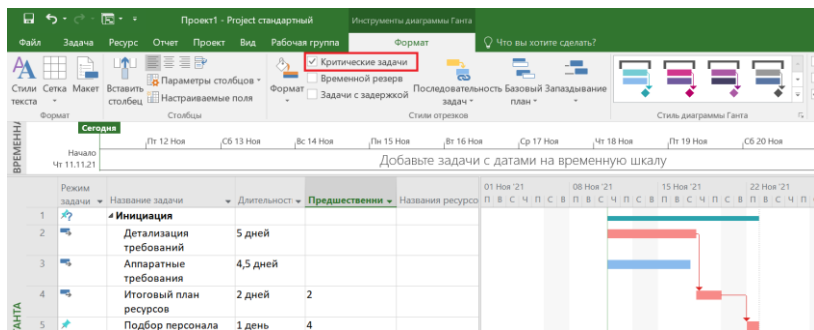


Рис. 26. Нахождение критического пути в Microsoft Project

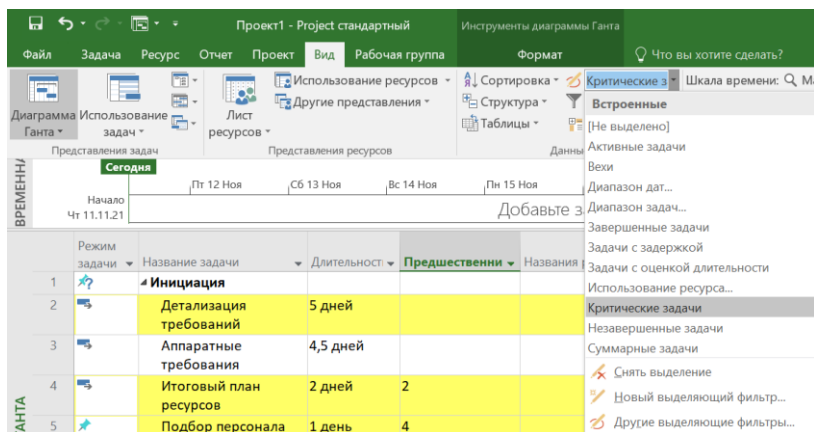



Рис. 27. Отображение критического пути в Microsoft Project

Чтобы просмотреть только задачи на критическом пути, щелкните стрелку фильтра и выберите *Критический*. На диаграмме сети задачи на критическом пути автоматически отображаются красным цветом.

Критический путь можно увидеть также в мастере проектов. При управлении проектом для каждого подпроекта может быть найден критический путь. Чтобы определить критические пути подпроектов, можно настроить Microsoft Project для обработки подпроектов как сводных задач.

Выберите *Файл > Параметры*.

Выберите пункт *Расписание*, а затем прокрутите вниз и выберите *Расчетные параметры проекта* (рис. 28).

Параметры расчета для этого проекта:  Проект1 ▼

☒ Обновлять состояние ресурса при обновлении состояния задачи ⓘ

☒ Рассчитывать вставленные проекты как суммарные задачи

☒ Фактические затраты всегда высчитываются в Project

☐ Распределять изменения итоговых фактических затрат до даты отчета о состоянии


Начисление фиксированных затрат по умолчанию: Пропорционально ▼

Рис. 28. Microsoft Project: расчетные параметры проекта

Можно включить задачи с одним или несколькими днями задержки на критическом пути, чтобы определить любые потенциальные проблемы, которые могут возникнуть.

Для этого:

- выберите *Файл > Параметры*;
- нажмите кнопку *Дополнительно*, затем прокрутите вниз до *Расчетные параметры проекта*;
- измените количество дней в параметре *Считать критическими задачи, имеющие резерв не более* (рис. 29).

Параметры расчета для этого проекта:  Проект1 ▼

☐ Переместить окончание завершенных частей назад на дату отчета о состоянии

☐ А также переместить начало оставшихся частей назад на дату отчета о состоянии

☐ Переместить начало оставшихся частей вперед на дату отчета о состоянии

☐ А также переместить окончание завершенных частей вперед на дату отчета о состоянии

☐ Распределять изменения итогового % завершения для задач до даты отчета о состоянии ⓘ

☐ Рассчитывать несколько критических путей ⓘ

Считать критическими задачи, имеющие резерв не более 1 дн.

Рис. 29. Microsoft Project:
включение задачи с одним или несколькими днями задержки

Метод PERT

Методика оценки и обзора программ (PERT) была впервые разработана в 1958 г. Управлением специальных проектов ВМС

США по ракетной системе «Полярис». Существующее комплексное планирование в таком крупном масштабе было признано недостаточным, поэтому военно-морской флот привлек авиастроительную корпорацию «Локхид» и консалтинговую фирму по управлению «Буз, Аллен и Гамильтон». Традиционные методы, такие как линия баланса, диаграммы Ганта и другие системы, были устранены, и PERT эволюционировала как средство для работы с различными периодами времени необходимыми для завершения критических мероприятий общего проекта.

Метод PERT считается одной из разновидностей стрелочных диаграмм. В методе критического пути планирования проектов продолжительность каждого вида деятельности обычно определяется с разумной степенью достоверности. Для некоторых проектов может быть трудно оценить разумную единственную продолжительность дополнительного мероприятия в графике проекта. Метод оценки и обзора программы, или метод планирования проекта PERT, использует три продолжительности для каждого вида деятельности и фундаментальную статистику для определения вероятности завершения проекта раньше или позже, чем ожидалось. Хотя метод PERT не слишком широко используется в инженерных и строительных проектах, он предоставляет полезную информацию для оценки рисков смещения графика в проекте.

Метод PERT использует сетевую диаграмму со стрелками для отображения логической последовательности действий в проекте, в то время как СРМ использует диаграмму приоритета. На диаграмме PERT действия представлены стрелкой с кругами на каждом конце стрелки. Круги называются событиями, которые представляют собой мгновение во времени. Круг в начале действия представляет начало действия, а круг в конце стрелки — завершение действия. Основное различие между методом PERT и СРМ заключается в оценке продолжительности деятельности.

Метод PERT применим для проектов, в которых существует высокая степень неопределенности. Каждому виду деятельности присваиваются три длительности: a — оптимистическое время, b — пессимистическое время, m — наиболее вероятное время. Оптимистическое время — это максимально короткое время, в течение которого деятельность может быть завершена при

условии, что все идет хорошо. Пессимистическое время — это самое долгое время, которое может потребоваться для деятельности, если предположить, что все идет плохо. Наиболее вероятное время — это время, когда действие может быть выполнено, если его можно повторить много раз в точно таких же условиях. При этом a и b могут не быть симметричными относительно m .

3.4. Диаграмма контрольных событий

Диаграмма контрольных событий управления — это инструмент для построения графика проекта, включающий следующие основные этапы [15]:

- сбор исходной информации;
- построение сетевой диаграммы;
- выбор масштаба детализации событий;
- определение наиболее насущных событий (контрольных);
- определение порядка следования контрольных событий;
- отображение контрольных событий в графическом виде;
- контроль равномерности выполнения событий.

Результаты процесса разработки графика [15]:

1) график проекта. Может быть разработан подробно или агрегированно в виде расписания контрольных событий, которое представляется как в табличной, так и в графической форме (графики, сетевые диаграммы, гистограммы, а также диаграммы контрольных событий);

2) обязательные данные для расписания проекта. Включают события управления расписанием, запланированные операции, параметры операций и документацию всех имеющихся допущений и ограничений. Дополнительные данные включают потребности в ресурсах на периоды времени, альтернативные графики и резервы на случай непредвиденных обстоятельств;

3) базовый план расписания — специальная версия графика проекта, разработанная путем анализа сети графиков модели расписания. Принимается и утверждается командой управления проектом в качестве начального плана расписания с указанным базовым началом и базовым завершением. Базовый план-график

используется для выявления отклонений фактических сроков выполнения операций от запланированных;

4) потребности в ресурсах;

5) параметры операции;

6) календарь проекта;

7) запрошенные изменения. Во время разработки расписания могут появиться запрошенные изменения, которые обрабатываются в общем процессе управления изменениями;

8) план управления проектом (обновления). План обновляется, чтобы отразить все утвержденные изменения в способе управления расписанием проекта.

Для построения диаграммы контрольных событий необходимо на вертикальной оси отметить даты наступления контрольных событий, записанных в базовом расписании, — запланированные события. На горизонтальной оси отмечены те же даты наступления контрольных событий (рис. 30).

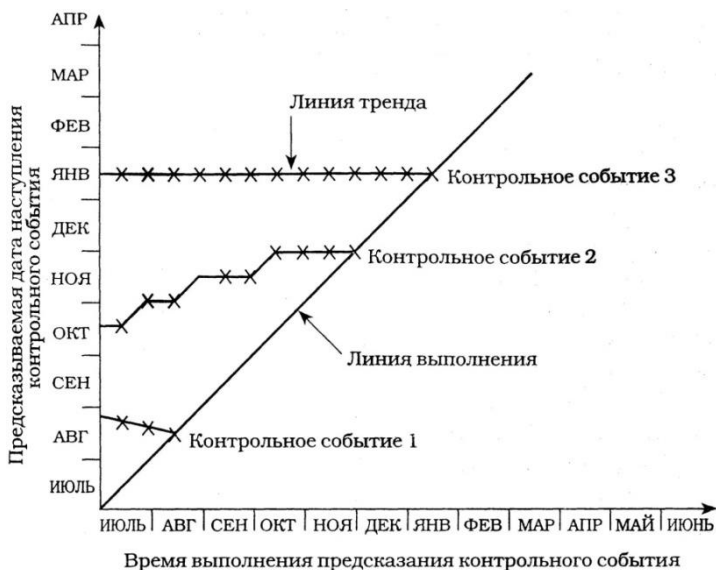


Рис. 30. Диаграмма контрольных событий¹

¹ Организация управления расписанием проекта. URL: <https://rykovodstvo.ru/exspl/16618/index.html?page=19>.

Далее рисуем запланированную линию выполнения проекта, она проходит под углом 45° к каждой из осей. Запланированные контрольные события отмечаются в строке выполнения.

Владелец первого контрольного события оценивает ход выполнения (выполнение операций, обеспечивающих достижение контрольного события) и фиксирует его на графике, а также оценивает текущие проблемы, вызывающие отклонения от базового графика, прогнозирует даты контрольного события, определяет степень влияния фактических отклонений на зависимые контрольные события.

Графически представленная информация о ходе реализации обеспечивает визуализацию внутренней динамики проекта.

3.5. Управление человеческими ресурсами ИТ-проекта

Управление человеческими ресурсами ИТ-проекта включает в себя процессы, которые организуют, управляют и руководят командой проекта. Одним из ключевых элементов управления человеческими ресурсами является четкое распределение ролей и обязанностей между членами проектной группы.

Кроме того, даже команда, работающая над задачами управления проектами, является частью проектной команды и отвечает за функции управления проектами, такие как инициирование, планирование, выполнение, мониторинг и контроль, а также закрытие, наряду с другими руководящими действиями.

В организации могут быть как функциональные менеджеры, так и менеджеры проектов. Различия заключаются в следующем.

Функциональный менеджер — это тот, кто управляет бизнес-функцией или отделом. Например, менеджер по персоналу отвечает за отдел кадров в организации. Таким образом, менеджер по персоналу будет функциональным менеджером.

Как правило, функциональный менеджер также является экспертом по предмету. Роли и обязанности функционального менеджера зависят от основной деятельности организации и общей организационной структуры.

Функциональные менеджеры обычно владеют ресурсами и обеспечивают проекты необходимыми ресурсами. Они также ведут переговоры с руководителем проекта относительно ресурсов [16].

Менеджер проекта несет ответственность за успех или неудачу проекта и качество проекта. Он не является техническим экспертом.

Менеджер проекта может не иметь полномочий в отношении ресурсов в зависимости от организационной структуры. Ему часто приходится договариваться с функциональным менеджером о ресурсах и отвечать за качество продукции.

Рассмотрим пример, чтобы понять роли, которые играют функциональные и проектные менеджеры.

Генеральный директор организации инициирует новый проект. Для этого проекта потребуются члены команды из каждого отдела компании, и тот, кто является руководителем этого проекта, должен будет попросить функционального менеджера назначить одного из членов своей команды для проекта. Ресурсы, назначенные для проекта, возвращаются в соответствующие отделы после завершения проекта.

Существует четыре процесса управления человеческими ресурсами ИТ-проекта:

- 1) планирование управления человеческими ресурсами;
- 2) приобретение проектной команды;
- 3) разработка проектной команды;
- 4) управление командой проекта.

Планирование управления человеческими ресурсами — это процесс определения и документирования подробных ролей и обязанностей каждого из членов проектной группы. Кроме того, частью этого процесса является завершение иерархии проектов, в которой определяется, кто кому подчиняется, и разработка плана управления персоналом.

Первым шагом в этом процессе является план управления проектом. В плане управления проектами и вспомогательных планах может содержаться много информации, имеющей отношение к созданию плана управления человеческими ресурсами. Например, процессы жизненного цикла проекта, сроки, способ выполнения работ и т. д.

Другим ключевым шагом в этом процессе является потребность в ресурсах деятельности. Члены команды могут быть определены на основе этих требований. В дополнение к этому также важны факторы окружающей среды предприятия, такие как текущие перспективы занятости и т. д.

Например, если экономика находится на подъеме и у членов команды есть широкие возможности для трудоустройства, система вознаграждения и признания может не заинтересовать команду к участию в проекте.

Актив организационного процесса также является вкладом в этот процесс. Примером активов процесса организации может быть шаблон или форма для запроса нового ресурса.

Роли и обязанности отражены в организационных схемах и должностных инструкциях. Существуют различные способы построения плана управления проектом в виде диаграммы; это может быть либо иерархическая диаграмма, которая показывает функцию отчетности, либо текстовая форма.

Другой метод, используемый для разработки плана людских ресурсов, — создание сетей. Это включает в себя формальное или неформальное взаимодействие внутри организации и за ее пределами, осуществляемое руководителем проекта для определения лучшей команды для ИТ-проекта в зависимости от различных межличностных факторов.

Организационная теория помогает понять, каким образом люди объединяются и функционируют как организация. Для составления плана человеческих ресурсов требуются экспертные заключения и совещания. Этот план содержит такие документы, как план управления персоналом, организационная схема проекта и матрица ролей и обязанностей. План управления персоналом — это всеобъемлющий документ, в котором описывается, когда и как члены команды будут добавлены в проект и высвобождены из него. Еще одним компонентом этого плана является система вознаграждения. Она помогает руководителю проекта мотивировать свою команду.

План управления персоналом имеет отношение к проектам функциональной или матричной организации, где члены проектной группы всегда сталкиваются с дилеммой о приоритизации проектной работы по сравнению с обычной работой отдела.

Руководитель ИТ-проекта должен убедиться, что существует организационная схема, подготовленная специально для проекта. В ней должны быть четко определены роли и отношения отчетности между членами команды.

Еще одним способом четкого разграничения обязанностей между ними является использование матрицы распределения ответственности. Одной из форм матрицы распределения ответственности является матрица (диаграмма) RACI. В диаграмме RACI: R — ответственный; A — подотчетный; C — заинтересованный или консультируемый; I — информированный.

Для каждого важного вида деятельности или результата проекта необходимо четко определить, кто несет ответственность или вносит в него свой вклад, кто в конечном итоге несет ответственность, кто просто обеспокоен успехом, а кто нуждается в информации или постоянно должен быть в курсе.

Пример диаграммы RACI показан в табл. 10. Руководитель проекта Петров отвечает за план проекта, а члены команды Сергеев, Пименов и Загородский отвечают за выполнение плана. Аналогично вы можете отметить роли каждого ресурса в других видах деятельности.

Таблица 10

Пример организационной схемы

Деятельность	Петров	Сергеев	Пименов	Загородский
План проекта	R	R	R	R
Управление конфигурацией	C	A	R	R
План тестирования	C	R	A	R
Дизайн	C	I	R	A
Бюджет команды	C	A	R	R
Взаимодействие с клиентами	A	C	R	I
Построение команды	R	R	A	C

Следующий инструмент — гистограмма ресурсов.

Гистограмма ресурсов — это метод отображения количества ресурсов необходимых для жизни проекта. На рис. 31 видно, что количество часов для работы над проектом увеличивается до максимума к периоду с 16 по 22 января, а затем имеет тенденцию сокращаться.

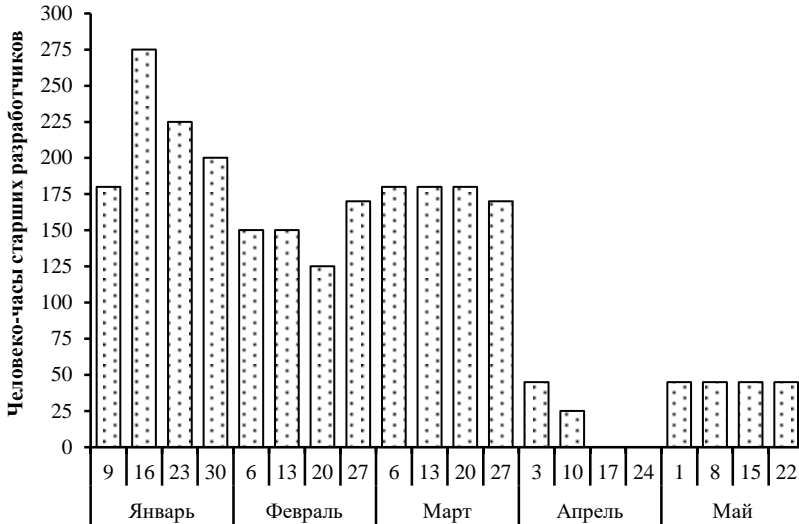


Рис. 31. Пример гистограммы ресурсов

Уровень требуемого персонала, особенно для долгосрочных проектов, может значительно варьироваться в течение определенного периода. Это имеет последствия и для других частей плана, например, стоимость будет меняться по мере изменения количества ресурсов.

Эта информация помогает руководителю проекта понять, какое время ему необходимо для использования ресурсов в рамках и за пределами ИТ-проекта.

