

5) Руководитель проекта: Кто будет следить за тем, чтобы все, что необходимо завершить, было завершено?

Таким образом, проект — это не что-то рутинное. Повседневные операции или техническое обслуживание не считаются проектом, поскольку у них нет определенного начала и конца.

Тогда что же такое «Управление проектами»?

Управление проектами — это практика применения знаний, навыков, инструментов и методов для завершения проекта в соответствии с конкретными требованиями. Все сводится к выявлению проблемы, созданию плана ее решения, а затем выполнению этого плана до тех пор, пока проблема не будет решена. Это определение может показаться простым, но на каждом этапе процесса в это понятие входит очень многое.

Как ни странно, но корни управления проектами можно проследить еще со времен строительства пирамид в Гизе и Великой Китайской стены — крупномасштабные работы всегда требовали организации и управления.

Современное развитие управления проектами началось в 19 веке, когда железнодорожные компании закупили тонны сырья и наняли тысячи людей для работы на трансконтинентальной железной дороге.

Это особенно актуально при создании высокотехнологических проектов с применением современных методов, средств и инструментов.

Первые «попытки» систематизировать знания в этой области были сформированы к началу 20-го века, когда Фредерик Тейлор применил концепции управления проектами к рабочему дню, разработав стратегии для более эффективной работы и повышения эффективности вместо того, чтобы требовать, чтобы работники работали усерднее и дольше.

Генри Гант, сотрудник Тейлора, взял эти концепции и использовал столбцы и диаграммы для отображения того, когда были выполнены определенные задачи или ряд задач, создавая новый способ визуализации управления проектами.

Во время Второй мировой войны военные и промышленные лидеры использовали еще более подробные стратегии управления, что в итоге привело к более стандартизированным процессам, таким как метод критического пути.

Эти методы набирали популярность во всех отраслях промышленности, и в 1965 и 1969 годах были основаны Международная ассоциация управления проектами и Институт управления проектами соответственно. В 2001 году гибкие методологии управления проектами были кодифицированы путем создания Манифеста Agile (гибкой методологии разработки проектов и решений).

Область управления проектами продолжает меняться по мере усиления конкуренции, необходимости быстрых изменений и появления на рынке новых технологий (автоматизация, искусственный интеллект и т. Д.).

Каковы же основные этапы управления проектами, согласно ретроспективе?

Мы можем разделить процесс управления проектом на четыре группы.

1) Инициирование: Цель этого этапа - определить проект.

- 2) Планирование: Этот этап включает разработку дорожной карты, которой должны следовать все.
- 3) Выполнение и мониторинг: На этом этапе создается проектная группа и создаются результаты. Менеджеры проектов будут отслеживать и измерять эффективность проекта, чтобы убедиться, что он остается на должном уровне.
- 4) Закрытие: Проект завершен, проводится вскрытие, и проект передается другой команде, которая будет его поддерживать.

Почему же важно управление проектами?

Эффективное управление проектами поможет организации:

- 1) иметь более предсказуемый процесс планирования и выполнения проекта;
- 2) придерживаться бюджетов проектов, графиков и руководящих принципов по охвату;
- 3) быстрее и проще устранять препятствия и проблемы проекта;
- 4) выявлять и завершать проекты, которые не имеют соответствующей бизнес-ценности;
- 5) становиться более эффективными в решении задач;
- 6) улучшать сотрудничество между командами и внутри команд;
- 7) выявлять и планировать риски

Исходя из этого, давайте рассмотрим, как грамотный менеджер проекта может помочь развиваться команде и завершать проект вовремя.

Менеджеры проектов в общем случае отвечают за планирование, выполнение, мониторинг, контроль и завершение проектов.

Однако это только верхушка айсберга управления проектами. Вот некоторые из основных обязанностей руководителя проекта, которые определяют эффективность работы и положительный исход.

- 1) Составленный план: Руководители проектов отвечают за разработку наиболее реалистичного курса для проекта. План должен включать объем проекта, сроки и бюджет. Это также может включать определение подходящих инструментов для работы.
- 2) Эффективная и грамотная команда: Определение подходящей команды имеет решающее значение для успеха проекта. Каждая проектная группа будет отличаться в зависимости от масштаба инициативы и функций, необходимых для завершения проекта. Идеально подходит поиск специалистов и экспертов по предмету для каждой из необходимых задач.
- 3) Грамотное распределение задач: Руководители проектов должны предоставить своей команде четкое определение конкретных задач и сроков для каждой части проекта. Хотя каждый член команды будет отвечать за свои собственные задания, многие задачи потребуют сотрудничества как от внутренних, так и от внешних членов команды.
- 4) Эффективное руководство командой: Теперь, когда команда собрана и их задачи распределены, руководитель проекта должен

поддерживать машину в хорошем состоянии. Это будет включать проверку отдельных лиц на предмет обновления статуса, выявление и устранение препятствий, урегулирование разногласий, поддержание высокого морального духа команды, а также обеспечение обучения и наставничества.