Приобретение проектной команды. Приобретение ресурсов может не потребоваться на этапе планирования и будет осуществлено на этапе выполнения (хотя вы можете предусмотреть частичное приобретение на этапе планирования). Это действует в тех случаях, когда крупные проекты выполняются в течение многих лет и включают несколько дополнительных мероприятий.

Отправным моментом для приобретения проектной команды является план управления человеческими ресурсами. Факторы окружающей среды предприятия помогают понять, какие члены команды будут доступны для работы над проектом, сколько они будут стоить и т. д.

Ключевым методом приобретения проектной команды являются переговоры. Как руководитель проекта вы должны вести переговоры с высшим руководством компании, менеджером ресурсов или с потенциальными кандидатами, чтобы обеспечить наиболее подходящие ресурсы для проекта.

Иногда ресурсы могут быть предварительно назначены проекту. Обычно это касается «ключевых ресурсов», без которых проект не может быть запущен. Иногда проектная команда приобретает путем получения необходимых ресурсов от внешних поставщиков. Большинство крупных организаций имеют список поставщиков, которые предоставляют ресурсы на договорной основе.

В настоящее время также довольно популярна концепция виртуальной команды. Несмотря на то что проект задумывается в одной стране, большая часть команды может находиться в другой стране. Например, проект задуман в США, а большая часть команды может находиться в Германии, и обе части команды могут координировать свои действия по телефону, электронной почте, в чате и т. д.

Анализ многокритериальных решений также является важным инструментом для достижения поставленных целей. При приобретении человеческих ресурсов для проекта важно учитывать ряд факторов, таких как доступность, стоимость, набор навыков, опыт, отношение, потребности в переезде и т. д. Эти критерии должны быть рассмотрены в совокупности и сбалансированы с требованиями проекта.

Результатом являются назначения сотрудников проекта, календари ресурсов и обновления плана управления проектом.

Построение проектной команды. Следующим шагом после того, как команда проекта будет приобретена, является ее разработка (развитие). Цель состоит в том, чтобы улучшить взаимодействие между членами команды, повысить их компетентность и облегчить сплочение команды.

Конечная цель — улучшить производительность команды и, следовательно, проекта. Отличная командная работа может привести к исключительным результатам, и руководитель проекта должен сосредоточиться на развитии команды.

Входные данные для этого процесса включают назначения сотрудников по проекту, календари ресурсов и план управления человеческими ресурсами.

При построении проектной команды используются навыки межличностного общения. Это скорее искусство, чем наука, и именно здесь индивидуальные особенности менеджеров проектов имеют большое значение.

Если руководитель проекта пользуется доверием и уважением членов команды, развитие команды становится легким.

Члены команды связываются с руководителем проекта не только для получения инструкций по конкретному проекту, но и для своего карьерного роста. Социальные мероприятия, такие как празднование дня рождения члена команды, посещение вечеринок вне офиса и т. д., также помогают в построении команды.

Важным инструментом для развития проектной команды является обучение, поскольку оно помогает повысить знания и компетенции ее членов.

Иногда размещение членов команды в одном месте также помогает в развитии сплоченности. Это особенно важно, если члены команды работают в другом географическом месте. Личное знание человека действительно помогает уменьшить конфликт.

Руководители проектов также могут поощрять членов команды, давая им признание и награды.

Инструменты оценки персонала дают руководителю проекта представление о сильных сторонах и возможностях для улучшения членов команды, помогают понять индивидуальные устремления участников, чтобы он мог лучше согласовать их с целями проекта.

Наконец, руководитель проекта должен установить некоторые основные правила, чтобы гарантировать, что команда выполняет свою работу без каких-либо проблем.

Результатом этого процесса является документ, оценивающий эффективность работы команды. По мере развития информации о команде могут обновляться факторы окружающей среды предприятия.

Управление командой проекта отличается от ее разработки. Это связано с измерением производительности членов команды, предоставлением обратной связи, решением проблем и управлением изменениями для оптимизации производительности проекта.

Чтобы измерить производительность членов команды, менеджер должен в деталях знать назначенную им работу.

План управления человеческими ресурсами служит руководством для управления проектными группами. Некоторые организации имеют шаблон и процедуры для проведения такой оценки.

Отчеты о выполнении работы и журналы проблем будут входными данными, поскольку в этом контексте необходимо выполнять действия по управлению командой. Наблюдение и обсуждение — это метод, при котором руководитель проекта контролирует и обговаривает задачи, выполняемые командой.

Это подразумевает не официальные отчеты о состоянии проекта, а неформальное взаимодействие с командой перед документированием. Знание о проблемах на ранней стадии помогает раньше решить их. Формальная оценка полезна для эффективности работы члена команды в проекте.

В большинстве организаций существует устоявшаяся практика оценки работы членов команды. Это может повлиять на их будущие назначения, зарплату, продвижение по службе и т. д. Поскольку полномочия по оценке лежат на руководителе проекта, он может использовать это как эффективный инструмент для управления командой.

Когда несколько человек работают над одним проектом, вполне естественно возникают конфликты. Ответственность за разрешение конфликтов также лежит на руководителе проекта.

3.6. ИТ-команды.

Требования к составу ИТ-команды

Каждая ИТ-команда отличается от других, это зависит от культуры и потребностей компании, опыта и навыков членов команды, а также типов систем, над которыми они работают.

Некоторые ИТ-команды являются технологическими универсалами, работающими с широким спектром систем и сервисов. Другие ИТ-команды специализируются на конкретных технологиях (таких как сетевые или веб-сервисы) или на определенном типе систем (поддержка продаж, производство, логистика и т. д.). Некоторые ИТ-команды фокусируются только на технологиях, в то время как другие могут включать специалистов по данным или аналитиков с большим опытом работы с бизнес-процессами. ИТ-команды успешны, когда способны преодолеть любую проблему.

Хотя каждая ИТ-команда уникальна, существует несколько общих принципов, по которым они формируются. Некоторые ИТ-команды являются постоянными, в то время как другие существуют в течение периода времени для достижения определенного результата, а затем распускаются.

Основными видами ИТ-команд являются [17]:

1) оперативные группы. Эти команды сосредоточены на эксплуатации технологической инфраструктуры (такой как сети, центры обработки данных и веб-службы), их мониторинге и обеспечении доступности и нормальной работы служб и систем для поддержки бизнес-операций. Эти команды часто называют командами управления ИТ-услугами (ITSM), и их главная цель — поддерживать ИТ-экосистему в рабочем состоянии. Оперативные группы являются постоянными, и часто их поддерживает сложная инфраструктура мониторинга и командного центра;

2) проектные группы. Эти команды собираются для решения конкретной проблемы, внедрения системы или внесения изменений. Проекты являются временными, поэтому, как только проект завершен, команда либо распускается (с членами команды, назначенными в другое место), либо целиком переходит в другой проект. Команды ИТ-проектов, как правило, сосредоточены на одном выпуске или группе выпусков, но очень редко «владеют» системой после ее запуска;

3) группы поддержки. Эти команды похожи как на оперативные, так и на проектные. Группы ИТ-поддержки являются постоянными, как и оперативные группы, выполняют функции управления ИТ-услугами, чтобы поддерживать бесперебойную работу бизнес-операций. Однако эти команды похожи и на про-

ектные команды, поскольку они также решают конкретные проблемы. Команда ИТ-поддержки часто рассматривается как оперативная группа, назначенная для выполнения множества мини-проектов каждый день;

4) команды процессов. Эти команды (также иногда известные как группы поддержки ИТ-бизнеса) являются постоянными, и им поручено владеть, управлять и совершенствовать ИТ-системы конкретного бизнес-процесса. Эти команды часто укомплектованы аналитиками, руководителями проектов и экспертами по данным и сотрудничают с проектными группами для достижения конкретных результатов.

Роли и обязанности членов ИТ-команды зависят от ее заданий. Некоторые ИТ-команды назначаются для достижения конкретных результатов проекта, в то время как другие отвечают за обеспечение долгосрочной производительности и прибыльности бизнес-процессов. Однако все они имеют некоторое сходство. Основная цель любой ИТ-команды — помочь достичь бизнес-результатов, поэтому сотрудникам бизнеса не нужно становиться экспертами в области технологий. ИТ-команды управляют технологией, а бизнес-персонал может сосредоточиться на деловой деятельности.

К ИТ-командам в компании предъявляются определенные требования, в числе которых [17]:

1) технологический опыт. ИТ-команды отличаются от команд бизнес-процессов тем, что они укомплектованы технологическими экспертами и людьми, которые являются специалистами в понимании того, как технологии могут быть применены для повышения эффективности бизнеса;

2) управление данными. Многие ИТ-команды являются распорядителями данных для своих организаций. Поскольку бизнес-данные все чаще хранятся и управляются ИТ-системами, ИТ-команды лучше всего понимают данные компании, их связь с другими данными и их использование;

3) автоматизация процессов. Одним из основных направлений деятельности ИТ-команд является использование технологий для автоматизации бизнес-процессов. Автоматизация процессов включает в себя понимание бизнес-процессов, а также

технологий и адаптацию системных рабочих процессов для интеграции с деятельностью бизнес-персонала;

4) системная интеграция. Большинство бизнес-процессов полагаются на более чем одну ИТ-систему, чтобы быть успешными. ИТ-команды, как правило, отвечают за разработку, эксплуатацию и поддержание интеграции между ИТ-системами компании, а также со сторонними системами и сервисами;

5) управление портфелем. ИТ-команды управляют портфелем технологических инвестиций компании. Они не только отвечают за правильную работу систем для поддержания бизнеса, но и определяют общую стоимость владения ИТ-системами и принимают решения о технологическом жизненном цикле;

6) управление и обслуживание ИТ-систем. Компании вкладывают огромное количество времени и денег во внедрение ИТ-систем. Чтобы обеспечить максимальную отдачу от этих инвестиций, ИТ-команды должны постоянно поддерживать системы в актуальном состоянии и модернизировать их в соответствии с меняющимися потребностями бизнеса;

7) информационная и системная безопасность. ИТ-системы являются важнейшей частью деятельности компании. ИТ-команды назначаются не только для создания и запуска этих систем, но и для обеспечения их безопасности. Оценка, управление и смягчение уязвимостей безопасности и связанных с технологиями рисков являются важными частями работы ИТ-команды;

8) управление поставщиками. Большинство ИТ-систем включают в себя сторонние компоненты. Инфраструктура телефонии, облачные сервисы и стороннее программное обеспечение распространены в большинстве бизнес-сред. ИТ-команда несет ответственность не только за обеспечение гарантий обслуживания этих систем, но и за отношения с внешними поставщиками технологий и группами поддержки;

9) технологическая поддержка. Пожалуй, самая заметная роль ИТ-технологий — восстановление технологических систем, когда они выходят из строя. ИТ-команды часто предоставляют службу поддержки, полевую поддержку и услуги по реагированию на инциденты для устранения технологических сбоев.

По мере расширения технологической среды компании увеличивается и размер ее ИТ-организации. Поэтому важно, что-

бы каждая ИТ-команда (и подгруппа) имела четкий объем ответственности и полномочий для принятия решений о ресурсах, которые они развертывают, и технологиях, которые они внедряют. Вот несколько примеров типичных организационных структур ИТ-команды.

1. Команды, ориентированные на технологии. Это была одной из первых конструкций ИТ-команды, которая являлась организационной структурой по умолчанию на протяжении большей части 1990-х годов. По сути, это иерархическая организационная структура, классифицируемая по технологическим областям, например, сетевая команда, команда центра обработки данных, команда базы данных и веб-команда.

2. Команды, ориентированные на бизнес. Эта организационная структура была разработана для поддержки крупных платформенных систем, таких как ERP, HRM и CRM-системы, которые стали популярными в начале 2000-х годов. Здесь ИТ-команды согласованы с функциями бизнес-процессов и служат в качестве полуавтономной ИТ-организации для бизнес-подразделения.

3. Функциональные команды ITSM. Сложные ИТ-проекты часто требуют ресурсов из разных дисциплин, а также различных навыков и опыта. Поскольку проектные группы являются временными, компании часто используют подходы матричного управления, при которых ресурсы проекта остаются согласованными с основной организационной структурой (бизнес- или технологически согласованными) и назначаются проектным командам, которые могут координироваться с другими частями ИТ-организации.

4. Команды матричных проектов.

5. Гибридные бизнес/ИТ-команды. Цифровая трансформация бизнеса и расширение интеграции технологических услуг и бизнес-процессов привели к тому, что многие ИТ-команды объединились с командами бизнес-процессов. Гибридные команды часто включают как экспертов по бизнес-процессам с небольшим опытом работы в области технологий, так и технологов и экспертов по данным.

6. Виртуальные команды. Технология совместной работы и широкая доступность широкополосного доступа в Интернет

снизили потребность в локализованном размещении ИТ-групп для обеспечения эффективности. Виртуальные команды широко распространены в ИТ-индустрии, что позволяет компаниям привлекать как специализированных экспертов, так и недорогие технологические ресурсы со всего мира. Виртуальные команды особенно эффективны для краткосрочных проектов, где члены команды также несут ответственность за продолжение своей повседневной работы.

7. Офисы управления проектами/программами (PMOs). Это организационные структуры, предназначенные для обеспечения управления, координации и централизованного принятия решений в распределенных ИТ-организациях. РМО часто управляют такими видами деятельности, как составление бюджета, планирование выпуска и управление рисками, где требуется высокая степень межфункциональной координации.

8. Центры передового опыта. Это организационная структура, используемая в сочетании со структурами других ИТ-команд (такими как команды, ориентированные на бизнес, или команды матричных проектов) для сбора, разработки и обмена передовым опытом в рамках всей организации. Данные центры продвигают контролируемые инновации без риска того, что автономные ИТ-команды предложат непоследовательные способы работы.

9. Постоянные проектные группы. ИТ-проекты являются временными и рассчитаны на полное завершение, а затем расформирование ИТ-команды. Это может вызвать проблемы с планированием карьеры и управлением людскими ресурсами, если нет структурированного плана перераспределения ресурсов. Общий подход заключается в организации ИТ-ресурсов в постоянные команды, которые перемещаются как единое целое от проекта к проекту. Когда один проект завершен, вся команда переходит к следующему заданию.

10. Пулы ресурсов. Основываясь на бизнес-моделях консалтинга, некоторые компании организуют ИТ-ресурсы в пулы на основе рабочих дисциплин. Когда возникают проекты и другие ИТ-потребности, ресурсы извлекаются из рабочей группы и назначаются для участия. Когда работа завершена, ресурс

возвращается в неназначенное состояние и перераспределяется в другое место.

11. Географически ориентированные команды. Глобальные компании часто структурируют часть своей ИТ-организации как локализованные команды, ориентированные на конкретные географические регионы. Это особенно распространено в ситуациях, когда языковые, культурные факторы или нормативные документы требуют локализованного присутствия. Географически ориентированные команды часто используются в сочетании с централизованными структурами ИТ-команд для содействия глобальным стандартам и минимизации затрат на ИТ.

Требования к работе ИТ-команды разнообразны. Хотя команда может восприниматься как группа технологов, многие специальности людей, работающих в ней, в настоящее время не являются техническими. Они включают в себя сочетание деловых и операционных знаний, позволяющих ИТ-команде достичь наилучших результатов. Как правило, обязанности внутри команды распределяются следующим образом:

1) руководители проектов координируют деятельность членов команды;

2) бизнес-аналитики и ИТ-аналитики выявляют возможности повышения производительности и затрат и анализируют потребности бизнеса в ИТ-решениях;

3) проектировщики бизнес-процессов проектируют и совершенствуют как бизнес-процессы, так и системные рабочие процессы, чтобы уменьшить трения и повысить эффективность процессов;

4) ИТ-инженеры/разработчики создают, поддерживают, исправляют ИТ-оборудование, программное обеспечение и услуги;

5) тестировщики проводят тесты для проверки работоспособности ИТ-систем и выявления дефектов, требующих устранения;

6) дизайнеры пользовательских интерфейсов разрабатывают графические пользовательские интерфейсы, позволяющие людям эффективно, доступно и эргономично взаимодействовать с ИТ-системами;

7) специалисты по обработке данных и аналитики данных разрабатывают, внедряют и управляют компонентами данных

ИТ-системы, уделяя особое внимание интеграции данных и удобству использования данных для принятия решений;

8) архитекторы решений проектируют и планируют ИТ-системы с несколькими компонентами, включая системные интерфейсы, учитывая соображения масштабируемости и совместимости;

9) эксперты по информационной безопасности и рискам. Риски являются ключевой проблемой для большинства компаний, что заставляет их включать экспертов по безопасности и рискам в ИТ-команды;

10) специалисты ITSM — менеджеры по инцидентам, менеджеры по проблемам, менеджеры по изменениям и другие сотрудники ITSM — отвечают за обслуживание ИТ-оборудования, программного обеспечения и предоставление услуг;

11) управление поставщиками регулирует и облегчает взаимодействие с поставщиками и внешними сторонами, непосредственно участвующими или поддерживающими усилия ИТ-команды.

3.7. Заинтересованные стороны (участники) проекта

Заинтересованная сторона (участник проекта) — это физическое лицо, группа или организация, которые могут влиять, быть задействованы или воспринимать себя задействованными в решениях, деятельности или результатах проекта. Заинтересованные стороны проекта могут быть внутренними или внешними по отношению к проекту, они могут быть активно вовлечены, пассивно вовлечены или не знать о проекте. Могут оказывать положительное или отрицательное влияние на проект, а также быть положительно или отрицательно затронуты проектом. Примеры заинтересованных сторон включают [5]:

1) внутренние заинтересованные стороны — спонсор, менеджер ресурсов, отдел управления проектами (ОУП), руководящий комитет, руководитель проекта, руководители иных проектов, члены команды, пр.;

2) внешние заинтересованные стороны — клиенты, конечные пользователи, поставщики, акционеры, регулирующие органы, конкуренты, пр.

Участие заинтересованных сторон может варьироваться от случайного участия в опросах и фокус-группах до полного спонсорства проекта, которое включает предоставление финансовой, политической или других видов поддержки. Тип и уровень участия в проекте могут меняться в течение жизненного цикла проекта. Поэтому успешное выявление, анализ и вовлечение заинтересованных сторон, а также эффективное управление их ожиданиями и участием в проекте на протяжении всего жизненного цикла проекта имеют решающее значение для его успеха.

Руководитель проекта — это лицо, назначенное исполняющей организацией для руководства командой, ответственной за достижение целей проекта. Отношения отчетности руководителя проекта основаны на организационной структуре и управлении проектом.

В дополнение к конкретным техническим навыкам и общим навыкам управления руководители проектов должны обладать следующими характеристиками:

- знания об управлении проектами, бизнес-среде, технических аспектах и другой информации необходимой для эффективного управления проектом;
- навыки необходимые для эффективного руководства проектной командой, координации работы, сотрудничества с заинтересованными сторонами, решения проблем и принятия решений;
- способности разрабатывать и управлять расписаниями, бюджетами, ресурсами, рисками, планами, презентациями и отчетами;
- прочие свойства, такие как харизма, этика и лидерство.

Руководители проектов выполняют работу через проектную группу и другие заинтересованные стороны. Они полагаются на такие важные навыки межличностного общения, как лидерство, командообразование, мотивация, общение, влияние, принятие решений, политическая и культурная осведомленность, ведение переговоров, содействие, управление конфликтами, коучинг.

Руководитель проекта добивается успеха, когда цели проекта достигнуты. Еще одним аспектом успеха является удовлетворенность заинтересованных сторон, соответственно руководитель проекта должен учитывать их потребности, проблемы и ожидания. Для достижения успеха подход к проекту, жизненный цикл и процессы управления должны соответствовать требованиям проекта и продукта.

Области знаний по управлению проектами — это области специализации, которые обычно используются в данной деятельности. В большинстве проектов они востребованы большую часть времени и охватывают основные функции в рамках работы над ИТ-проектом:

1) управление интеграцией проекта — действия, направленные на определение, объединение, унификацию и координацию различных процессов и мероприятий по управлению проектами;

2) управление масштабом проекта — процессы, обеспечивающие включение в проект всей работы необходимой для успешного завершения (но не более);

3) управление расписанием проекта — процессы, обеспечивающие своевременное завершение проекта;

4) управление стоимостью проекта — процессы, связанные с планированием, оценкой, составлением бюджета, финансированием, управлением и контролем затрат, чтобы проект мог быть завершен в рамках утвержденного бюджета;

5) управление качеством проекта — процессы, реализующие политику организации в области качества в отношении планирования, управления и контроля требований к качеству проекта и продукции в целях удовлетворения ожиданий заинтересованных сторон;

6) управление ресурсами проекта — процессы выявления, приобретения и управления ресурсами, требуемыми для успешного завершения проекта;

7) управление коммуникациями проекта — процессы необходимые для обеспечения своевременного и надлежащего планирования, сбора, создания, распространения, хранения, поиска, управления, контроля, мониторинга и окончательного распоряжения информацией о проекте;

8) управление проектными рисками — идентификация и анализ рисков, планирование и реализация ответных мер и мониторинг рисков по проекту;

9) управление закупками проекта — процессы необходимые для приобретения продуктов, услуг за пределами проектной группы;

10) управление заинтересованными сторонами проекта — выявление людей, групп или организаций, которые могут повлиять или подвергнуться воздействию проекта, для анализа их ожиданий и влияния на проект, а также разработка соответствующей стратегии управления для эффективного вовлечения заинтересованных сторон в принятие решений и выполнение проекта.

3.8. Управление командой ИТ-проекта

Самым первым шагом в управлении командой ИТ-проекта является собственно формирование и развитие команды.

Развитие команды — это процесс улучшения компетенций, взаимодействия членов команды и общей командной среды для повышения эффективности проекта. Этот процесс приводит к улучшению командной работы, повышению навыков и компетенций межличностного общения, мотивации сотрудников, снижению потерь и повышению общей эффективности проекта (PMBoK).

Этот процесс выполняется на протяжении всего проекта.

Менеджерам ИТ-проектов требуются навыки создания, поддержания, мотивации, руководства и вдохновения проектных команд для обеспечения высокой производительности и достижения целей проекта. Командная работа является важнейшим фактором успеха проекта, и создание эффективных проектных команд — одна из основных обязанностей руководителя проекта. Менеджеры ИТ-проектов должны создавать среду, которая облегчает командную работу и постоянно мотивирует команду, предотвращая проблемы и предоставляя возможности, обеспечивая своевременную обратную связь и поддержку по мере необходимости, а также признавая и вознаграждая хорошую работу.

Высокая производительность команды может быть достигнута за счет использования таких моделей поведения, как:

- открытая и эффективная коммуникация;
- предоставление возможностей для создания команды;
- развитие доверия между членами команды;
- конструктивное управление конфликтами;
- поощрение совместного решения проблем;
- поощрение совместного принятия решений.

Руководители проектов работают в глобальной среде, где проекты характеризуются культурным разнообразием.

Члены команды часто имеют разный опыт работы в отрасли, общаются на нескольких языках, а иногда работают на «языке команды» или в рамках культурной нормы, которая может существенно отличаться от их родной. Команда управления проектом должна научиться извлекать выгоду из культурных различий, сосредоточиться на развитии и поддержании проектной команды на протяжении всего жизненного цикла проекта и поощрять совместную взаимозависимую работу в атмосфере взаимного доверия. Развитие проектной команды улучшает навыки персонала, технические компетенции, общую командную среду и производительность проекта. Это требует четкой, своевременной, эффективной и действенной коммуникации между членами команды на протяжении всего срока реализации проекта.

Цели развития команды ИТ-проекта:

1) совершенствование знаний и навыков членов команды для повышения их способности добиваться результатов при одновременном снижении затрат, сокращении сроков и повышении качества;

2) улучшение чувства доверия и согласия между членами команды для поднятия морального духа, снижения конфликтности и повышения эффективности командной работы;

3) создание командной культуры в целях:

- повышения индивидуальной и командной производительности, командного духа и сотрудничества;
- обеспечения совместного обучения и наставничества между членами команды, обмена знаниями и опытом;

4) предоставление команде возможности участвовать в принятии решений и брать на себя ответственность за принятые ре-

шения для повышения производительности и достижения наилучших результатов.

Одной из моделей, используемых для описания командного развития, является лестница Такмана. Она включает в себя пять этапов развития, которые могут пройти команды. Хотя обычно эти этапы идут по порядку, нередко команда застревает на определенной стадии или регрессирует на более раннюю стадию. Проекты с членами команды, которые работали вместе в прошлом, могут пропустить определенный этап. Этапы следующие:

1) формирование. Члены команды встречаются и узнают о проекте и своих официальных ролях и обязанностях. На этом этапе они, как правило, независимы и не очень открыты;

2) притирка. Команда приступает к работе над проектом, техническим решениям в соответствии с подходом к управлению проектом. Если члены команды не сотрудничают или не открыты для различных идей и перспектив, окружающая среда может стать контрпродуктивной;

3) нормализация. Члены команды начинают работать вместе и приспособливают свои рабочие привычки и поведение к интересам команды; учатся доверять друг другу;

4) исполнение. Команды, достигшие стадии исполнения, функционируют как хорошо организованная единица. Они взаимозависимы и решают проблемы плавно и эффективно;

5) прерывание. Команда завершает работу и переходит к новому проекту. Это обычно происходит, когда персонал освобождается от проекта по мере достижения результатов или в рамках процесса закрытия проекта или фазы.

Для развития команды ИТ-проекта необходимы проектные документы:

1) журнал регистрации ошибок при управлении командой. Уроки, извлеченные ранее в рамках проекта в отношении развития команды, могут быть применены на более поздних этапах проекта для повышения эффективности работы команды;

2) расписание проекта. Определяет, как и когда проводить обучение проектной команды и развивать компетенции необходимые на различных этапах. Также определяет потребность в команде, стратегии развития, основанные на изменениях, если таковые имеются, во время выполнения проекта;

3) назначение команды ИТ-проектной группы. Определяет роли и обязанности команды и ее членов;

4) календари ресурсов. Определяют время, когда члены проектной группы могут участвовать в мероприятиях по разработке команды. Это также помогает проиллюстрировать доступность команды в течение всего проекта;

5) устав команды. В уставе содержатся руководящие принципы работы команды. Оценка команды и руководящие принципы обеспечивают структуру, описывающую, как команда будет работать.

Для развития команды, как правило, используются следующие инструменты:

1) колокация. Включает в себя размещение большинства или всех наиболее активных членов проектной группы в одном физическом месте, чтобы повысить их способность работать в команде. Колокация может носить разовый характер, например, в стратегически важные моменты во время проекта, или может осуществляться перманентно в течение всего проекта;

2) виртуальная команда. Может дать такие преимущества, как использование более квалифицированных ресурсов, снижение затрат, сокращение расходов на поездки и переезд, а также близость членов команды к поставщикам, клиентам или другим ключевым заинтересованным сторонам;

3) создание сетевой командной среды, в которой могут храниться файлы, обсуждаться проблемы и вестись календарь команды;

4) коммуникационные технологии. Важны для решения проблем развития команд в коллективные и виртуальные. Это помогает создать гармоничную среду для коллективной команды и лучше понять виртуальную команду, особенно тех, кто работает в разных часовых поясах. Примерами коммуникационных технологий здесь являются:

– общий портал — совместное хранилище данных и информации (например, веб-сайт, облачное программное обеспечение для совместной работы или сток информации и т. д.); эффективен, как правило, для виртуальных проектных групп;

– видеоконференции — важный метод эффективного общения с виртуальными командами;

– аудиоконференции — еще один метод для создания взаимопонимания и доверия в виртуальных командах;

– электронная почта/чат. Регулярное общение с помощью электронной почты и чата также является эффективным методом.

Межличностные и командные навыки, которые могут быть использованы для этого процесса, включают:

1) управление конфликтами. Руководитель проекта должен своевременно и конструктивно разрешать конфликты, чтобы создать высокоэффективную команду;

2) влияние. Навык влияния заключается в сборе актуальной информации для решения важных вопросов и достижения соглашений при сохранении взаимного доверия;

3) мотивация — побуждение кого-то к действию. Команды мотивируют тем, что дают им возможность участвовать в принятии решений и поощряют их к самостоятельной работе;

4) переговоры. Переговоры между членами команды используются для достижения консенсуса по потребностям проекта; могут укрепить доверие и гармонию между членами команды;

5) сплочение команды. Подразумевает проведение мероприятий, которые улучшают социальные отношения в команде и создают благоприятную рабочую среду.

Мероприятия по созданию команды могут варьироваться от пятиминутного пункта повестки дня на совещании по обзору статуса до выездного, профессионально организованного мероприятия, направленного на улучшение межличностных отношений. Цель мероприятий по созданию команды состоит в том, чтобы помочь отдельным членам команды эффективно работать вместе. Стратегии создания команды особенно ценны, когда члены команды работают удаленно, без личного контакта. Неформальное общение и мероприятия могут помочь в построении доверия и установлении хороших рабочих отношений. Хотя создание команды имеет важное значение на начальных этапах проекта, это должен быть непрерывный процесс. Изменения в среде проекта неизбежны, и для эффективного управления ими могут быть приложены постоянные или возобновленные усилия по созданию команды. Руководитель проекта должен постоянно контролировать функциональность и производительность коман-

ды, чтобы определить, необходимы ли какие-либо действия для предотвращения или исправления различных проблем.

Часть процесса развития команды включает в себя признание и поощрение желательного поведения. Вознаграждение будет эффективным только в том случае, если оно удовлетворяет потребность, которая ценится этим человеком. Решения о вознаграждении принимаются формально или неформально в процессе управления командой проекта. Как правило, деньги рассматриваются как материальный аспект любой системы вознаграждения, но нематериальные вознаграждения могут быть столь же или даже более эффективными. Большинство членов проектной команды мотивированы возможностью расти, добиваться успеха, быть оцененными и применять свои профессиональные навыки для решения новых задач.

Обучение включает в себя все мероприятия, направленные на повышение компетентности членов проектной группы. Обучение может быть формальным или неформальным. Примеры методов обучения включают аудиторное, онлайнное, компьютерное обучение на рабочем месте у другого члена проектной команды, наставничество и коучинг. Если члены проектной команды не обладают необходимыми управленческими или техническими навыками, такие навыки могут быть развиты в рамках проектной работы.

Плановое обучение проходит в соответствии с планом управления ресурсами. Внеплановое обучение проводится в результате наблюдения, беседы и оценки эффективности проекта, проведенной во время управления проектной командой. Расходы на обучение могут быть включены в бюджет проекта или подержаны исполняющей организацией, если дополнительные навыки могут пригодиться в будущих проектах. Обучение могут проводить собственные или внешние тренеры.

Индивидуальные и командные инструменты оценки важны и дают руководителю проекта и проектной команде представление о сильных и слабых сторонах. Эти инструменты помогают руководителям проектов оценивать предпочтения, стремления членов команды, понять то, как они обрабатывают и организуют информацию, как они принимают решения и как взаимодействуют с людьми. Доступны различные инструменты, такие как

опросы отношения, конкретные оценки, структурированные интервью, тесты способностей и фокус-группы. Эти инструменты могут обеспечить лучшее понимание, доверие, приверженность и коммуникацию между членами команды и способствовать продуктивности команды на протяжении всего проекта.

Встречи используются для обсуждения и рассмотрения соответствующих тем для развития команды. Участники — руководитель проекта и команда проекта. Типы совещаний: по ориентации проекта; по созданию команды; по развитию команды.

Проектные документы, которые могут рассматриваться в качестве исходных данных для этого процесса:

1) журнал проблем. Проблемы возникают в процессе управления проектной командой. Журнал проблем может использоваться для документирования и мониторинга того, кто отвечает за решение конкретных вопросов к намеченной дате;

2) регистрация полученного опыта. Уроки и опыт, извлеченные ранее в рамках проекта, могут быть применены на более поздних этапах проекта для повышения эффективности и результативности управления командой;

3) задания проектной группы. Назначения проектной группы определяют роли и обязанности членов команды;

4) устав команды. Устав определяет, как команда будет принимать решения, проводить встречи и разрешать конфликты.

Отчет о выполнении работы может быть представлен в физическом или электронном виде. Содержащиеся в нем сведения предназначены для принятия решений, действий или информирования. Отчеты о производительности, которые также могут помочь в управлении проектной группой, включают результаты контроля расписания, контроля затрат, контроля качества и проверки области применения.

Информация из отчетов о результатах работы и связанных с ними прогнозов помогает определить будущие потребности команды в ресурсах, признание и награды, а также обновления плана управления ресурсами.

Команда управления проектом все время проводит формальную или неформальную оценку эффективности работы команды проекта. Это позволяет предпринимать действия для ре-

шения проблем, изменения коммуникации, улучшения взаимодействия в команде и устранения конфликтов.

Активы организационного процесса, которые могут влиять на управление командой, включают:

- благодарственные сертификаты;
- корпоративную одежду;
- другие организационные привилегии.

Межличностные и командные навыки, которые могут быть использованы для управления командой, обязательно включают управление конфликтами. Конфликт неизбежен в проектной среде. К источникам конфликтов относятся ресурсы, приоритеты планирования, личные стили работы и т. д.

Основные правила команды, групповые нормы и надежные методы управления проектами, такие как планирование коммуникаций и определение ролей, уменьшают количество конфликтов.

Успешное управление конфликтами приводит к повышению производительности и позитивным рабочим отношениям. При правильном управлении различия во мнениях могут привести к повышению креативности и лучшему принятию решений. Если различия становятся негативным фактором, члены проектной группы несут ответственность за их разрешение. Если конфликт обострится, руководитель проекта должен способствовать удовлетворительному разрешению проблемы. Конфликт следует решать на ранней стадии и, как правило, в частном порядке. Если разрушительный конфликт продолжается, могут быть использованы официальные процедуры, включая дисциплинарные меры.

Успех менеджеров проектов в управлении своими проектными командами часто зависит от их способности разрешать конфликты. При этом могут использоваться разные методы. Факторы, влияющие на выбор методов разрешения конфликтов:

- важность и интенсивность конфликта;
- время для разрешения конфликта;
- относительная власть людей, вовлеченных в конфликт;
- важность поддержания хороших отношений;
- мотивация для разрешения конфликта на долгосрочной или краткосрочной основе.

Существует пять общих методов разрешения конфликтов:

1) предотвращение. Отступление от реальной или потенциальной конфликтной ситуации; откладывание вопроса, чтобы он был лучше подготовлен или решен другими;

2) приспособление. Подчеркивание областей согласия, а не областей разногласий; признание своих позиций по отношению к потребностям других людей для поддержания гармонии в отношениях;

3) компромисс (примирение). Поиск решений, которые принесут определенную степень удовлетворения всем сторонам, чтобы временно или частично разрешить конфликт. Такой подход иногда приводит к ситуации проигрыша;

4) навязывание. Утверждение своей точки зрения за счет других; предложение только беспроеигрышных решений, обычно навязываемых через позицию власти для разрешения чрезвычайной ситуации. Такой подход часто приводит к ситуации как выигрыша, так и проигрыша;

5) совместное решение проблем. Включение нескольких точек зрения и идей с разных точек зрения; требует сотрудничества и открытого диалога, который обычно приводит к консенсусу и приверженности. Такой подход может привести к беспроеигрышной ситуации.

Принятие решений в контексте управления командой предполагает способность вести переговоры и влиять на организацию и команду управления проектами, а не набор инструментов. Вот некоторые руководящие принципы для принятия решений:

- сосредоточенность на целях, которые должны быть достигнуты;
- контроль процесса принятия решений;
- изучение факторов окружающей среды;
- анализ имеющейся информации;
- стимулирование творческого потенциала команды;
- учет рисков.

Эмоциональный интеллект — это способность идентифицировать, оценивать и управлять личными эмоциями, как своими, так и других людей, а также коллективными эмоциями групп людей. Руководитель может использовать эмоциональный интел-

лект для снижения напряженности и расширения сотрудничества путем выявления, оценки и контроля настроения членов проектной команды, предвидеть их действия, признавать их проблемы и следить за этими проблемами.

Поскольку руководители проектов часто имеют мало или вообще не имеют прямых полномочий для влияния на членов команд в матричной среде, их способность своевременно влиять на заинтересованные стороны имеет решающее значение для успеха проекта. Ключевые навыки влияния:

- способность быть убедительным;
- умение четко сформулировать пункты и позиции;
- высокий уровень активных и эффективных навыков слушания;
- осведомленность о различных перспективах в любой ситуации и их учет;
- сбор соответствующей информации для решения проблем и достижения соглашений при сохранении взаимного доверия.