- 5) Управление бюджетом: Большинство проектов потребуют определенных затрат, а это означает, что понимание того, как составить бюджет проекта и управлять затратами, имеет решающее значение для успеха. Это будет включать сравнение реальных расходов с оценками и корректировку плана проекта, если это необходимо.
- 6) Управление сроками: Как и в случае с бюджетом, руководителям проектов поручено выполнять все по графику, чтобы команда соблюдала намеченные сроки завершения. Это потребует установления реалистичных крайних сроков на протяжении всего жизненного цикла проекта, постоянного общения со своей командой для получения обновлений статуса и поддержания подробного графика.
- 7) Привлечение заинтересованных сторон: Заинтересованные стороны играют большую роль в вашем проекте. Как правило, это влиятельные люди, на которых влияет проект. Менеджеры проектов должны поддерживать хорошие отношения и открытую линию связи с заинтересованными сторонами, которые могут не только помочь устранить препятствия и расширить возможности вашей команды, но также создать ненужные узкие места и сорвать проект, если они будут недовольны направлением.
- 8) Передача проекта: Просто потому, что цели проекта были достигнуты, это не значит, что работа руководителя проекта закончена. Теперь руководитель проекта должен передать проект команде, которая будет управлять, поддерживать и эксплуатировать его в дальнейшем. На этом этапе менеджер проекта больше не будет «переходить» к человеку и будет назначен на новый проект.
- 9) Документирование процесса: Выявление и документирование «извлеченных уроков» - это не только хорошая практика для личного роста руководителя проекта, но и для передачи этого опыта другим командам по всей организации для использования в будущем. Это поможет другим избежать повторения тех же ошибок.

Теперь озвучим основные принципы управления проектами, согласно современной концепции.

- 1) Имейте четко определенные цели и задачи проекта.

Этот принцип находится на первом месте в нашем списке по уважительной причине. Цели, которые вы ставите перед своим проектом, будут играть решающую роль в его успехе или провале в проектах. Когда вы устанавливаете цели своего проекта до начала работы, вы, ваш клиент и ваша

команда находитесь на одной волне, и в будущем можно избежать недоразумений.

Хорошие цели реалистичны, ясны и поддаются измерению.

Реалистично – Можем ли мы достичь этой цели за отведенное время и доступные нам ресурсы?

Четкий– Знаем ли мы точно, чего от нас требуют? Все ли понимают?

Измеримый– Существуют ли поддающиеся количественной оценке показатели, с помощью которых мы можем судить о каждой цели?

2) Определите свои результаты

Как только цели и задачи проекта будут определены, вы сможете определить результаты вашего проекта. Например, если цель заказчика состоит в том, чтобы конечные пользователи могли управлять своим собственным контентом, конечными результатами может быть программное обеспечение, позволяющее пользователям управлять контентом, а также учебные материалы для сотрудников и конечных пользователей о том, как использовать недавно созданное программное обеспечение.

3) Работайте над созданием и поддержанием организационной согласованности

Существует два способа мышления об организационной согласованности:

- взгляд, ориентированный на организацию;
- взгляд, ориентированный на сотрудников.

Взгляд, ориентированный на организацию, подчеркивает несколько важных компонентов организации, поддерживающих друг друга. Цель, стратегия, возможности, структура и системы компании должны работать вместе.

Взгляд, ориентированный на сотрудников, побуждает менеджеров оценивать, насколько хорошо сотрудник подходит с точки зрения индивидуальной роли, профессиональных целей, членства в команде, а также видения и миссии организации.

Как руководитель проекта, вы, возможно, не контролируете все эти факторы, но в той мере, в какой вы можете повлиять на изменения, вам следует использовать эти концепции организационной согласованности для более успешного проекта.

4) Имейте четкие командные роли и обязанности

Мало что вызывает большую путаницу и напряженность в команде, чем отсутствие ясности в отношении ролей и обязанностей. Когда проектная команда не знает, каковы их роли или как эти роли связаны с другими людьми в команде, границы пересекаются и возникают ненужные конфликты.

Как руководитель проекта, вы несете ответственность за четкое определение роли каждого члена команды, чтобы помочь всем хорошо работать вместе.