Успешные проекты требуют руководителей с сильными лидерскими качествами. Лидерство — это способность руководить командой и вдохновлять ее хорошо выполнять свою работу. Оно охватывает широкий спектр навыков, умений и действий.

Лидерство важно на всех этапах жизненного цикла проекта. Существует множество теорий и стилей лидерства, которые следует использовать по мере необходимости для каждой ситуации или команды. Особенно важно донести это видение и вдохновить проектную команду на достижение высокой производительности.

Выводы по главе 3

Основными методами планирования расписания ИТ-проектов являются:

- 1) математический анализ, включая метод критического пути (МКП) и методику оценки и анализа программы (PERT);
- 2) сжатие продолжительности, включая быстрое отслеживание и сбой;

- 3) моделирование;
- 4) эвристика выравнивания ресурсов;
- 5) список задач;
- 6) диаграмма Ганта;
- 7) электронный календарь.

В диаграмме Ганта разделение пространства представляется как количество времени и объем работы, которую нужно выполнить за это время. Равные деления пространства на одной горизонтальной линии представляют одновременно:

- 1) равные отрезки времени;
- 2) различные объемы запланированных работ;
- 3) различные объемы выполненных работ.

Основой метода критического пути является построение модели проекта, которая включает в себя:

- список всех задач необходимых для завершения проекта;
- зависимости между задачами;
- оценку времени (продолжительности), которое потребуется для завершения каждого действия.

Диаграмма контрольных событий управления включает основные этапы:

- сбор исходной информации;
- построение сетевой диаграммы;
- выбор масштаба детализации событий;
- определение наиболее насущных событий (контрольных);
- определение порядка следования контрольных событий;
- отображение контрольных событий в графическом виде;
- контроль равномерности выполнения событий.

Управление человеческими ресурсами ИТ-проекта включает четыре процесса:

- 1) планирование управления человеческими ресурсами;
- 2) приобретение проектной команды;
- 3) построение проектной команды;
- 4) управление командой проекта.

Основные виды ИТ-команд:

- 1) оперативные группы;
- 2) проектные группы;
- 3) группы поддержки;
- 4) команды процессов.

Заинтересованные стороны проекта могут быть внутренними или внешними. Внутренние заинтересованные стороны — это спонсор, менеджер ресурсов, отдел управления проектами (ОУП), руководящий комитет, руководитель проекта, руководители иных проектов, члены команды, пр.

Внешние заинтересованные стороны — клиенты, конечные пользователи, поставщики, акционеры, регулирующие органы, конкуренты, пр.

Командная работа является важнейшим фактором успеха проекта. Поэтому руководители ИТ-проектов должны создавать среду, которая облегчает командную работу и постоянно мотивирует команду, предотвращая проблемы и предоставляя возможности, обеспечивая своевременную обратную связь и поддержку.

Вопросы для самоконтроля

1. Подробно раскройте суть метода критического пути.
2. Опишите методику оценки и анализа программы (PERT).
3. Что представляет собой инструмент «Сжатие продолжительности»? Какие две разновидности инструмента существует?
4. Как используется моделирование в процессе планирования расписания ИТ-проектов?
5. Что такое эвристика выравнивания ресурсов? Какова ее роль в процессе планирования расписания ИТ-проектов?
6. Как используется список задач в процессе планирования расписания ИТ-проектов?
7. Что собой представляет электронный календарь? Каким образом он может быть использован при планировании расписания?
8. Что является отличительной чертой диаграммы Ганта? Почему она так популярна?
9. Как построить диаграмму Ганта?
10. Как построить диаграмму по методу критического пути?
11. Как построить диаграмму контрольных событий?
12. Какие процессы включает управление человеческими ресурсами ИТ-проекта?
13. Какие виды ИТ-команд существуют?

| Глава 3

14. В чем особенность оперативных групп?
15. В чем особенность проектных групп?
16. В чем особенность групп поддержки?
17. В чем особенность команд процессов?
18. Назовите заинтересованные стороны, которые могут оказаться задействованными в проекте.
19. Каковы основные принципы управления командой?
20. Каковы основные процессы управления командой ИТ-проекта?

Глава 4

Управление рисками и финансовое обоснование ИТ-проекта

4.1. Основные понятия управления рисками ИТ-проекта

ИТ-проекты, особенно проекты разработки программного обеспечения, различаются по размеру и сложности. Чем сложнее проект, тем выше риск возникновения событий в течение его жизненного цикла, которые могут оказать положительное или отрицательное влияние на его цели. Другими словами, рискованные события влияют на время, стоимость, объем и качество результатов проекта. Примером рискованных событий являются значительные изменения в объемах работ, сбои в работе продуктов, произведенных в рамках проекта, конкретные задержки из-за отказа от работы или отсутствия рабочей силы, судебные разбирательства против компании и т. д. [18].

Управление рисками рассматривается и как форма страхования, и как инвестиции. Управление рисками связано с будущими событиями в ИТ-проекте, особенно в проектах разработки программного обеспечения, точный результат которых неизвестен, и с тем, как заранее справиться с этими неопределенностями. Это также искусство и наука планирования, оценки (выявления и анализа), обработки, мониторинга и реагирования на риски на протяжении всего срока реализации проекта и в будущем.

Необходимо понимать потенциальные рисковые события и то, как они могут помешать успеху проекта.

Определение риска

В РМВоК под риском понимается неопределенное событие или условие, которое, если оно происходит, оказывает положительное или отрицательное влияние на цели проекта [5].

Также риск определяется как возможность потери, повреждения или любого другого нежелательного события. Риск ИТ-проекта включает в себя понимание потенциальных проблем, которые могут возникнуть в проекте, и того, как они могут помешать успеху проекта.

Конкретные события, которые могут произойти в ущерб проекту, известны как события риска.

Виды рисков

Существует 3 типа рисков, которые могут возникнуть [19]:

- известные риски;
- предсказуемые (прогнозируемые) риски;
- непредсказуемые риски.

Известные риски — те, которые могут быть выявлены после тщательной оценки плана проекта, деловой и технической среды, в которой разрабатывается проект, и других надежных источников информации (например, известные риски — это нереалистичная дата поставки, отсутствие документированных требований или объема программного обеспечения, плохая среда разработки).

Прогнозируемые риски экстраполируются из прошлого опыта проекта (например, это текучесть кадров, плохая коммуникация с заказчиком, ослабление усилий персонала по мере обслуживания текущих запросов на техническое обслуживание).

Могут возникнуть непредсказуемые риски, но их чрезвычайно трудно определить заранее. Путем выявления известных и предсказуемых рисков руководитель проекта делает первый шаг к тому, чтобы избежать их, когда это возможно, и контролировать их, когда это необходимо.

Основные риски, связанные с типичными ИТ-проектами, следующие:

1) риски, связанные с содержанием проекта. Они включают изменения в области применения, при которых проект

усложняется, проблемы интеграции, дефекты аппаратного и программного обеспечения, а также изменение зависимостей между участниками проекта;

2) риски планирования. Расписание ИТ-проекта часто не соблюдается, когда задачи проекта и риски несоблюдения расписания не согласованы должным образом. Это может быть связано с неправильной оценкой времени и ресурсов, неспособностью сразу определить сложные функции продукта и время необходимое для разработки этих функций, а также с неожиданным расширением ИТ-проекта;

3) риски, связанные с ресурсами. В основном они связаны с аутсорсингом и проблемами с персоналом. Считается, что привлечение нового работника на более позднем этапе может значительно замедлить проект;

4) технологические риски. Как правило, они приводят к сбою функциональности и производительности. Причины технологических рисков — постоянное изменение требований, отсутствие передовой технологии или незрелость применяемой технологии, сложность реализации проекта, а также трудности в интеграции модулей проекта.

Цели управления рисками:

- 1) уменьшить вероятность возникновения рисков;
- 2) минимизировать потенциальные риски и максимизировать потенциальные возможности;
- 3) смягчить последствия рисков.

План управления рисками

План управления рисками документирует процедуры управления рисками на протяжении всего проекта. Также существует план действий на случай непредвиденных обстоятельств, которые команда проекта предпримет, если произойдет рисковое событие.

Резервы на случай непредвиденных обстоятельств — это резервы, удерживаемые спонсором проекта, которые могут быть задействованы при возможных изменениях в объеме или качестве проекта для снижения затрат и (или) риска планирования.

Процесс управления рисками

Процесс активного управления рисками включает следующие шаги:

- 1) идентификация рисков — определение того, какие риски могут повлиять на проект;
- 2) качественный и количественный анализ рисков — оценка диапазона влияния рисков на результаты проекта;
- 3) разработка мер реагирования на риски — принятие мер по расширению возможностей и реагированию на угрозы;
- 4) контроль рисков — реагирование на риски в ходе проекта.

4.2. Идентификация рисков ИТ-проекта

Идентификация рисков известна как процесс понимания того, какие потенциальные неудовлетворительные результаты связаны с конкретным проектом, и документирования их характеристик. Выявляются и описываются их основные причины, указываются индикаторы, которые будут предупреждать об их возникновении, подробно описываются шаги по смягчению последствий в текущих планах, перечисляются их владельцы для будущих действий и обсуждаются их последствия.

Идентификация рисков проводится людьми, знакомыми с проектом, и, возможно, другими лицами, обладающими опытом работы в аналогичных проектах или в качестве независимых экспертов.

Методологии идентификации ИТ-рисков

Существует несколько различных способов определения рисков [20]:

- 1) исторический. Наиболее распространенный, в том числе в страховых компаниях. Причину риска можно установить, основываясь на эмпирических данных о том, какие события и с какой частотой происходили ранее. Во время идентификации и оценки рисков специалисты по управлению рисками всегда должны использовать исторические данные, если это возможно. То, что произошло в прошлом, поможет убедиться, что та же проблема не повторится в будущем, когда этого никто не ожидал. Но не для всех ИТ-проектов это работает;

2) системный подход. Предполагает привлечение экспертов по предмету, людей, действительно разбирающихся в технологиях, имеющих представление о среде угроз и способных выразить экспертное мнение о том, какие проблемы могут быть связаны с технологиями, которыми в настоящее время обладает субъект. В некоторых случаях они даже прогнозируют события, которые еще не произошли. В данной методологии с целью сформировать рисковое событие рассматривают не только одиночные точки отказа, но и агрегированный риск от различных источников, которые функционируют одновременно;

3) индуктивный или теоретический анализ. Также построен на системном подходе, но индуктивно направлен на действия, которые необходимо произвести, в тех областях или случаях, где еще никто не является экспертом. Например, когда есть новая технология или новый бизнес-процесс, невозможно всегда знать, какие типы угроз будут возникать. Но, используя системный подход экспертного мнения, воображение и способность предвидеть, как будут развиваться события, можно провести успешный индуктивный анализ. При этом могут возникнуть новые идеи, и в некоторых случаях можно выявить типы событий, которые способны привести к рисковому событию в будущем.

Идентификация рисков помогает во многих случаях обосновать, должна ли организация сделать определенные инвестиции или нет. Инвестиции в ИТ часто являются очень большими капиталовложениями. Это также большие операционные расходы. В любом случае, рассматривая текущие риски, управление бизнесом должно быть в состоянии определить, стоит инвестировать в новую систему или нет.

В бизнесе есть серьезный риск получения третьим лицом несанкционированного доступа к конфиденциальным данным. Кража конфиденциальных данных может привести к тяжелым финансовым последствиям — штрафам, а также репутационному ущербу. Кроме того, необходимо понимать, какие риски связаны с достоверностью. Есть ли вероятность, например, что данные, которые сотрудник имеет в своей системе, будут неточными? Такие вещи должны быть рассмотрены на этапе идентификации риска. Каков риск неудачных инвестиций? Каков риск,

связанный с несанкционированным доступом и отсутствием безопасности? Каковы риски, связанные с потерей целостности?

На этапе идентификации рисков бизнес распознает угрозы, уязвимости и активы, связанные с ИТ-проектом. Идентификация риска вместе с оценкой риска отвечает за определение стоимости активов. Основными методами идентификации рисков являются байесовский подход, метод «галстук-бабочка», метод мозгового штурма, структурированные интервью. Используя эти методы, специалист по управлению рисками заимствует информацию из области обеспечения непрерывности бизнеса и аварийного восстановления и проводит анализ их влияния на проекты в прошлом. Также специалист по управлению рисками может сопоставить причины и следствие рисков, и по результатам оценить, что может привести к событию и каким будет его влияние. В данном случае речь идет о причинно-следственном анализе. Специалист по управлению рисками стремится понять первопричины рисков и то, как эти причины могут влиять на ИТ-проект.

Одним из методов, опирающихся на исторический опыт, является анализ контрольных списков. Метод основан на прецедентах рискованных событий в аналогичных проектах в прошлом.

Можно воспользоваться советами опытных людей в организации, которые понимают влияние рисков на бизнес, и это часто делается с помощью метода Дельфи. Метод Дельфи применяется, когда требуется учесть мнение всех заинтересованных сторон, иногда даже анонимно, чтобы каждый мог внести свой вклад в процесс сбора и анализа данных.

Бывает, что многократно повторяемое рискованное событие происходит вследствие различных событий. В таких случаях анализ дерева событий может помочь определить, какова совокупность обстоятельств, которые, собираясь вместе, приводят к рискованному событию. Одной из разновидностей данного метода является анализ дерева ошибок.

Специалист по управлению рисками также может использовать анализ критических контрольных точек, в которых в рамках организации выявляется высокий уровень риска, способный привести к очень серьезным неблагоприятным последствиям, часто основанным на единичных точках отказа.

Также существует метод эксплуатационных исследований, однако он хранит в себе определенную опасность. Можно рассматривать исторический опыт, анализировать технологии в рамках бизнеса и в соответствии с ними строить стратегию управления рисками, однако нельзя забывать, что многие риски связаны с персоналом, человеческим фактором, соответственно, приобретает значимость анализ надежности персонала.

Одним из способов, позволяющих организации по-настоящему эффективно оценивать риски, является изучение влияния технического обслуживания на надежность систем и продуктов.

Большинство людей не понимают оценку рисков, но они понимают сценарий, историю или пример. Специалист по управлению рисками может использовать сценарий или примеры для сбора более точной информации, позволяя людям лучше понять, каково может быть влияние различных типов событий. Такой подход называется структурированной техникой «что, если».

Несмотря на многообразие методик идентификации рисков, часто результаты оказываются неточными. В дальнейшем это может привести к неправильным решениям при реагировании на риск. Это связано с тем, что во многих случаях риск непредсказуем. Одно и то же событие может произойти 15 раз, и все 15 раз могут иметь совершенно разный уровень последствий. Или, что еще хуже, одно и то же событие может произойти 15 раз, 14 раз имея одинаковые последствия, а один раз совершенно другой результат.

Трудно рассматривать рисковое событие изолированно. Оценка риска работает на больших выборках, но не на отдельном случае. Поэтому следует извлекать уроки из каждого рискового события, которое произошло в прошлом.

4.3. Качественный и количественный анализ рисков

Количественная оценка (анализ) рисков — это процесс оценки рисков с целью выявления диапазона возможных результатов проекта. Этот процесс основан на изучении влияния нескольких оценок рисков, чтобы лучше понять важнейшие аспек-

ты проекта, такие как среднее влияние рисков, минимальное и максимальное воздействие рисков, определение приоритетов рисков, анализ чувствительности и многое другое. Количественный анализ, как правило, обеспечивает «моментальный снимок» во времени среды риска проекта.

Количественный анализ рисков часто проводится с помощью моделирования по методу Монте-Карло. При моделировании используется представление или модель системы для анализа ожидаемого поведения или производительности системы.

Анализ методом Монте-Карло многократно моделирует результат модели, чтобы обеспечить статистическое распределение рассчитанных результатов. Моделирование основано на рисках, которые идентифицируются и количественно оцениваются в виде вероятностных распределений на детальном уровне проекта, таком как степень активности расписания или затраты на реализацию иерархической структуры проектных мероприятий низкого уровня. Моделирование далее объединяет распределения вероятностей неопределенных результатов и уровень детализации с использованием моделей проекта, таких как график критического пути. Полученные результаты включают в себя:

- 1) вероятность достижения цели проекта в назначенную дату;
- 2) непредвиденные обстоятельства, которые необходимо учесть для достижения приемлемого уровня вероятностных результатов проекта;
- 3) наличие в рамках проекта рисков, которые представляют наибольшую угрозу.

Идея количественной оценки риска состоит в том, что организации пытаются оценить весь риск в денежном выражении. Однако проблема заключается в том, что многие аспекты не могут быть измерены количественно, например, доверие клиентов и моральные качества сотрудников. Поэтому довольно часто риск рассматривают с качественной точки зрения.

Качественная оценка рисков — методика, при которой специалист по управлению рисками рассматривает сценарий. Если одна система выйдет из строя, как это повлияет на другие отделы? Как правило, в данном случае риски оцениваются с точки зрения очень низкого, умеренного, высокого или очень

высокого воздействия. Вместо того чтобы оценить их в денежном выражении, специалист по управлению рисками смотрит на них с точки зрения целого ряда ценностей, которые они могут затронуть.

Он ведет переговоры со многими отделами, чтобы выяснить, как тот или иной сценарий повлияет не только на один отдел, но и на другие отделы в организации. Далее производится сопоставление диапазона уровней риска из разных областей, и после этого специалист по управлению рисками может устанавливать приоритеты в соответствии с рассматриваемым риском. В результате риск рассматривается с точки зрения перспективы, а не только денег, и вообще в большинстве случаев мы оцениваем не количественные значения рисков.

Количественный риск рассчитывается прежде всего исходя из того, во сколько обойдется бизнесу одно-единственное событие. Довольно часто такой риск определяют как ожидание одиночных потерь.

Оценивая качественный риск по различным типам сценариев, специалист по управлению рисками может рассматривать вещи, которые находятся за пределами денежных ценностей, такие как мораль, репутация, доверие клиентов.

Как количественный, так и качественный анализ рисков имеют свои преимущества и недостатки.

Количественный подход не охватывает важные неденежные факторы, в то время как качественный подход не дает нам стоимостной оценки необходимой для оправдания затрат на контроль. Поэтому часто организации объединяют подходы в гибрид, в котором проводят полуколичественную оценку рисков, сравнивая как количественные, так и качественные показатели, чтобы получить более полную картину риска.