5) Создайте стратегию для инициирования и выполнения

Инициирование проекта включает в себя всю предварительную работу, которая должна быть проделана до начала любых других мероприятий по проекту. Эту предварительную работу можно разделить на три категории:

- составление бизнес-обоснования проекта,
- составление отчетов о технико-экономическом обосновании проекта с участием заинтересованных сторон проекта
- создание Документа об инициировании проекта (PID).

Выполнение проекта — это то, что большинство людей имеет в виду, когда думают об управлении проектами. Обычно это начинается с совещания по запуску проекта, чтобы официально начать проект. Это когда вы делитесь видением и планом проекта, делегируете задачи членам команды и отправляете всех в путь, чтобы все было сделано.

На этапе выполнения убедитесь, что существует план документирования ошибок, исправлений и других изменений.

б) Знайте «все цифры», тщательно составляйте бюджет и планируйте

Каждый проект и каждый руководитель проекта имеют ограниченные ресурсы. Почти само собой разумеется, что вам необходимо тщательно планировать свои финансовые ресурсы, предоставлять себе некоторый запас на случай непредвиденных расходов и принимать разумные меры для экономии затрат в ходе вашего проекта. Ваш бюджет неразрывно связан с графиком вашего проекта; если ваш график будет нарушен, ваш бюджет проекта, вероятно, тоже будет нарушен.

Убедитесь, что у вас есть инструменты планирования проектов, и посмотрите, учитывали ли вы не только то, сколько времени должна занимать каждая задача проекта, но и такие вещи, как праздники, корпоративные мероприятия и мероприятия заинтересованных сторон, а также отпуска членов команды.

7) Заранее определите приоритеты и основные этапы

Приоритеты говорят вам, на чем сосредоточиться, а этапы проекта говорят вам, где вы находитесь. Когда вы находитесь в разгаре проекта, иногда легко отвлечься от важных вещей на менее важные детали, которые кажутся неотложными в данный момент. Когда вы определяете приоритеты своих задач в начале проекта, вы уже знаете, куда направить энергию вашей команды в случае возникновения конфликта. Также легко потерять представление об общей картине, когда вы вовлечены в детали проекта.

Определение основных этапов на этапе планирования проекта поможет вам понять, идете ли вы по курсу и по графику. Признание важных достижений также полезно для морального духа. Ваша команда будет более мотивирована, если будет ощутимое ощущение прогресса в проекте.

8) Создайте средства подотчетности и ответственности

Говоря о мотивации, одним из лучших мотиваторов для членов команды является наделение их чувством ответственности и подотчетности. Предоставляя отдельным лицам ответственность за их собственную работу, вы снимаете с себя бремя микроменеджмента и даете своим сотрудникам возможность использовать свои сильные стороны и осваивать новые навыки

управления проектами, что в итоге лучше для вашего проекта и вашей организации, не говоря уже о сотрудниках.

Частью предоставления членам команды индивидуальной ответственности является создание средств подотчетности. Вам нужна система, в которой можно было бы отслеживать делегирование задач и сроки выполнения проекта, и каждый член команды мог бы визуализировать свой вклад в контексте более крупного проекта.

9) Создайте коммуникационный план

Объясните и применяйте строгие принципы коммуникации с самого начала вашего проекта. Будь то электронная почта, текстовые сообщения, служба чата или какая-то комбинация вещей, убедитесь, что все члены команды понимают, почему общение в управлении проектами важно и как использовать выбранную вами технологию.

Вам также необходимо четко определить, какие виды информации необходимо передавать и кого необходимо уведомлять при определенных обстоятельствах. Как руководитель проекта, обязательно смоделируйте тот тип коммуникации, который вы ожидаете от всех заинтересованных сторон.

10) Будьте прозрачны

В сфере проектов прозрачность означает создание системы, в которой все члены команды могут легко и эффективно получать доступ ко всей соответствующей информации о проекте. Создать прозрачность проекта относительно легко, особенно если вы используете правильное программное обеспечение для управления проектами.

Чтобы создать или повысить прозрачность проекта, позвольте всем увидеть общую картину, сделайте данные проекта доступными для всей вашей команды, предоставьте хорошие инструменты для совместной работы и поделитесь календарями между членами команды и даже внешними заинтересованными сторонами. Прозрачность проекта приводит к лучшим результатам как для команды, так и для самого проекта.

11) Проведите оценку рисков

Оценка риска — это признание того, что что-то может пойти не так. Важно определить риски управления проектом и снизить их в начале вашего проекта, а не быть застигнутым врасплох позже. Спросите членов вашей команды, какие риски, по их мнению, вам необходимо учитывать.

Вы не можете исключить все риски из своего проекта, но готовность к ним может спасти вас от провала проекта.