4.4. Меры реагирования на риски

Меры реагирования на риски могут быть классифицированы следующим образом:

- предотвращение риска — устранение причин конкретной угрозы или риска;

- принятие риска — готовность и компенсация последствий в случае возникновения риска;
- снижение риска — уменьшение воздействия рискового события за счет снижения вероятности его наступления;
- перенос риска.

Существует три области рисков, которые могут повлиять на успех ИТ-проекта.

1. Риски проекта, т. е. то, что угрожает плану проекта. Если риски проекта станут реальными, вполне вероятно, что график его реализации сорвется и затраты увеличатся. Риски проекта определяют потенциальные проблемы с бюджетом, графиком, персоналом, ресурсами, заинтересованными сторонами и требованиями к проекту, а также их влияние на ИТ-проект. Примеры проектных рисков: сложность проекта, его размер и степень структурной неопределенности.

2. Технические риски. Они ставят под угрозу качество и своевременность создаваемого программного обеспечения. Если технический риск становится реальным, реализация всего проекта может стать трудной или невозможной задачей. Технические риски определяют потенциальные проблемы проектирования, внедрения, интерфейса, проверки и технического обслуживания. Кроме того, неоднозначность спецификаций, техническая неопределенность и техническое устаревание также известны как технические риски.

3. Бизнес-риски. Они угрожают жизнеспособности создаваемого программного обеспечения. Бизнес-риски часто ставят под угрозу проект или продукт. Основными бизнес-рисками являются:

- создание отличного продукта или системы, которые на самом деле никому не нужны (рыночный риск);
- создание продукта, который больше не вписывается в общую бизнес-стратегию компании (стратегический риск);
- создание продукта, который сложно продавать из-за непонимания продавцов (риск продаж);
- потеря поддержки ИТ-проекта высшим руководством из-за изменения приоритетов или вследствие смены руководства (управленческий риск);

– потеря бюджетной или частной финансовой поддержки (бюджетный риск).

Частично снизить риск способны следующие базовые стратегии:

1) для тех рисков, которые команда проекта может контролировать, надо использовать все ресурсы необходимые для их снижения;

2) риски, которые находятся вне контроля команды проекта, следует попытаться перенести.

Возможные способы переноса риска:

- переход на другое оборудование;
- перемещение функции программного обеспечения в другое подразделение системы, которое лучше справляется с ней;
- передача работы на субподряд более опытному разработчику.

4.5. Контроль рисков

Контроль рисков включает в себя процессы управления рисками и план управления рисками для реагирования на рискованные события. Риски должны контролироваться на основе определенных решений и стратегий смягчения последствий. В тех случаях, когда нет планов действий в чрезвычайных ситуациях, необходимы обходные пути или незапланированные реакции на рискованные события.

Отслеживание топ-10 элементов риска является одним из инструментов для поддержания осведомленности о риске на протяжении всего срока реализации проекта. Для этого необходимо отслеживать 10 основных элементов риска проекта. Затем следует свести текущее ранжирование, предыдущее ранжирование, количество рискованных случаев в списке топ-10 за определенный период времени и информацию о прогрессе, достигнутом в устранении элемента риска.

Полезно использовать сводные базы данных для отслеживания рисков. Электронные таблицы также могут использоваться для отслеживания и количественной оценки рисков.

Для контроля и реагирования на риски используются следующие исходные данные:

- план управления рисками;
- список приоритетных рисков;
- список приоритетных количественных рисков;
- вероятностный анализ проекта;
- список потенциальных ответных мер;
- владельцы рисков;
- ранжирование рисков проекта;
- вероятность достижения целей по затратам и времени;
- пороговые значения рисков;
- общие причины риска;
- тенденции в результатах количественного анализа

рисков.

Основные стратегии реагирования на риски:

- предотвращение рисков;
- перенос рисков;
- смягчение рисков;
- принятие рисков.

Результаты разработки стратегии реагирования:

- план реагирования на риски;
- остаточные риски;
- вторичные риски;
- договорные соглашения;
- суммы резерва на непредвиденные расходы, необходимые материалы для других процессов;
- материалы для пересмотренного плана проекта.

Наиболее распространенные источники риска для ИТ-проектов:

- миссия и цели;
- драйверы принятия решений;
- управление организацией;
- клиент или конечный пользователь;
- бюджет;
- расписание;
- характеристики проекта;
- процесс разработки;
- среда разработки;

- персонал;
- оперативная среда;
- новые технологии.

Все источники рисков, упомянутые выше, фактически можно разделить на шесть более общих категорий:

1) размер проекта. Большим и сложным ИТ-проектом разработки программного обеспечения сложно управлять, и будет очень сложно завершить его вовремя и в рамках выделенного бюджета и ресурсов. Это связано с большим количеством индивидуальных задач и взаимозависимостей в рамках проекта, а также с необходимостью вовлечения в работу большого количества людей. Следовательно, требуется более высокий уровень координации. Обычно ожидается, что крупный проект принесет значительные выгоды для бизнеса. Но, к сожалению, по разным причинам не все крупные проекты способны оправдать ожидания;

2) сложность проекта. Сложность также может привести к перерасходу ресурсов ИТ-проекта, поскольку он будет зависеть от:

- разнообразия бизнес-функций, которые будут затронуты (с последствиями для процесса управления организационными изменениями);

- интеграции с другими информационными системами (с последствиями для процесса управления техническими изменениями);

- технической сложности самой системы (возможно, из-за необходимости использовать только недавно разработанную технологию, которая ранее не использовалась в организации);

3) проблемы с людьми. Они связаны с тем, в какой степени высшее руководство привержено проекту и участвует в нем. Необходимо найти правильное сочетание деловых и технических навыков, должна появиться эффективная коммуникация между бизнес-пользователями и поставщиками системы. Пример таких проблем — программисты с необходимым набором навыков недоступны, и существует недостаток информации от конечных пользователей;

4) контроль проекта. Временные, стоимостные и качественные аспекты проекта должны тщательно контролироваться.

Определение основных этапов проекта, ИТ-стандартов, разработка систем и методологий управления проектами, бюджетный контроль и управление изменениями имеют большое значение. Жестко спланированный проект может не приспособиться к неожиданным изменениям, поэтому важна гибкость. Риск будет заключаться в точности процесса планирования и степени, в которой изменения могут быть адаптированы;

5) новизна. Другой потенциальный риск заключается в том, что потребности бизнеса или используемые технологии меняются так быстро, что система не сможет предоставить то, что требуется. Однако поскольку рассматриваемый проект призван обеспечить значительное конкурентное преимущество, может возникнуть необходимость в таких рисках. Это общая проблема баланса между риском и вознаграждением;

6) стабильность требований. Чем выше степень определенности с точки зрения бизнеса и техники, тем ниже уровень риска, связанного с проектом. Когда требования четко не определены, затраты и ожидаемые выгоды труднее измерить. В дополнение к этому динамичная бизнес-среда может означать, что основные потребности бизнеса меняются относительно быстро, и это будет отражено в изменении требований к системам.

Существует семь принципов, обеспечивающих эффективное управление рисками (табл. 11).

Т а б л и ц а 11

Принципы эффективного управления рисками

Принцип	Описание
1. Глобальная перспектива	Просмотр разработки программного обеспечения в контексте определения, проектирования и разработки на уровне более крупных систем. Признание как потенциальной ценности возможностей, так и потенциального воздействия неблагоприятных последствий
2. Ориентация на будущее	Размышление о завтрашнем дне, выявление неопределенностей, предвосхищение потенциальных результатов. Управление ресурсами и деятельностью проекта при одновременном предупреждении неопределенностей
3. Открытая коммуникация	Поощрение свободного обмена информацией на всех уровнях проекта и между ними.

Принцип	Описание
	Обеспечение формального, неформального и импровизированного общения. Использование процессов, в которых ценятся индивидуальные особенности (предоставление уникальных знаний и понимания для выявления и управления рисками)
4. Интегрированное управление	Управление рисками — неотъемлемая и жизненно важная часть управления проектами. Адаптация методов и инструментов управления рисками к инфраструктуре и культуре проекта
5. Непрерывный процесс	Поддержание постоянной бдительности. Регулярное выявление и управление рисками на всех этапах жизненного цикла проекта
6. Общее видение продукта	Общее видение продукта, основанное на общей цели, совместной собственности и коллективной коммуникации. Сосредоточение внимания на результатах
7. Командная работа	Совместная работа для достижения общей цели. Объединение талантов, навыков и знаний

4.6. Основные методы и инструменты контроля качества ИТ-проекта

Хотя качество является одним из ограничений общих для всех ИТ-проектов, эта область не всегда получает такое же внимание, как объем работ, бюджет или график. Однако отсутствие управления качеством может оказать серьезное негативное влияние на проект [21].

Контроль качества — это процесс рассмотрения результатов ИТ-проекта для подтверждения соответствия установленным стандартам и внесения соответствующих изменений для устранения причин неприемлемого качества. План управления качеством является основой для конкретных мероприятий, осуществляемых в ходе контроля качества. Мероприятия по обеспечению качества, процедуры, используемые для выполнения мероприятий по обеспечению качества, и необходимые ресурсы документируются в плане управления качеством.

Контроль качества осуществляется на протяжении всего проекта. Инструменты и методы обеспечения качества исполь-

зуются для определения соответствия минимальному стандарту достаточности, а мероприятия по обеспечению качества часто являются ключевой частью официального процесса утверждения завершения этапа.

Основными инструментами, которые используются в контроле качества, являются: инспекция (тестирование); диаграмма Парето; контрольная диаграмма; статистическая выборка; блок-схема; анализ тенденций.

Инспекция (тестирование)

Инспекция — это широкая категория, которая включает в себя изучение, измерение или тестирование результатов работы. Наиболее распространенным методом проверки ИТ-проектов является тестирование. Тестирование — это то, чему всегда надо стараться уделять дополнительное время. Тестирование — это скучная работа, кто-то должен запускать код снова и снова, тестируя разные аспекты в разных модулях, делая заметки о его производительности. Но это должно быть сделано. В вашем распоряжении должны быть тестировщики, которые готовы тщательно проработать ваш новый код.

Тестирование проводится на протяжении всего проекта. Несколько видов тестирования являются общими для ИТ-проектов:

– тестирование отдельных модулей. Программист завершает модуль кода и должен его протестировать. Некоторые модули не поддаются тестированию сепарированно, поскольку они предназначены для взаимодействия с другими модулями. Тем не менее программист может проверить, что переменные загружаются правильно, что код идет в те места, куда он должен идти в зависимости от ввода, что память очищается и программа правильно завершает работу. Часто разработчики могут запускать код с помощью средства проверки, которое пошагово проверяет код, чтобы увидеть, как он ведет себя и как загружает переменные памяти. Процесс может быть одновременно очень интересным и в то же время чрезвычайно неприятным, потому что если у разработчика что-то идет не так, он не всегда может понять, где именно. На этом этапе библиотеки динамических ссылок (DLL) и другие информационные файлы также тестируются на полноту и точность содержимого;

– модульное тестирование. После того как несколько модулей, которые идут вместе, были удовлетворительно протестированы, разработчики могут протестировать их все сразу в модульном тестировании. Они могут протестировать целую систему печати или набор алгоритмов, которые что-то вычисляют, а также функциональность частей, которые были собраны вместе, чтобы сформировать сплоченную группу модулей;

– системное тестирование. Затем разработчики тестируют всю систему в целом. Они следят за тем, чтобы система работала должным образом, чтобы пользователь не сталкивался с неожиданными циклами или ошибками, чтобы система была быстрой и функциональной и обеспечивала то, что ожидает клиент. Тестирование системы должно занять много времени, потому что тщательно проверяются каждый компонент и целое;

– приемочное тестирование пользователей (UAT). Это время, когда фактически привлекается небольшой набор пользователей, чтобы начать тестирование результата. К тому времени, когда разработчики доберутся до UAT, они должны будут устранить все ошибки, проблемы со скоростью исполнения или логикой, а также проблемы с потоком. Система должна быть в самом лучшем состоянии. Как ожидается, пользователь будет использовать ее в производстве до тех пор, пока тестировщики UAT не обнаружат проблемы, которые пропустили другие;

– заводское приемочное тестирование (FAT). Иногда система настолько велика, что лучше тестирование пользователей проводить на том месте, где была выполнена первоначальная разработка. Например, разработка новой радиолокационной системы для прогнозирования погоды, в которой разрабатывается новое оборудование и новое программное обеспечение. Все это осуществляется на сайте поставщика при подготовке к доставке в новое место после того, как клиент принял новую систему;

– приемочные испытания на месте. Включают в себя тестирование на территории заказчика. Рассмотрим пример радиолокационной системы, упомянутый выше. Специализированный подрядчик по радиолокации разработал новую радиолокационную систему и все ее элементы. Клиент приходит на фабрику и находит систему приемлемой. Теперь компания поставляет новую систему заказчику, и они вместе проходят еще один раунд

тестирования, чтобы убедиться, что система работает должным образом.

Тестирование является важным шагом в подтверждении качества ИТ-проекта, и в некоторых сценариях может потребоваться провести еще больше сеансов тестирования.

Географически рассредоточенные команды проектных групп часто распределены по нескольким местам. Основным результатом работы программного обеспечения может быть код, скомпилированный из модулей, написанных несколькими командами программистов. Несмотря на то что были установлены стандарты связи и работы для команд, отсутствие ежедневных личных контактов может увеличить риск неправильного толкования и привести к различиям в модулях, что повлияет на скомпилированный пакет. В этой ситуации может потребоваться более подробное тестирование отдельных модулей, чтобы выявить проблемы на ранней стадии.

Другой сценарий, в котором тестирование приобретает важное значение, — это когда поставщик производит конечный продукт. Особенности того, как и когда вы можете тестировать в процессе разработки, должны быть включены в контракт. Интеграционное тестирование поставляемого поставщиком продукта с остальной частью системы обычно является требованием для окончательной приемки работы поставщика. Тесты должны охватывать все требования, связанные с работой, на выполнение которой был заключен контракт с поставщиком.

Хотя тестирование наиболее часто используется для контроля качества в ИТ-проектах, это не единственный доступный метод. Любой, кто участвовал в ИТ-проекте, вероятно, в какой-то степени знаком с объемом тестирования, проводимого в ходе проекта. В сочетании с тестированием могут использоваться другие инструменты и методы для устранения дефектов качества.

Диаграмма Парето

Диаграмма Парето используется для ранжирования важности проблемы на основе частоты ее возникновения с течением времени. Эта диаграмма основана на принципе Парето, который чаще называют правилом 80/20. Принцип Парето назван в честь Вильфредо Парето, итальянского социолога и экономиста, который заметил, что 80 % богатства в Италии принадлежит 20 %

населения. С тех пор как Парето впервые открыл его, этот принцип был применен ко многим дисциплинам. Применительно к контролю качества этот принцип говорит, что большинство дефектов проекта вызваны небольшим набором проблем. Диаграмма Парето помогает выделить основные проблемы, чтобы вы могли предпринять действия, которые окажут наибольшее влияние. Гистограмма используется для отображения проблем в порядке убывания их возникновения, чтобы можно было установить приоритеты для улучшения.

Назначение диаграммы Парето двоякое:

- 1) она отображает относительную важность дефектов;
- 2) она направляет усилия по совершенствованию в те области, которые окажут наибольшее влияние.

Построение диаграммы Парето обычно начинается с таблицы, в которой перечислены сведения о частоте дефектов или сбоев, обнаруженных во время тестирования. В табл. 12 показана частота отказов для элементов А–Е.

Т а б л и ц а 12

Частота отказов

Элемент	Частота случаев дефекта	Дефекты, которые представляет этот элемент, %	Совокупный процент дефектов
А	800	0,33	0,33
В	700	0,29	0,62
С	400	0,17	0,79
Д	300	0,13	0,92
Е	200	0,08	1,00

Имея эти данные, можно создать диаграмму Парето, как показано на рис. 32. Столбцы упорядочены слева направо в зависимости от частоты.

Взглянув на рис. 32, вы можете увидеть, что наиболее значимыми проблемами, на которых надо сосредоточиться, являются А и В. Исправление этих двух элементов устранит более половины дефектов.

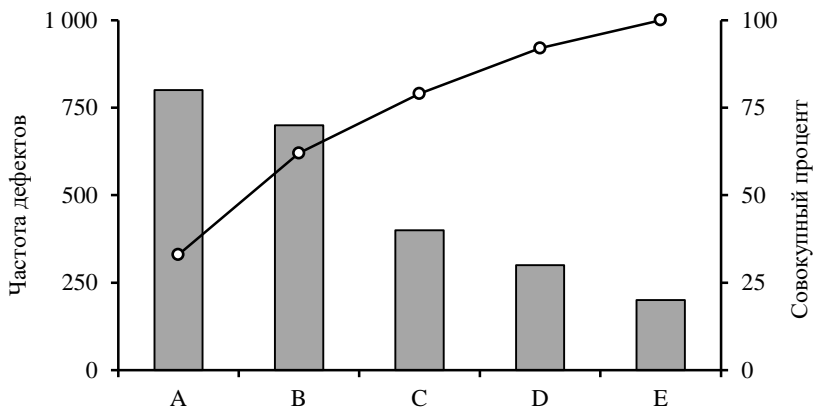


Рис. 32. Диаграмма Парето

Контрольная диаграмма

Контрольная диаграмма — это изображение дисперсии нескольких выборок одного и того же процесса с течением времени. Чаще всего используется в производстве. Контрольная диаграмма основана на среднем значении, верхнем контрольном пределе и нижнем контрольном пределе. Верхний контрольный предел — это точка, за которой предотвращение дополнительных дефектов становится непомерно дорогостоящим. Нижний уровень — это предел, при котором клиент или конечный пользователь отклонит продукт из-за дефектов. Цель состоит в том, чтобы оставаться в средней области (среднее значение), где получается лучший продукт с наименьшими затратами. Пример контрольной диаграммы показан на рис. 33.