12) Контролируйте течение проекта, выполняйте мониторинг и измерение прогресса

В рамках планирования вашего проекта вы установите ключевые показатели эффективности управления проектами в виде бюджета, сроков реализации проекта и ожидания в отношении качества.

В ходе вашего проекта вы должны регулярно отслеживать ход выполнения проекта и проверять свои ключевые показатели эффективности, чтобы вы могли быстро выявлять проблемы и вносить исправления. Не

забывайте отмечать успехи, когда ваши ключевые показатели эффективности говорят вам, что команда достигла цели!

Когда дело доходит до управления проектами, нужно многому научиться, но эти двенадцать принципов – отличная отправная точка. Если вы стремитесь к ясности во всех аспектах управления проектами и ставите во главу угла здоровье своей команды, вы на верном пути к тому, чтобы успешно завершить проект.

МОДУЛЬ 2. ОСНОВЫ ЭФФЕКТИВНОГО УПРАВЛЕНИЯ ПРОЕКТАМИ

2.3. Тема № 3. Основы управления проектами

2.3.1. Основные качества управляемого проекта

Как мы уже успели узнать, проект — это совокупность взаимосвязанных мероприятий для достижения определенной цели в рамках графика, бюджета и качества. Этот процесс включает в себя координацию групповой деятельности, при которой менеджер планирует, организует, направляет персонал и контролирует достижение цели с ограничениями по времени, стоимости и производительности конечного продукта. Управление проектами — это сочетание проекта и управления.

Давайте рассмотрим основные характеристики проекта.

1) Цели.

Каждый проект начинается с какой-то цели или с нескольких целей, например время, бюджет, качество и количество. Когда цели проекта не выполнены, появляются проблемы. Вы можете изначально определить цели проекта, чего на самом деле нужно достичь. Цели — это ключевые характеристики проекта, вместе с которыми вы будете видеть ход выполнения проекта, а анализ времени от времени покажет вам результат того, чего вы достигли.

2) Целостность.

Проект - это одно целое. Это означает, что проект несмотря на то, что разные люди вносят свой вклад, все равно признается единым целым. Команды часто специально собираются для одного проекта.

3) Продолжительность жизни.

Ни один проект не может быть непрерывным и неопределенным. У него должна быть точка завершения, дальше чего он не может продвинуться. Каждый проект неизменно ограничен по времени. Во время планирования вы увидите временную фазу проекта, на которой команда может независимо работать над модулями проекта. Давайте рассмотрим пример проекта, который разделен на три модуля, скажем, А, В и С. Если общий промежуток времени проекта составляет 5 месяцев, то вы можете установить промежуток времени для модулей независимо, например, А может быть завершен за 2 месяца, а также В может быть завершен за 2 месяца, а С может быть завершен за 1 месяц в соответствии с требованиями.

4) Требуемые средства.

Каждому проекту нужны средства, чтобы достичь конечной точки. Без достаточных средств ни один проект не может быть успешно реализован. Оценка затрат является одним из важнейших факторов для любой организации. Таким образом, предварительный расчет необходимых средств для проекта будет очень эффективным.

5) Жизненный цикл.

Каждый проект имеет жизненный цикл с различными стадиями, такими как начало, рост, зрелость и упадок. Проект должен пройти через различные

этапы, чтобы считаться завершенным. Давайте рассмотрим пример, когда проект связан с разработкой программного обеспечения, тогда вы можете сказать, что SDLC (Software Development lifecycle, жизненный цикл разработки программного обеспечения) будет жизненным циклом проекта, где вы увидите множество этапов, таких как планирование, определение, проектирование, сборка, тестирование и развертывание и т.д.

6) Командный Дух.

Командный дух необходим для завершения проекта, потому что в проекте участвуют разные участники с разными характеристиками и из разных дисциплин. Но для достижения общей цели необходимы гармония, миссионерское рвение, командный дух.

7) Риск и неопределенность.

Проект, как правило, основан на прогнозировании. Таким образом, риск и неопределенность всегда связаны с проектами. В тех проектах, которые не определены должным образом, будет высокая степень риска. Только степень контроля над рисками и неопределенностью варьируется в зависимости от задумываемого проекта на основе имеющейся информации.

8) Направления.

Проект всегда выполняется в соответствии с указаниями заказчиков в отношении времени, качества и количества и т.д. Удобство, экономические аспекты, доступность рабочей силы или ресурсов, управленческий талант и т.д., - все это второстепенные проблемы, первичными являются требования заказчика.

9) Уникальность.

Каждый проект уникален сам по себе, и у него есть свои особенности. Нет двух одинаковых проектов, даже если тип организации один и тот же. Уникальность проекта можно измерить, учитывая множество факторов, таких как цели, особенности проекта, применение проекта и т.д.

10) Гибкость.

Изменение и проект - синонимы. Проект претерпевает множество изменений на протяжении всего своего жизненного цикла. Эти изменения могут сделать проекты более динамичными и гибкими.

11) Субподряд.

Субподряд — это подмножество каждого проекта, без которого ни один проект не может быть завершен, если только он не является частной фирмой или крошечным по своей природе. Чем сложнее проект, тем больше будет объем контрактов. Каждый проект нуждается в помощи стороннего консультанта, инженера или эксперта в этой области.

12) Стоимость.

Стоимость проекта может быть динамической величиной. Если качество проекта будет изменено, это может повлиять на стоимость проекта. Стоимость может увеличиться, если для более быстрого завершения проекта потребуются больше ресурсов.

Таким образом, мы определили 12 основных характеристик проекта, а далее перейдем к вопросам внедрения в существующий проект современных методов управления.

2.3.2. Подготовка проекта к внедрению современных методов управления

Часто встречаются ситуации, когда существующий проект сопровождается без применения современных методов и инструментом. В какой-то момент руководство проекта решает перейти на «новые рельсы» с имеющимися результатами работы и с незаконченными задачами.

В этом случае необходимо иметь план и шаблонные решения для «бесшовной» интеграции существующих подходов к управлению и новых методик.

Давайте рассмотрим необходимые шаги для «миграции».

1) Необходимо создать план миграции.

Самое важное, что следует учитывать при перемещении проекта, — это то, как миграция повлияет на процессы, протекающие внутри проекта. Обычно для начала рассматривают все имеющиеся ресурсы проекта и все команды, работающие над проектом, как единое целое.

Хотя миграция не всегда приведет к прямым изменениям существующей конфигурации проекта, изменению иерархии ресурсов, но вероятно, окажет влияние на функционирование проекта и его запущенных процессов.

Возможно даже временная остановка проекта и адаптация команды.

Перед миграцией вашего проекта следует рассмотреть возможность создания плана миграции, чтобы определить готовность как вашей организации, так и проектов, которые вы хотите мигрировать на новую технологию. В плане миграции проведите инвентаризацию каждого процесса, запущенного в вашем проекте, и любых других процессов, на которые может повлиять миграция.

2) Инвентаризация.

Необходимо провести ревизию и инвентаризацию проекта: всех задач, имеющихся результатов работы, ресурсов команд, и даже имеющихся компетенций сотрудников.

Инвентаризация должна заканчиваться оформленным документом, входящим в план миграции.

На основании этого документа можно прогнозировать стоимость и длительность процесса миграции, а также управлять рисками проекта.

3) Проверка политики.

Когда вы перенесете свой проект, он больше не будет полностью наследовать политики существующих ресурсов, иерархию ресурсов и будет подвергнут переоценке политики в рамках новых процессов. Необходимо убедиться, что «новые» политики после миграции в максимально возможной степени будут соответствовать «прошлым» политикам.

Любая политика, применяемая непосредственно к проекту, все равно будет заменена после завершения миграции. Так что необходимо убедиться в том, что правильные политики применяются с момента завершения миграции.

Примерами таких политик могут быть: политики управления идентификацией и доступом, а также политики организации, политики распределения ресурсов.

4) Предварительный просмотр.

Необходимо «посмотреть в будущее» и определить, как будет выглядеть проект после применения миграций. Каждая внедренная функция, политика, процесс должны работать эффективно после перехода, а также заменяться/дополняться новыми функциями, политиками и процессами.

5) План возврата.

План возврата разрабатывается на случай полного или частичного «отката» миграции, если она окажется неудачной. Например, рабочие процессы окажутся не готовы к переходу или нет соответствующих специалистов-управленцев с должной квалификацией.

Такой план позволит взглянуть с другой стороны на процесс миграции и «подстраховаться» на случай неудачи.

б) Оценка стоимости перехода.

Миграция – это такой же процесс управления в проекте, как и другие. Нужно понимать, что процессы имеют временную, стоимостную оценки, а также требуют времени. К тому же может быть необходимо приобрести дополнительные инструменты разовыми платежами или ежемесячной платой, — это дополнительные расходы, которые должны быть учтены и оценена целесообразность процесса перехода.

Грамотно реализовать переход проекта к новым формам управления помогут акценты и основные точки внимания.

2.3.3. Акценты и основные точки внимания для внедрения

Можно рассматривать миграцию как отдельный технический проект, а можно как социальный. Социальный потому, что нужно изменить и дополнить существующие коммуникативные формы в организации, - людям нужно взаимодействовать в рабочих процессах и делать это максимально эффективно.