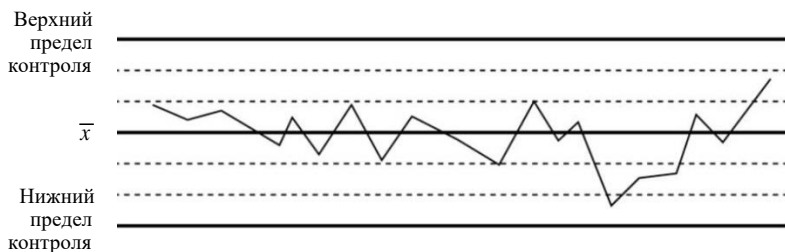


Рис. 33. Контрольная диаграмма

Статистическая выборка

Если имеются многочисленные результаты работы, требующие проверки или тестирования, можно попробовать использовать статистическую выборку, которая собирает подмножество всех применимых результатов работы и случайным образом выбирает небольшое количество для тестирования или проверки. Статистическая выборка может быть очень экономичной, особенно в проектах, где производится несколько версий одного и того же продукта [21].

Блок-схема

Блок-схема есть средство создания процесса, который производит продукт. Блок-схемы также могут быть эффективным инструментом во время контроля качества, чтобы помочь определить, как возникла проблема.

Анализ тенденций

Анализ тенденций — это математический метод, который может быть использован для прогнозирования будущих дефектов на основе исторических результатов.

Результаты, полученные в результате тестирования или применения других инструментов и методов контроля качества, используются для определения того, следует ли предпринимать какие-либо действия для исправления низкого качества.

4.7. Меры по контролю качества ИТ-проекта

Контроль качества ИТ-проекта начинается с одной из двух альтернатив: 1) контроль на основе хорошо известных, широко распространенных стандартов для выполняемых типов работ; 2) настройка процессов внутренней среды таким образом, чтобы был гарантирован успех проекта.

Оба варианта требуют участия на всех уровнях.

Стандарты

Когда интернет-разработчики объединяются в группу и выработывают стандарты, результатом почти всегда является более последовательный способ ведения бизнеса. Возможны, например, следующие стандарты разработки приложений:

1) все новые приложения будут разрабатываться таким образом, чтобы они запускались из браузера, использующего XML;

2) все приложения будут использовать серверные веб-службы, такие как J2EE, а не локализованные службы;

3) все приложения будут использовать протокол простого доступа к объектам (SOAP) в качестве метода передачи от приложения к клиенту.

Эти стандарты, конечно, не являются всеобъемлющими и могут не соответствовать стандартам, которые установил бы конкретный интернет-разработчик, но они фундаментальны, просты для понимания и используют общепринятые протоколы в качестве связующего звена. Устанавливая эти стандарты, мы получаем надежный способ управления новыми приложениями в масштабах всего предприятия.

При создании организации, которая будет заниматься ИТ-проектами, разумно собрать всех заинтересованных в ней лиц и посмотреть, возможно ли составить список общих стандартов, которых будут придерживаться все команды организации. Эти стандарты могут включать:

- используемые серверы и построение коммуникации между ними;

- стандарты установки рабочих станций, т. е. версии ОС и пакета MS Office, исправления и пакеты обновления, настройки браузера, настройки панели управления, профили и т. д.;

- стандарты разработки приложений;

- стандарты работы сети, включая приемлемые сетевые протоколы.

Организации по стандартизации

Организации по стандартизации разрабатывают стандарты практически для всего, что только можно себе представить, и сфера ИТ не исключение. Наиболее влиятельными организациями по стандартизации являются [21]:

1) Международная организация по стандартизации (ISO). Это ведущая организация, в частности, ISO 9000 может использоваться (и используется) корпорациями для управления контролем качества;

2) Целевая группа по распределенному управлению (DMTF). Эта организация ориентирована на нахождение способов управления системами с помощью стандартизированного интерфейса. Microsoft внедрила этот стандарт в свой продукт

Windows с помощью стандарта Web-Based Enterprise Management (WBEM), который использует интерфейс управления и базу данных на каждом компьютере с Windows 2x и выше. WBEM используется сервером Microsoft Systems Management Server (SMS) для сбора информации о ресурсах с ПК и загрузки данных в центральную базу данных с целью ведения централизованной системы инвентаризации;

3) Институт инженеров электроники и электротехники (IEEE) — организация, играющая важную роль в создании широко адаптированных сетевых протоколов. Набор протоколов беспроводной сети 802.11 — это стандарт IEEE, который развился в результате длительной работы экспертов комитета;

4) Библиотека инфраструктуры информационных технологий (ITIL). Она сосредотачивает свою философию на наилучшем способе управления операционной средой (серверы, сетевая инфраструктура, мэйнфреймы и т. д.). Эта организация имеет стандарты, которые помогут вам в разработке отличных операционных методологий;

5) американский национальный институт стандартов (ANSI). Например, руководство PMI по PMBoK является стандартом ANSI.

Настройка процессов внутренней среды

Есть много способов создать организацию по разработке ИТ-проектов. Возможно, первое, с чего следует начать — это оценить организацию, насколько она на самом деле способна создавать качественные проекты. Здесь может быть использована модель зрелости возможностей (Capability Maturity Model — CMM). CMM была создана Институтом разработки программного обеспечения (SEI) при Университете Карнеги-Меллона (CMU).

Идея CMM заключается в том, что пять операционных уровней описывают усилия по разработке проектов любой организации: начальные, повторяемые, определенные, управляемые и оптимизирующие [21]:

– начальные усилия. Носят случайный характер и иногда даже хаотичны. Как правило, в этих процессах нет никакого прагматизма и здравого смысла. Организация берется за проект и пытается «взять его штурмом». Это самый низкий уровень CMM. Организации на данном этапе имеют условный статус CMM 1;

– повторяемые усилия. На этом уровне организация устанавливает основные процессы управления проектами, которые служат для определения затрат, графиков и прогнозируемых результатов любого конкретного проекта. На таком уровне организации присваивается статус СММ 2. Она считается «зрелой для проекта», но определенно предпринявшей шаги по систематизации своих процессов;

– определенные усилия. В СММ 3 все процессы были определены, опубликованы, стандартизированы и введены в эксплуатацию. Все проекты используют стандартную документацию, проектные запросы, утверждения и т. д. для достижения обозначенных результатов. Организации в стадии СММ 3 уже систематизировали свою внутреннюю среду, но им еще предстоит пройти длинный путь, прежде чем они станут полноценно функциональными;

– управляемые усилия. Организации на стадии СММ 4 количественно контролируют результаты своих проектов. Любой проект подвергается строгому количественному контролю для оценки качества результатов. В крупных организациях и проектах это происходит с помощью метода шести сигм. Шесть сигм — это программа, которая учит контролировать качество результатов с помощью статистического анализа; в ней используются различные цвета поясов боевых искусств, чтобы обозначить, как далеко вы продвинулись в программе. Контроль в таких организациях жесткий, операции четко определены, а проекты выполняются единообразно и с неизменным качеством. Ключевое слово в СММ 4 — «количественно». На этом уровне применяются количественные показатели и анализ, чтобы гарантировать высокое качество продукции;

– оптимизирующие усилия. Как и в случае с высшей ступенью (самореализация) знаменитой иерархии потребностей Абрахама Маслоу, организация в СММ 5 не только генерирует проекты неизменно высокого качества, которые умело контролируются и управляются, но и постоянно ищет способы улучшения процессов.

Ключ к успеху любого ИТ-предприятия заключается в способности эффективно внедрять методологии управления проектами и осуществлять проекты с определенной последова-

тельностью действий. Если вы сначала поймете, в каком состоянии находится ваша организация по сравнению с ее возможностями, и приведете свои усилия в соответствие с хорошо известными стандартами качества, у вас будет высокий шанс на успех.

Очень важно всегда понимать процесс или потоки бизнес-подразделения, прежде чем применять технологию. Если будет соблюдаться этот порядок, все остальное должно стать на свои места.

Также по мере осуществления мероприятий по контролю качества необходимо принимать решения о соответствующем курсе действий на основе полученных результатов. Любое действие, предпринятое для решения проблем качества, имеет компромиссы, поэтому вам нужно будет привлечь к процессу принятия решений другие заинтересованные стороны. Наиболее распространенными действиями, предпринимаемыми в результате мероприятий по обеспечению качества, являются доработка, корректировка процесса и принятие.

Доработка

Доработка — это любое действие, которое предпринимается в результате качественных действий по исправлению дефекта. Тест модуля может привести к перезаписи части кода.

Доработка звучит как идеальное решение любой найденной проблемы качества. В мире без временных и бюджетных ограничений это было бы верно, но доработка часто влияет как на график проекта, так и на бюджет. Время достижения результата будет больше, чем предполагалось, поскольку возникнут дополнительные затраты времени для устранения проблемы, а кроме того, люди, выполняющие переработку, потребуют дополнительной оплаты. Могут быть и исключения из увеличения оплачиваемых часов, если в контракте прописаны соответствующие стандарты качества. Тем не менее всегда нужно учитывать возможные финансовые последствия.

Решение о доработке часто связано с серьезностью дефекта и его влиянием на способность конечного пользователя использовать продукт. Ваш клиент, спонсор и другие заинтересованные стороны должны быть вовлечены в принятие решений по доработке.

Корректировки процесса

Изменение процесса может вызвать цепную реакцию, распространяющуюся на всю остальную часть проекта. Если нет уверенности, что изменение процесса относится к небольшой рабочей группе или нескольким членам команды, лучше всего использовать процесс управления изменениями для анализа последствий и получения официального одобрения перед внесением каких-либо изменений.

Принятие

Принятие — это решение принять любые дефекты, обнаруженные в результате тестирования качества. Это общее действие, основанное на анализе серьезности и частоты дефектов, обнаруженных в ходе тестирования. Например, некоторые коммерчески доступные программные продукты выпускаются для продажи широкой публике с известными дефектами, которые впоследствии будут исправлены с помощью обновления. Соблюдение объявленной даты выпуска продукта важнее, чем устранение дефектов; другими словами, график имеет приоритет перед качеством. Общее воздействие принятия дефекта должно быть проанализировано и доведено до сведения заинтересованных сторон проекта. Вам нужно получить их согласие, чтобы принять дефекты.

4.8. Качество документации

ИТ-проекты часто создают документацию, как техническую, так и нетехническую. Действия по контролю качества включают обеспечение качества всей документации, связанной с проектом, и других материалов, переданных группам, которые используют или управляют вашим результатом. Это включает в себя документацию для пользователей, обучение пользователей, обучение в службе поддержки и документацию для других групп поддержки.

Документация пользователя

Любая документация, будь то онлайн или бумажная копия, которая требуется пользователю для работы с системой, должна быть вычитана для проверки орфографии, грамматики и содержания. Все инструкции, касающиеся использования системы,

должны быть проверены для подтверждения точности. Бумажные копии должны быть подготовлены и переплетены для распространения среди пользователей. Документация на экране справки должна быть полной и тщательной. Нет ничего более раздражающего, чем нажать F1 для получения справки только для того, чтобы найти «Никакой помощи по этой теме» или крайне минимальную информацию.

Обучение пользователей

Обучение пользователей возможно под руководством инструктора, в виде самостоятельной работы с книгой или в режиме онлайн. Независимо от средства доставки, учебные планы должны быть проверены на предмет содержания и полноты. Пилотный класс может быть проведен для подтверждения того, что обучение является тщательным и понятным и что любые функции онлайн-практики базы данных обучения работают правильно. Тренеры, которые будут обеспечивать или поддерживать обучение, сами должны быть обучены новой системе и готовы к работе, а любая учебная документация должна быть напечатана, переплетена и готова к занятиям. Учебные материалы, такие как демонстрации и визуальные эффекты, должны быть готовы. Планирование учебных занятий должно быть частью расписания проекта, чтобы гарантировать, что пользователи будут готовы к развертыванию.

Обучение в службе поддержки

Менеджеры ИТ-проектов часто хорошо относятся к обучению конечных пользователей и подготовке классов для пользователей, но они могут упустить из виду специалиста службы поддержки, который в конечном итоге будет получать звонки от пользователей о том, как работает новая система. Необходимо принять во внимание, где служба поддержки будет соответствовать новой системе, и заранее подготовить техников службы с обучением, которое им потребуется для консультирования пользователей. Обучение в службе поддержки должно включать все аспекты обучения конечных пользователей, а также более подробные аспекты решения проблем и устранения неполадок. Все материалы, обсуждаемые в связи с качеством обучения пользователей, также применимы к обучению в службе поддержки. Особенно на ранних этапах выпуска новой системы у пользова-

телей будут возникать вопросы о том, как работает система, и они могут обратиться за ответами в службу поддержки. Служба поддержки должна быть готова к работе во время развертывания.

Иная документация группы поддержки

Получение качественных результатов по ИТ-проектам выходит далеко за рамки выполнения требований заказчика. Другие участники системы должны знать, как эти результаты повлияют на их повседневную деловую деятельность. Поэтому важно принять во внимание, задолго до закрытия проекта, документацию, которая применяется к другим группам в общей структуре поддержки. Общий успех проекта требует точной и полезной документации для всех групп, которые играют определенную роль в эксплуатации и обслуживании вашей системы.

Администраторы серверов должны быть осведомлены о влиянии, которое новая система окажет на серверы. Неразумно устанавливать новое серверное программное обеспечение и не передавать информацию о нем администраторам серверов.

Специалисты по ПК также могут быть осведомлены о любом влиянии новой системы на клиентский компьютер. Например, если ваше новое приложение использует Oracle forms или интерфейс Visual Basic или если приложение с тонким клиентом требует загрузки клиента Java, на клиентских компьютерах могут возникнуть проблемы. Важно, чтобы специалисты по ПК знали, как новая система влияет на клиентские компьютеры, чтобы поддерживать конечного пользователя.

Администраторы баз данных также должны быть осведомлены об изменениях в среде корпоративных баз данных. Особенно важна информация об индексах, связях, триггерах, хранимых процедурах, макетах таблиц, именах столбцов и другой информации, относящейся к системным базам данных. Неразумно размещать новую систему, не обновляя базы данных в соответствии с ее требованиями.

Новые системы, использующие инфраструктуру телефонии или интернета (маршрутизаторы, WAN-каналы и т. д.), требуют помощи от людей, которые являются экспертами в этих областях. Важно, чтобы у них была документация о том, как новая система будет работать на их оборудовании.

Выводы по главе 4

Процесс управления рисками состоит из процессов идентификации рисков, количественной и качественной оценки рисков (анализа), разработки мер реагирования на риски и контроля рисков.

Идентификация рисков — это процесс понимания того, какие потенциальные неудовлетворительные результаты связаны с конкретным проектом, и документирования их характеристик.

Существует два основных типа рисков: известные и предсказуемые (прогнозируемые). Известные риски — это те, которые могут быть выявлены после тщательной оценки плана проекта, бизнес-среды и технической среды, в которой осуществляется проект. Прогнозируемые риски экстраполируются из прошлого опыта проекта. Могут также возникнуть непредсказуемые риски, но их чрезвычайно трудно определить заранее.

Количественная оценка (анализ) рисков позволяет определить возможное смещение сроков проекта.

Оценка риска — это процесс определения вероятности и воздействия выявленного риска. Существует три области рисков, которые могут повлиять на успех проекта: риски проекта, технические риски и бизнес-риски.

Общие источники риска — миссия и цели, факторы принятия решений, управление организацией, клиент или конечный пользователь, бюджет, расписание и т. д.

Семь принципов, обеспечивающих эффективное управление рисками, включают: глобальную перспективу, перспективный взгляд, открытую коммуникацию, интегрированное управление, непрерывный процесс, общее видение продукта и командную работу.

Основными инструментами, которые используются в контроле качества, являются:

- 1) инспекция (тестирование), в том числе:
 - тестирование отдельных модулей;
 - модульное тестирование;
 - системное тестирование;
 - приемочное тестирование пользователей (UAT);

- заводское приемочное тестирование (FAT);
- приемочные испытания на месте;
- 2) диаграмма Парето;
- 3) контрольная диаграмма;
- 4) статистическая выборка;
- 5) блок-схема;
- 6) анализ тенденций.

Вопросы для самоконтроля

1. Дайте определение риска.
2. Что такое управление рисками?
3. Какие основные этапы включает управление рисками?
4. Какие разновидности рисков существуют?
5. Какие способы идентификации рисков вы можете назвать? В чем суть каждого из способов?
6. Что такое качественная оценка рисков? В чем ее преимущества и недостатки?
7. Что такое количественная оценка рисков? В чем ее преимущества и недостатки?
8. Какие существуют методы и способы количественной оценки рисков?
9. Какие существуют методы и способы качественной оценки рисков?
10. Назовите основные методы и инструменты контроля качества ИТ-проекта.
11. Охарактеризуйте инструмент «тестирование отдельных модулей».
12. Охарактеризуйте инструмент «модульное тестирование».
13. Охарактеризуйте инструмент «системное тестирование».
14. Охарактеризуйте инструмент «приемочное тестирование пользователей».
15. Охарактеризуйте инструмент «заводское приемочное тестирование».
16. Охарактеризуйте инструмент «приемочные испытания на месте».
17. Охарактеризуйте инструмент «диаграмма Парето».
18. Охарактеризуйте инструмент «контрольная диаграмма».
19. Охарактеризуйте инструмент «статистическая выборка».
20. Охарактеризуйте инструмент «блок-схема».
21. Охарактеризуйте инструмент «анализ тенденций».

Глава 5

Основные аспекты ИТ-разработки корпоративной информационной системы промышленного предприятия

5.1. Основные требования к корпоративной информационной системе промышленного предприятия

Корпоративная информационная система (КИС) рассматривается в виде совокупности программных модулей, каждый из которых осуществляет управление конкретной сферой деятельностью организации. Программные модули также включают концепции и методологии планирования и управления в различных сферах деятельности организации. Выбор состава модулей в рамках КИС зависит от размеров организации, ее технологических особенностей, сферы ее деятельности, особенностей сбыта продукции и т. д.

При создании КИС промышленного предприятия наиболее предпочтительной концепцией является ERP, так как позволяет планировать все виды ресурсов. КИС промышленного предприятия, разрабатываемая в соответствии с концепцией ERP, должна состоять из следующих функциональных модулей [22]:

- 1) программный модуль управления производственными запасами;

- 2) программный модуль управления планово-аналитической деятельностью;
- 3) программный модуль управления учетной деятельностью;
- 4) программный модуль управления производством;
- 5) программный модуль управления сбытом готовой продукции.