Если ставить акцент на техническую реализацию процесса миграции, то мы получаем набор новых инструментов для различных процессов (старых и новых), но можем получить команду, неготовую к эффективной эксплуатации этих инструментов.

Если также однобоко поставить акцент на социальную составляющую современных технологий управления проектами, «взять на вооружение» модные технологии Agile, Kanban, Scram, meetings, task boards, time tracking и так далее, то мы получим отлично взаимодействующую команду, неумеющую работать с инструментами.

Следовательно, необходимо правильно расставлять приоритеты в процессе миграции, обучать персонал, приобретать и овладевать современными инструментами эффективной работы.

2.3.4. Риски внедрения системы управления проектами

Как мы уже говорили выше, миграция – это процесс, и как любой процесс, она сопряжена с определенными рисками. Тем более, что миграция может затронуть практически все процессы работы на предприятии. По крайней мере на какое-либо время.

Вообще само понятие «риск» - многогранно. Это сложное явление, которое может иметь множество причин и может возникнуть под воздействием различных факторов. Именно поэтому, единого определения риска в современной литературе не существует.

В широком смысле под риском понимается возможность отклонения от цели, для достижения которой принималось то или иное решение.

В любом случае, даже пользуясь обобщенными критериями, рисками проекта нужно управлять, то есть рассматривать их с точки зрения проектного менеджмента.

А в проектном менеджменте риск – это неопределенное события или условие, наступление которого положительно или отрицательно повлияет на достижение целей проекта.

Собственно говоря, выявление рисков, в том числе и миграционных, определяет последствия наступления событий, которые необходимо предотвратить, уменьшить или устранить.

В общем говоря, мы можем разделить риски миграции на:

- 1) общие;
- 2) специфические.

Общие риски, - это деловые риски, технические, временные и финансовые.

Специфические риски, - это организационные, интеграционные, риски неприятия изменений, коммерческие и другие.

Деловые риски обусловлены внедрением гибких методологий в первую очередь. Особенно если миграция осуществляется для проектов, без учета их ценности для пользователей. До проведения миграции в случае определения деловых рисков, следует создать специальную межфункциональную группу, члены которых обладают знаниями не только технического характера, но и знаниями потребностей потребителей, так как гибкие методологии в основном направлены на достижения целей заказчика и ориентацию на изменения.

Технические риски миграционных процессов – это второй по распространенности вид рисков. Он заключается в том, что команде нужно осваивать новые инструменты до определенного уровня. Некоторые инструменты интуитивно понятны, но для некоторых требуется определенное образование или дополнительные курсы.

Следует учитывать, что современные технические средства имеют множество функций, умелое обращение с которыми повышает эффективность работы.

Собственно говоря, команда должна быть готова к нивелированию таких рисков до начала миграции, чтобы не останавливать процесс разработки или производства из-за незнания новых инструментов.

Финансовые риски – это третий по распространенности вид риска. Минимальный перерасход бюджета получается, когда технические инструменты бесплатны или имеют невысокую стоимость – в таком случае финансовые риски связаны в основном с задержкой этапов выполнения проекта. Однако нужно быть готовому к такому развитию событий, когда за самые современные и мощные инструменты придется платить дополнительно.

Тема № 5. Разработка концепции проекта и оценка его эффективности

5.1.1. Понятие концепции проекта

Мы уже определили проект как набор задач, которые необходимо выполнить, чтобы достичь определенной цели или результата. В зависимости от размера и масштаба проекта эти задачи могут быть простыми или сложными, но все проекты могут быть разбиты на цели и то, что необходимо сделать для их достижения.

Цели проекта и пути их достижения должны быть оформлены в виде специального «каркаса» или «модели», в которых описаны все основные факторы проекта.

В концепции проекта должна быть указана цель его создания, целевая аудитория, шаги, направленные на достижение цели и т.д.

Концепция проекта обычно содержит следующую информацию:

- 1) обоснование актуальности проекта, которая определяется тем, насколько он значим для общества;
- 2) описание потребностей, которые можно удовлетворить, реализовав предлагаемый проект;
- 3) обоснование инициации проекта — почему реализация предлагаемого проекта является лучшим решением описанной проблемы;
- 4) описание альтернатив предлагаемого проекта (какими еще способами можно решить возникшую проблему или удовлетворить потребности);
- 5) формирование целей и задач проекта, т.е. чего необходимо достичь в результате реализации проекта и какими конкретно действиями;
- 6) описание конечного результата проекта (продукта или услуги) с указанием его качественных и количественных характеристик;
- 7) критерии успеха проекта (количественные и качественные показатели);
- 8) перечень заинтересованных сторон проекта с указанием их целей, включая активных участников проекта (заказчики, исполнители, подрядчики, инвесторы и т.д.);
- 9) описание процедур сотрудничества и взаимодействия между участниками проекта;
- 10) предварительная оценка стоимости и сроков проекта;
- 11) предполагаемые источники финансирования и предварительный бюджет проекта;
- 12) укрупненный план реализации проекта;

13) план управления рисками.

Концепция содержит в себе некую дорожную карту с ключевыми событиями и основными этапами реализации проекта, которая в дальнейшем помогает руководителю проекта сформировать детальный план работ и определить контрольные точки с заранее определенными показателями эффективности.

На основе концепции проекта также составляется бюджет проекта и регламент взаимодействия участников проектной деятельности.

Концепция позволяет не просто описать, но и формализовать идею, обосновать ее реализуемость и эффективность, оцифровать основные и промежуточные результаты реализации проекта и определить ключевые факторы успеха.

Таким образом, концепция проекта является незаменимым инструментом защиты проекта на стадии инициации, а также является основой для формирования документов на стадии планирования проекта и преследует следующие цели:

- 1) Планирование проекта (стратегическое и оперативное в виде плана-графика реализации проекта).
- 2) Составление бюджета проекта (определение инвестиционных затрат и источников финансирования).
- 3) Обоснование проекта (его актуальность и эффективность).
- 4) Определение заинтересованных сторон.

При составлении концепции проекта, как правило, используют готовые шаблоны документов, принятых в компании. Также используется опыт подготовки аналогичных документов по другим проектам. Главную роль в данном процессе играет квалификация руководителя проекта, а также менеджера, ответственного за ее разработку.

Условно бизнес-процесс подготовки концепции проекта можно разделить на четыре этапа.

- 1) Начальный этап – встреча с заказчиком и инвестором проекта, обсуждение основных положений и общей идеи концепции, ее целей и задач.
- 2) Структуризация – систематизация исходных данных для формирования концепции, основных требований к формулировкам. Также она включает определение этапов и сроков написания документа, дополнительно необходимых работ, например, предварительного маркетингового исследования.
- 3) Написание концепции – формирование документа в соответствии с разработанной структурой и собранными исходными данными. Это выполнение работ по формализации идеи, обоснованию ее реализуемости, определению ключевых факторов успеха проекта.
- 4) Согласование и утверждение – согласование разделов концепции, в т.ч. целей и задач проекта, конечного результата, объемов и источников финансирования, методов реализации и т. п.

Утверждается концепция проекта только после согласования ее всеми заинтересованными сторонами.

Концепция проекта фактически определяет его профиль и направленность с самого начала работы.

5.1.2. Разработка и фиксация концептуального профиля проекта

Формирование концепции проекта начинается с этапа его инициации.

Инициация проекта — это первая стадия жизненного цикла проекта, означающая его запуск. Это убеждение руководства организации или инвесторов, а также других заинтересованных лиц в необходимости выполнения проекта. Фаза инициации проекта подразумевает определение целей проекта, а также взаимосвязанных с ними целей продукта проекта. Успех проекта определяется достижением конкретных показателей по каждому сформулированному результату в определенные сроки.

Необходимым условием эффективного управления проектом является тщательное документирование всех действий, связанных с исполнением проекта, поэтому результатом фазы инициации является документ, который может быть назван концепцией (или паспортом) проекта.

Профиль по определению, — это согласованный (непротиворечивый) набор стандартов и/или их фрагментов (т. е. выбор из них), созданный для нормативного обеспечения конкретного проекта или задачи.

Для ИТ-проектов профиль означает набор стандартов и рекомендаций, применяемых для обеспечения качества разработки продукта при помощи методов и средств управления.

Профили делятся на следующие виды:

- 1) профили, описывающие методологию разработки («технологические профили»);
- 2) профили, описывающие технологии, применяемые в разрабатываемом программном продукте («функциональные профили»).

Профили дополняются и корректируются по мере их применения вплоть до завершения разработки.

По профилю проекта формируется его портфель или портфели.

5.1.3. Портфель проекта

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

Портфель проекта состоит из элементов или составляющих портфеля. Это могут быть субпроекты, программы, субпортфели и так далее. Они могут быть зависимы или независимы друг от друга.

Сам портфель и его элементы должны иметь следующие характеристики.

- 1) Отражать существующие или потенциальные инвестиции компании.

- 2) Соответствовать целям и задачам компании.
- 3) Иметь такие отличительные особенности, которые позволяли бы группировать их для более эффективного управления.
- 4) Поддаваться количественному определению.