Управление производственными запасами подразумевает формирование оптимального объема запасов предприятия, достаточного для обеспечения процесса производства в объеме, соответствующем потребностям рынка. Поскольку излишки производственных запасов «замораживают» оборотный капитал организации, а также увеличивают затраты на хранение, объем запасов должен быть всегда регламентированным в соответствии с уровнем спроса за прогнозируемый период.

Также следует учитывать, что в связи с сезонными колебаниями уровня цен на некоторые группы материалов в целях снижения общих затрат целесообразно проводить закупки товара в периоды, когда цена является наименьшей. При этом неотъемлемым условием формирования дополнительных запасов является:

$$\Delta_i = C_i + E_i,$$

где Δ_i — изменение цены единицы i -го материала за счет фактора сезонности; C_i — стоимость хранения единицы i -го материала за период хранения; E_i — альтернативные издержки, прибыль, которая могла быть получена при использовании суммы, затраченной на приобретение единицы i -го товара, в прочих видах деятельности предприятия.

Потребность организации в тех или иных запасах рассчитывается на основе анализа спроса. Также на основе объема спроса и нормативных данных расхода материалов на производство единицы изделия рассчитываются объемы запасов всех видов материалов, и полученные результаты заносятся в БД в соответствии с номенклатурой материалов.

Номенклатура требуемых материалов составляется в соответствии с потребностями организации.

Далее составляется план проведения закупок материалов (автоматический вывод формы). По мере проведения закупок фактические объемы каждого из видов материалов сравниваются с запланированными, при этом данные об объемах заносятся в БД.

При проведении закупок одного вида товаров несколькими партиями данные об объеме запаса суммируются, и при превышении фактического объема над запланированным выдается сообщение о превышении объема запаса и сумме дополнительных затрат, связанных с хранением материалов и «заморозкой» капитала.

Модуль управления производственными запасами может быть представлен в виде следующей схемы (рис. 34).

Основными функциями *планово-аналитического модуля* являются:

1) выявление данных о прогнозируемом объеме продаж на основе:

- данных об объемах продаж за предыдущие периоды;
- тенденций развития рынка (к понижению или повышению спроса на данную продукцию);

2) анализ хозяйственной деятельности;

3) составление плана на отчетный период в соответствии со стратегическими целями и проведение анализа хозяйственной деятельности, включая:

- планирование производственных запасов;
- планирование фонда заработной платы;
- планирование численности персонала и пр.

Прогнозирование объема продаж включает следующие этапы:

1) формирование запросов данных за предшествующие периоды к БД;

2) формализация обработки запрашиваемых данных в соответствии с принятыми на предприятии статистическими и математическими методами;

3) вывод данных в виде стандартизованных форм и представление обработанных данных для использования в прочих видах деятельности предприятия.



Рис. 34. Схема модуля управления производственными запасами:

- 1 — данные за предыдущий период; 2 — результаты анализа хозяйственной деятельности; результаты прогноза объема спроса; 3 — результаты анализа хозяйственной деятельности; результаты прогноза объема спроса; нормы расхода материалов на производство единицы продукции; данные о материалах, цены на которые подвержены сезонным колебаниям; 4, 5 — результаты планирования производственных запасов; 6 — фактические данные о формировании производственных запасов; 7 — данные о дополнительных затратах

Модуль управления планово-аналитической деятельностью представлен в виде схемы (рис. 35).

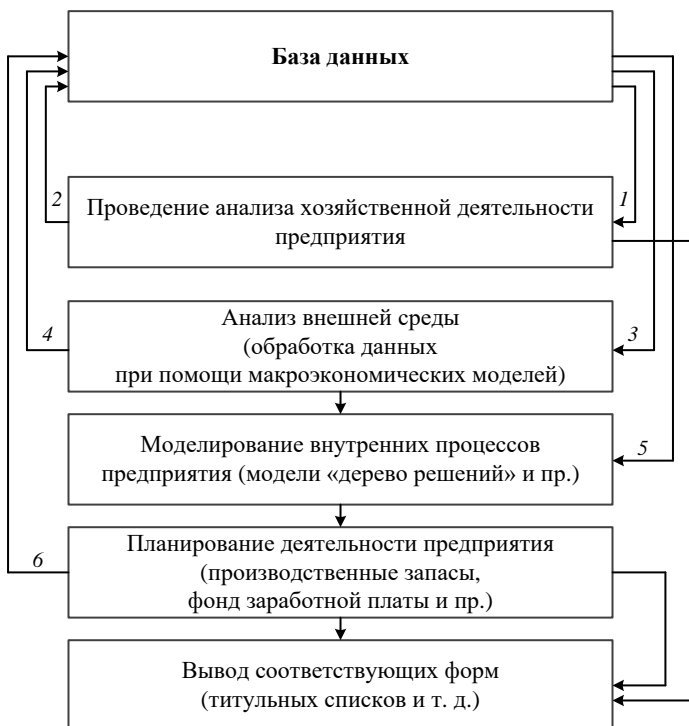


Рис. 35. Схема модуля управления
планово-аналитической деятельностью:

- 1 — данные за предыдущий период деятельности;
- 2, 5 — результаты проведения анализа хозяйственной деятельности;
- 3 — данные о состояниях внешней среды предприятия за предыдущие периоды;
- 4 — результаты проведения анализа внешней среды предприятия;
- 6 — результаты планирования

Разработка модуля управления учетной деятельностью подразумевает создание системы, позволяющей пользователям вводить данные в БД, а затем, на основе автоматических расчетов требуемых величин, получать формы первичной документации, а также отчетные формы.

Модуль управления производством представлен в виде схемы (рис. 36).

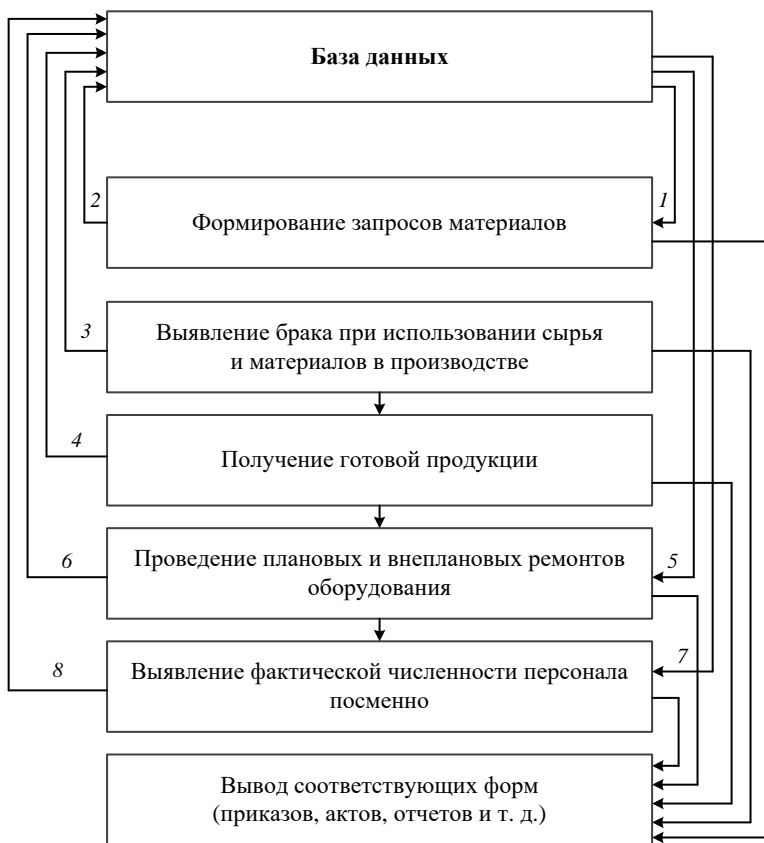


Рис. 36. Схема модуля управления производством:

- 1 — данные об объеме производственных запасов;
- 2 — данные об объеме материалов необходимых в производстве;
- 3 — данные о видах и объеме бракованных материалов;
- 4 — данные о полученной готовой продукции;
- 5 — данные о ранее проводимых ремонтах оборудования;
- 6 — данные о текущих проводимых ремонтах оборудования;
- 7 — данные о численности персонала;
- 8 — данные о фактической численности персонала посменно

Модуль управления сбытом готовой продукции. При осуществлении сбыта продукции наряду с маркетинговой деятельностью осуществляется выработка плана поставок на основе

данных о потребителях продукции предприятия и данных о наличии готовой продукции.

Схема модуля управления сбытом готовой продукции показана на рис 37.

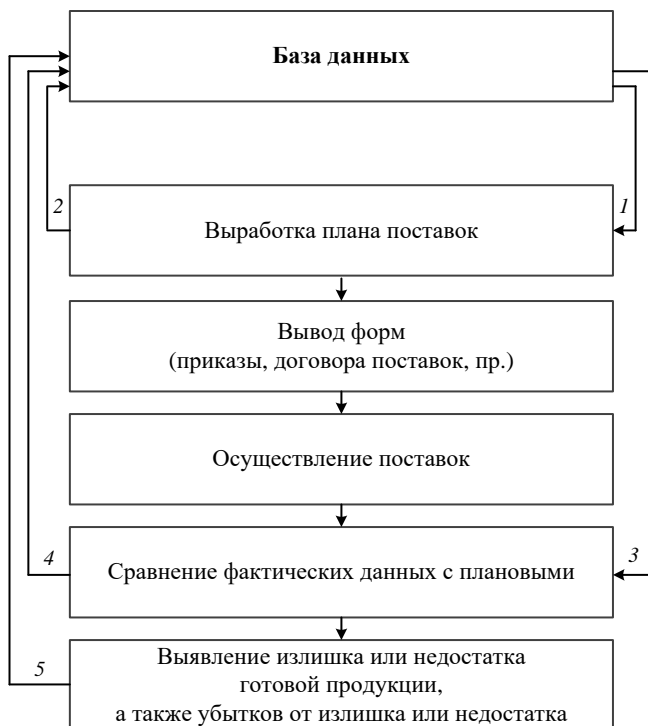


Рис. 37. Схема модуля управления сбытом готовой продукции:

- 1 — данные о потребителях продукции предприятия и о готовой продукции;
- 2, 3 — результаты планирования поставок;
- 4 — фактические данные о проведении поставок;
- 5 — данные о излишке / недостатке готовой продукции и связанных с ним убытках

При помощи программного модуля управления учетной деятельностью осуществляется структурирование, хранение, редактирование и интерпретация данных различных сфер деятельности предприятия в соответствии со схемами на рис. 34–37.

Всеми программными модулями КИС промышленного предприятия в совокупности выполняются следующие функции:

1) планирование продаж и производства. Результатом является разработка плана производства основных видов продукции;

2) управление спросом. Данный модуль предназначен для прогноза будущего спроса на продукцию, определения объема заказов, которые можно предложить клиенту, определения спроса дистрибьюторов и др.;

3) укрупненное планирование мощностей. Используется для конкретизации планов производства и определения степени их выполнимости;

4) формирование основного плана производства (план-график выпуска продукции). Определяется продукция в конечных единицах (изделиях) со сроками изготовления и количеством;

5) планирование потребностей в материалах. Определяются виды материальных ресурсов (сборных узлов, готовых агрегатов, покупных изделий, исходного сырья, полуфабрикатов и др.) и конкретные сроки их поставки для выполнения плана;

6) спецификация изделий. Определяет состав конечного изделия, материальные ресурсы, требуемые для его изготовления, и др. Фактически спецификация является связующим звеном между основным планом производства и планом потребностей в материалах;

7) планирование потребностей в мощностях. На данном этапе планирования более детально, чем на предыдущих уровнях, определяются производственные мощности;

8) маршрутизация/определение рабочих центров. Конкретизируются как производственные мощности различного уровня, так и маршруты, в соответствии с которыми выпускаются изделия;

9) проверка и корректировка цеховых планов по мощностям;

10) управление закупками, запасами, продажами;

11) управление финансами (ведение Главной книги, расчеты с дебиторами и кредиторами, учет основных средств, управление наличными средствами, планирование финансовой деятельности и др.);

12) управление затратами (учет всех затрат предприятия и калькуляция себестоимости готовой продукции или услуг).

В общем виде модель КИС промышленного предприятия (а также любой из программных модулей КИС) может быть представлена в виде следующей схемы (рис. 38).



Рис. 38. Схема КИС (программных модулей КИС) промышленного предприятия:

- 1 — обращение к программам управления данными (запуск программ);
- 2 — запросы данных, проведение выборки данных, внесение данных и пр.;
- 3 — данные; 4 — данные, результаты выборки, результаты;
- 5 — обращение к программам расчета величин и показателей на основе полученных данных (запуск программ);
- 6 — результаты проведения анализа; 7 — представление результатов расчетов в виде форм, результаты; 8 — формы

Далее приведено описание компонентов процесса обработки данных в рамках КИС промышленного предприятия [22].

Ядро системы (базы данных). В файлах структуры БД фиксируются структура хранения данных и свойства обработки для каждой единицы данных (для каждого поля данных). В фай-

лах структуры БД могут быть отражены следующие составляющие системы:

- система критериев оценки конкурентных преимуществ и состояния внешней среды;
- показатели и индикаторы, на основе которых проводится анализ и расчеты;
- система показателей, используемая в производственных процессах предприятия (нормативы, коэффициенты, шифры, ставки и т. д.);
- внутренние показатели организации (сведения о сотрудниках, произведенной продукции, данные о затратах и пр.);
- результаты вычислений, проведенных в процессе принятия решений на основе внутренних стандартов, показателей и индикаторов;
- пр.

Программы и алгоритмы взаимодействия с ядром системы. Основное назначение — осуществление управления БД. Все программы и алгоритмы можно подразделить на следующие группы:

- программы первичной обработки данных и ввода данных в БД (в том числе программы проверки вводимых данных на соответствие);
- программы, осуществляющие чтение данных в БД в заданной последовательности;
- программы, осуществляющие выборку данных из БД по заданным параметрам (критериям);
- программы интерпретации показателей и индикаторов;
- программы расчета значений на основе запрашиваемых данных.

Каждая программа и алгоритм представляет собой программный код. В целях избежания ошибок при чтении и вставке данных необходимо создание программных кодов всех видов запросов к БД и их последующая стандартизация, т. е. установление внутриорганизационных стандартов для процедур получения и вставки данных с учетом периодов времени, за которые запрашиваются (вставляются) данные, типов данных и пр. При этом отпадает необходимость написания программы-запроса при создании пользовательских приложений, вместо чего вызы-

вается уже стандартизированная программа, которая предоставляет обработанные данные в окно диалога.

Пользовательские приложения. Назначение пользовательских приложений в рамках КИС промышленного предприятия — управление данными. Пользовательские приложения дают сотрудникам предприятия возможность:

1) получения данных и результатов выборки за различные периоды времени. Периоды времени задаются вводом соответствующего параметра, который передается программе, исполняющей выборку в соответствии со значениями параметра;

2) ввода данных для внесения в БД и дальнейшей обработки. В этом случае приложение должно предоставлять пользователю для заполнения форму, содержащую поля ввода данных. По окончании заполнения пользователем формы вызывается исполняющая программа, которая обеспечивает ввод данных в БД. В целях избежания ошибок ввода рекомендуется предварительная проверка исполняющей программой введенных данных на соответствие требованиям к формату данных перед осуществлением процедуры ввода данных в БД;

3) редактирования ранее введенных данных. Как правило, подразумевается такая последовательность: чтение (выборка) данных из БД; обработка полученных данных в окне пользовательского приложения; вставка данных в БД. Эта процедура представляет собой совокупность двух вышеперечисленных;

4) удаление ранее введенных данных. В целях обеспечения возможности восстановления данных в случае непроизвольного их удаления пользователями рекомендуется следующая организация процедуры удаления:

- создание однобайтового поля индикатора активности учетной записи при проектировании структуры БД (таблица FDT);

- чтение данных последовательно, по каждой учетной записи, подлежащей удалению, и вывод данных в окне пользовательского приложения;

- присвоение полю индикатора активности каждой удаляемой учетной записи значения, соответствующего неактивному статусу учетной записи по команде пользователя, исходящей из приложения;

- 5) вставка данных в БД;
- 6) редактирование текста внутри окна пользовательского приложения;
- 7) отображение полученных данных в единой форме и возможность манипулирование данными, а также возможность вызова программ дальнейшего расчета для показателей, указанных посредством окна приложения;
- 8) отображение и печать форм, сформированных после проведения основных расчетов.

Расчетные программы и алгоритмы. Расчетные программы и алгоритмы осуществляют основные расчеты и преобразования и образуют модульный состав КИС промышленного предприятия. Математические и статистические методы, оптимизационные модели, модель «дерево решений» и пр. могут реализовываться в рамках программных модулей программистами самого предприятия, а также приобретаться у разработчиков программных и расчетных модулей (в том числе интегрированные пакеты SAS). Рассчитанные величины при необходимости окончательно интерпретируются в соответствии с принятыми на предприятии стандартами и передаются в формы.

Формы. Все результаты расчетов должны отображаться в виде стандартных форм (соответственно ГОСТам либо внутренним стандартам в зависимости от назначения).

5.2. Требования к среде проектирования приложений

Чтобы быть способной реализовать программное и алгоритмическое наполнение программных модулей КИС промышленного предприятия, а также осуществлять настройки программных модулей, среда разработки приложений должна обладать следующими возможностями [22]:

- 1) событийное программирование. Подразумевает выполнение определенной процедуры или подпрограммы на языке программирования среды разработки при совершении пользователем определенного события, т. е. действия с активными компонентами пользовательских приложений;

2) функциональность расчетных операторов языка программирования среды разработки;

3) управление параметрами. Должна быть предусмотрена возможность исполнения различных программ на основе задаваемых параметров, а также процедура ввода и изменения значений параметров;

4) иерархия приложений (приоритеты исполнения приложений и программ). Необходимо наличие средств назначения приоритетов перехода от приложения к приложению и порядка исполнения процедур различных приложений;

5) возможность использования программ и запросов несколькими пользовательскими приложениями. В целях оптимизации расчетов должна обеспечиваться возможность вызова или исполнения каждой программы взаимодействия с БД, а также расчетной программы из любого пользовательского приложения;

6) стандартизация запросов и программ обработки. Для обеспечения доступа приложений к программам необходимо каждый тип запроса к БД или расчета данных сохранить в виде стандартизированного файла.

5.3. Графический интерфейс и обработка отчетных форм

В целях представления данных в рамках пользовательских приложений КИС в наиболее понятном и удобном для конечного пользователя виде приложения необходимо оформлять максимально иллюстрированно. Также графический интерфейс служит для упрощения процедуры ввода данных (посредством использования полей со списком, полей выбора и пр. для типов заносимых данных, количество значений которых сравнительно невелико).

Ниже представлен перечень необходимых компонентов графического интерфейса:

- область управления;
- поле ввода;
- групповой фрейм;

- список;
- кнопка;
- ящик выбора;
- надпись;
- кнопка переключателя.

При проведении основных расчетов, анализа и интерпретации данных модулями КИС промышленного предприятия результаты оформляются и представляются в виде форм ведомостей и других документов. Таким образом, согласно реализуемой модели КИС компиляция форм должна осуществляться по запросу в рамках пользовательского приложения, с последующим возвратом полученных отчетных форм в окно приложения для дальнейшей проверки и распечатки.

5.4. Функционирование КИС в условиях распределенной среды

Под распределенной базой данных (Distributed DataBase — DDB) обычно подразумевают базу данных, включающую фрагменты из нескольких баз данных, которые располагаются на различных узлах сети компьютеров. Для функционирования в условиях распределенной среды СУБД КИС промышленного предприятия должна соответствовать следующим 12 условиям [22]:

1) локальная автономия. Управление данными на каждом из узлов распределенной системы выполняется локально. База данных, расположенная на одном из узлов, является неотъемлемым компонентом распределенной системы. Будучи фрагментом общего пространства данных, она в то же время функционирует как полноценная локальная база данных. Управление базой данных осуществляется локально и независимо от других узлов системы;

2) независимость от центрального узла. В идеальной системе все узлы равноправны и независимы, а расположенные на них базы являются равноправными поставщиками данных в единое информационное пространство. База данных на каждом из узлов самодостаточна, т. е. включает полный собственный сло-

варь данных и полностью защищена от несанкционированного доступа;

3) непрерывные операции. Возможность непрерывного доступа к данным в рамках распределенной системы вне зависимости от их расположения и операций, выполняемых на локальных узлах;

4) прозрачность расположения. Полная прозрачность расположения данных. Пользователь, обращающийся к распределенной БД, ничего не должен знать о реальном, физическом размещении данных в узлах информационной системы. Все операции над данными выполняются без учета их местонахождения. Транспортировка запросов к базам данных осуществляется встроенными системными средствами;

5) прозрачная фрагментация. Возможность распределенного размещения данных, логически представляющих собой единое целое. Существует фрагментация двух типов: горизонтальная (хранение строк одной логической таблицы в нескольких идентичных физических таблицах на различных узлах) и вертикальная (распределение столбцов логической таблицы по нескольким узлам). Следующий пример иллюстрирует оба типа фрагментации. Имеется таблица `employee` (`emp_id`, `emp_name`, `phone`), определенная в базе данных на узле в городе А (`city_A`). Имеется точно такая же таблица, определенная в базе данных на узле в городе В (`city_B`). Обе таблицы хранят информацию о сотрудниках компании. Кроме того, в базе данных на узле в Далласе определена таблица `emp_salary` (`emp_id`, `salary`). Тогда запрос «получить информацию о сотрудниках компании» может быть сформулирован так:

```
READ ALL IN employee@city_A, employee@city_B ORDER BY emp_id
```

6) прозрачность тиражирования. Тиражирование данных — это асинхронный процесс переноса изменений объектов исходной базы данных в базы, расположенные на других узлах распределенной системы. В данном контексте прозрачность тиражирования означает возможность переноса изменений между базами данных средствами, невидимыми пользователю распределенной системы;

7) обработка распределенных запросов. Возможность выполнения операций выборки над распределенной базой данных, сформулированных в рамках обычного запроса. Иными словами, операцию выборки можно сформулировать с помощью тех же языковых средств, что и операцию над локальной базой данных. Например,

```
FIND customer.name, customer.address, order.number, order.date  
IN customer@london, order@paris WHERE customer.cust_number =  
order.cust_number
```

8) обработка распределенных транзакций. Возможность выполнения операций обновления распределенной базы данных (STORE, UPDATE, DELETE), не разрушающая целостность и согласованность данных. Эта цель достигается применением двухфазового протокола фиксации транзакций (two-phase commit protocol), ставшего фактическим стандартом обработки распределенных транзакций. Его применение гарантирует согласованное изменение данных на нескольких узлах в рамках распределенной (или, как ее еще называют, глобальной) транзакции;

9) независимость от оборудования. В качестве узлов распределенной системы могут выступать компьютеры любых моделей и производителей (от мэйнфреймов до персональных компьютеров);

10) независимость от операционных систем. Многообразие операционных систем, управляющих узлами распределенной системы;

11) прозрачность сети. Доступ к любым базам данных может осуществляться по сети. Спектр поддерживаемых конкретной СУБД сетевых протоколов не должен быть ограничением системы с распределенными базами данных, т. е. в распределенной системе обращение к БД может производиться по любым сетевым протоколам;

12) независимость от СУБД. В распределенной системе могут сосуществовать СУБД различных производителей, возможны операции поиска и обновления в базах данных различных моделей и форматов.

5.5. Стоимость ПО среды проектирования и эксплуатации

Помимо технических и структурных характеристик СУБД и редакторов приложений, большое значение при принятии решения о приобретении некоторых программных продуктов для построения КИС промышленного предприятия имеет оценка повышения уровня производительности после внедрения, экономического эффекта от внедрения, срока окупаемости инновации и т. д. Таким образом, при разработке и использовании автоматизированной системы управления встает проблема оценки стоимости ее использования.

Выводы по главе 5

КИС промышленного предприятия, разрабатываемая в соответствии с концепцией ERP, должна состоять из следующих функциональных модулей:

- 1) программный модуль управления производственными запасами;
- 2) программный модуль управления планово-аналитической деятельностью;
- 3) программный модуль управления учетной деятельностью;
- 4) программный модуль управления производством;
- 5) программный модуль управления сбытом готовой продукции.

Основными компонентами КИС промышленного предприятия являются:

- базы данных;
- программы и алгоритмы взаимодействия с базами данных;
- пользовательские приложения;
- расчетные программы и алгоритмы;
- отчетные формы.

Среда разработки приложений должна обладать следующими возможностями:

- 1) событийное программирование;
- 2) функциональность расчетных операторов языка программирования среды разработки;
- 3) управление параметрами;
- 4) иерархия приложений;
- 5) возможность использования программ и запросов несколькими пользовательскими приложениями;
- 6) стандартизация запросов и программ обработки.

Вопросы для самоконтроля

1. Какие функциональные модули должна включать корпоративная информационная система промышленного предприятия?
2. Опишите состав и функциональное назначение программного модуля управления производством.
3. Опишите состав и функциональное назначение программного модуля управления производственными запасами.
4. Опишите состав и функциональное назначение программного модуля управления сбытом готовой продукции.
5. Опишите состав и функциональное назначение программного модуля управления учетной деятельностью.
6. Опишите состав и функциональное назначение программного модуля управления планово-аналитической деятельностью.
7. Назовите основные компоненты КИС промышленного предприятия.
8. Каково назначение базы данных в КИС промышленного предприятия?
9. Каково назначение программ и алгоритмов взаимодействия с базами данных?
10. Каково назначение пользовательских приложений?
11. Каково назначение расчетных программ и алгоритмов?
12. Каково назначение отчетных форм?
13. Назовите основные элементы интерфейса программных приложений.
14. Назовите 12 признаков распределенной среды промышленного предприятия.

Практическая работа

Задание

1. Представить формальное определение методологии, метода и стандарта. Подобрать примеры методологий, методов и стандартов в разрезе предметных областей.

2. Выбрать ИТ-проект, выделить группы участников проекта, произвести предварительный анализ воздействия производственной группы на результаты проекта. Далее, определив степень участия каждой из этих групп в проекте, предложить меры по сотрудничеству с ними.

3. Построить диаграмму Ганта в MS Excel по следующим условиям (самостоятельно спланировать начало и конец каждого этапа).

Этап	Задержка, дн.	Начало этапа	Длительность, дн.	Конец этапа	Выполнение, %
1. Переговоры	0		4		50
2. Оформление договора	1		2		100
3. Распределение заказов	0		7		50
4. Набор персонала	–2		8		30
5. Аренда оборудования	0		5		100
6. Аренда площадей	–1		3		70
7. Складские работы	0		7		60
8. Завоз товара на точки продаж	–3		9		80
9. Рекламная кампания	0		3		30
10. Подведение итогов	5		6		90
11. Банкет	1		20		40

Заключение

Управление ИТ-разработкой и внедрением представляет собой комплексную задачу, требующую вовлечения обширных финансовых, интеллектуальных и прочих ресурсов. Согласно исследованиям 97 % организаций считают, что управление разработкой и проектами имеет решающее значение для эффективности бизнеса. Грамотное управление проектами гарантирует, что ни один ресурс не останется вне цикла.

Обеспечение хорошей коммуникации и отношений между членами команды повышает производительность, а также приводит к позитивному культурному сдвигу в организации. Управление ИТ-разработкой учитывает нюансы взаимоотношений с каждой из заинтересованных сторон, что позволяет держать проекты под контролем в дружественной и благоприятной для всех атмосфере. У организации всегда должен быть план действий на случай непредвиденных обстоятельств. Управление рисками включает определение критических элементов успеха проекта и любых факторов, которые могут повлиять на них. Подход к проекту, его жизненный цикл и процессы управления должны адаптироваться и корректироваться в соответствии с требованиями проекта и продукта.

Для дальнейшего совершенствования навыков управления ИТ-разработкой авторы пособия рекомендуют более детально изучить автоматизированное управление ИТ-проектами. Некоторые издания приведены в списке рекомендуемой литературы.

Библиографический список

Использованная литература

1. *Project Life Cycle: phases and characteristics*. URL: <https://twoproject.com/blog/project-life-cycle-phases-and-characteristics/>.
2. *Project Scheduling Techniques / Various Techniques of Project Scheduling*. URL: <https://www.educba.com/project-scheduling-techniques/>.
3. *SDLC (Software Development Life Cycle) Phases, Process, Models*. URL: <https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>.
4. *Project evaluation methods – CEOpedia / Management online*. URL: https://ceopedia.org/index.php/Project_evaluation_methods.
5. *Project Management Institute. A guide to the project management body of knowledge (PMBOK guide)*. 6th ed. Newtown Square, PA. 976 p.
6. *Project Integration Management / How to Integrate Processes / Edureka*. URL: <https://www.edureka.co/blog/project-integration-management>.
7. *IT Project Management – Planning Phase*. URL: <https://www.visual-paradigm.com/tour/project-management/it-project-management-lifecycle-planning.jsp>.
8. *Galoppin L., Caems S. Managing Organizational Change during SAP Implementations*. Bonn, Germany: SAP Press, 2007. 365 p.
9. *Decomposition (computer science) / Wikipedia*. URL: [https://en.wikipedia.org/wiki/Decomposition_\(computer_science\)](https://en.wikipedia.org/wiki/Decomposition_(computer_science)).
10. *What is Precedence Diagramming Method in Project Management?* URL: <https://www.invensislearning.com/articles/pmp/precedence-diagramming-method>.
11. *Clark W., Gantt H. The Gantt chart, a working tool of management*. New York: The Ronald Press Company, 1922.
12. *Simple Gantt Chart Examples in Project Management*. URL: <https://blog.ganttpro.com/en/gantt-chart-example/>.
13. *What is Critical Path Method For in Project Management? / Hygger.io*. URL: <https://hygger.io/blog/what-is-critical-path-method-for-in-project-management/>.
14. *Beginner's Guide to Critical Path Method (CPM) / Smartsheet*. URL: <https://www.smartsheet.com/critical-path-method/>.

15. Милошевич Д. З. Набор инструментов для управления проектами / пер. с англ. Е. В. Мамонтова; под ред. С. И. Незвестного. М.: Академия АйТи: ДМК Пресс, 2008. 729 с.

16. CAPM Certification / Online CAPM Training Course. URL: <https://www.simplilearn.com/project-human-resource-management-tutorial/>.

17. IT Team. Types of IT Team. Try Freshservice ITSM Software. URL: <https://freshservice.com/it-team>.

18. Sommerville I. Software Engineering. 6th ed. Jakarta: Penerbit Erlangga, 2003.

19. Pressman R. S. Software Engineering: A Practitioner's Approach. 6th ed. USA: Irwin/McGraw-Hill, 2005.

20. Podziņš O., Romānovs A. IT Risk Identification and Assessment Methodology // Environment. Technology. Resources: Proceedings of the 11th International Scientific and Practical Conference. Rezekne: Rezekne Academy of Technologies, 2017. Vol. II. P. 124–127.

21. Heldman W., Cram L. IT Project + Study Guide (Exam PKO-002). 2nd ed. John Wiley & Sons, 2004. 512 p.

22. Воронов М. П. Методы модификации модульных структур учета продукции в корпоративных информационных системах лесопромышленных предприятий: дис. ... канд. техн. наук: 05.21.01. Екатеринбург, 2006. 157 с.

Рекомендуемая литература

Бедердинова О. И., Водовозова Ю. А. Автоматизированное управление ИТ-проектами: учеб. пособие. М.: ИНФРА-М, 2021. 92 с.

Управление проектами: фундаментальный курс: учебник / А. В. Алешин, В. М. Аньшин, К. А. Багратиони и др.; под ред. В. М. Аньшина, О. Н. Ильиной. М.: Изд. дом Высш. шк. экономики, 2013. 500 с.

Чекмарев А. В. Управление ИТ-проектами и процессами: учеб. для вузов. М.: Юрайт, 2020. 228 с.

Информация об авторах

Часовских Виктор Петрович — профессор кафедры шахматного искусства и компьютерной математики УрГЭУ, доктор технических наук, профессор, e-mail: u2007u@ya.ru

Воронов Михаил Петрович — доцент кафедры шахматного искусства и компьютерной математики УрГЭУ, кандидат технических наук, доцент, e-mail: mstrk@yandex.ru

Лабунец Валерий Григорьевич — профессор кафедры шахматного искусства и компьютерной математики УрГЭУ, доктор технических наук, профессор, e-mail: vlabunets05@yahoo.com

Стариков Евгений Николаевич — заведующий кафедрой шахматного искусства и компьютерной математики УрГЭУ, кандидат экономических наук, доцент, e-mail: starikov_en@usue.ru

Оглавление

Введение	3
Глава 1. Жизненный цикл ИТ-проекта.....	6
1.1. Жизненный цикл проекта: понятие, основные фазы.....	6
1.2. Модели жизненного цикла ИТ-проекта.....	11
1.3. Инициация проекта	20
1.4. Управление интеграцией ИТ-проектов.....	22
1.5. Оценка ИТ-проекта	32
<i>Выводы по главе 1.....</i>	<i>34</i>
<i>Вопросы для самоконтроля</i>	<i>36</i>
Глава 2. Планирование ИТ-проекта	37
2.1. Основные аспекты планирования ИТ-проекта	37
2.2. Определение содержания ИТ-проекта.....	49
2.3. Методы формирования списка операций проекта.....	56
2.4. Метод предшествования — определение взаимосвязи операций	58
<i>Выводы по главе 2.....</i>	<i>64</i>
<i>Вопросы для самоконтроля</i>	<i>66</i>
Глава 3. Управление расписанием и командой ИТ-проекта	67
3.1. Инструменты и методы составления расписания ИТ-проекта	67
3.2. Диаграмма Ганта.....	70
3.3. Метод критического пути	79
3.4. Диаграмма контрольных событий.....	87
3.5. Управление человеческими ресурсами ИТ-проекта.....	89
3.6. ИТ-команды. Требования к составу ИТ-команды.....	96
3.7. Заинтересованные стороны (участники) проекта.....	103
3.8. Управление командой ИТ-проекта	106
<i>Выводы по главе 3.....</i>	<i>115</i>
<i>Вопросы для самоконтроля</i>	<i>117</i>

Глава 4. Управление рисками и финансовое обоснование ИТ-проекта.....	119
4.1. Основные понятия управления рисками ИТ-проекта	119
4.2. Идентификация рисков ИТ-проекта	122
4.3. Качественный и количественный анализ рисков	125
4.4. Меры реагирования на риски	127
4.5. Контроль рисков	129
4.6. Основные методы и инструменты контроля качества ИТ-проекта	133
4.7. Меры по контролю качества ИТ-проекта	139
4.8. Качество документации	144
<i>Выводы по главе 4.....</i>	<i>147</i>
<i>Вопросы для самоконтроля</i>	<i>148</i>
Глава 5. Основные аспекты ИТ-разработки корпоративной информационной системы промышленного предприятия	149
5.1. Основные требования к корпоративной информационной системе промышленного предприятия	149
5.2. Требования к среде проектирования приложений	160
5.3. Графический интерфейс и обработка отчетных форм.....	161
5.4. Функционирование КИС в условиях распределенной среды	162
5.5. Стоимость ПО среды проектирования и эксплуатации	165
<i>Выводы по главе 5.....</i>	<i>165</i>
<i>Вопросы для самоконтроля</i>	<i>166</i>
Практическая работа.....	167
Заключение	168
Библиографический список	169
Информация об авторах	171

Учебное издание

Часовских Виктор Петрович,
Воронов Михаил Петрович,
Лабунец Валерий Григорьевич,
Стариков Евгений Николаевич

УПРАВЛЕНИЕ ИТ-РАЗРАБОТКОЙ И ВНЕДРЕНИЕМ

Учебное пособие

Редактор и корректор *Л. В. Матвеева*
Компьютерная верстка *И. В. Засухиной*

Поз. 50. Подписано в печать 15.11.2021.

Формат 60 × 84 1/16. Бумага офсетная. Печать плоская.

Уч.-изд. л. 7,0. Усл. печ. л. 10,2. Печ. л. 11,0. Заказ 565. Тираж 43 экз.

Издательство Уральского государственного экономического университета
620144, г. Екатеринбург, ул. 8 Марта/Народной Воли, 62/45

Отпечатано с готового оригинал-макета в подразделении оперативной полиграфии
Уральского государственного экономического университета



УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

ISBN 978-5-9656-0318-3



9 785965 603183 >