Для портфелей проектов характерны следующие детали:

- 1) Деловая область, которая меняется и развивается вместе со стратегическими целями организации.
- 2) Руководители портфеля проектов постоянно отслеживают изменения в окружающей среде.
- 3) Успех измеряется в терминах совокупного результата составляющих портфеля.
- 4) Стиль и степень лидерства должен помогать в принятии решений по портфелю проектов.
- 5) Руководители портфеля проектов могут руководить штатом управления портфелем проекта или координировать его деятельность.
- 6) Руководители портфеля проектов — лидеры, обеспечивающие способность проникновения в суть проблем и в процесс синтеза их решений.
- 7) Руководители портфеля проектов создают и поддерживают необходимый процесс и коммуникации относительно совокупного портфеля как внутри команды, так и с внешними субъектами.
- 8) Руководители портфеля проектов контролируют совокупную работу и оценивают показатели.

Портфелями проектов можно и нужно управлять. Это сложный процесс, который требует определенных знаний и навыков.

Управление портфелем проекта – это один из видов управленческой деятельности, который включает в себя:

- 1) идентификацию;
- 2) определение приоритетов;
- 3) авторизацию;
- 4) контроль выполнения проекта;
- 5) управление различными видами работ;
- 6) достижение различных стратегических целей.

При всем этом следует отличать два процесса разных уровней:

- 1) управление портфелем проекта;
- 2) управление портфелем проектов.

В первом случае мы выполняем управление совокупной структурой методов, инструментов и средств для одного проекта, а во втором случае – управление совокупностью нескольких проектов, для каждого из которых может существовать собственный портфель.

Для управления портфелем проектов существуют собственные процессы. Обычно их делят на следующие группы:

- 1) Идентификация проектов.
- 2) Категоризация проектов.

- 3) Оценка проектов.
- 4) Отбор проектов.
- 5) Определение приоритетов.
- 6) Балансирование портфеля.
- 7) Авторизация проектов.
- 8) Пересмотр портфеля и составление отчетов.
- 9) Стратегические изменения.

Стадия **идентификации проекта** представляет собой первый шаг в процессе продвижения проекта. Основная задача стадии идентификации – систематизация разрозненной исходной информации и первоначальная оценка идеи, сформулированной либо на основе выявленной проблемы, или проявившейся в тех или иных обстоятельствах рыночной возможности.

Категоризация проектов — компонент группы процессов выравнивания при управлении портфелем проектов. Категоризация проектов подразумевает под собой процесс объединения компонентов в однородные группы, имеющие общие цели, критерии оценки, селекции, независимо от места происхождения в организации. Категоризация, среди прочего, позволяет произвести балансирование портфеля.

Оценка проектов в портфеле проектов — это анализ отдельных проектов, для составления итогового инвестиционного портфеля компании, позволяющий ей правильно оценить и выгодно распределить финансовые средства по различным проектам, и получить с этого максимальную прибыль.

Отбор проектов заключается в проведении ряда конкурсов в условиях конкурентности. Обычно у компании, управляющей портфелем проектов, процесс отбора проектов четко определен, поэтому каждый инициатор проекта претендента понимает, какие минимальные результаты ожидаются от его проекта для поддержания его конкурентоспособности.

Определение или расстановка приоритетов — компонент группы процессов выравнивания при управлении портфелем проектов. Цель — ранжирование компонентов в соответствие со стратегией или финансовыми аспектами (например, инновации, экономия в расходах, рост, техническое обслуживание), периодом инвестирования (краткосрочный, средний, долгосрочный), значением риска и доходности, организационными аспектами (такими как потребитель, поставщик, внутренние характеристики). Данный этап ранжирует компоненты для последующего утверждения и балансировки портфеля.

Балансировка портфеля проектов — это процесс формирования комплекса проектов с учётом их приоритетов, взаимосвязи, синергии, влияния на достижение целей организации. Целью данного процесса является подбор наборов компонент портфеля с наибольшим потенциалом к реализации организационных стратегических инициатив и достижению стратегических целей. Балансировка портфеля проектов позволяет планировать и распределять ресурсы (финансовые, материальные, информационные, человеческие) в соответствии со стратегическими направлениями, а также

дает возможность максимизировать выгоды от реализации портфеля проектов в рамках приемлемого для компании уровня риска.

Пересмотр проекта и подготовка отчетов и обзоров функционирования портфеля проектов — это компонент группы процессов мониторинга и контроля проектов при управлении портфелем проектов.

Стратегические изменения — это главные носители нового качества в процессе развития организации, которые ключевым объектом управления при реализации не только специализированной стратегии, но и в целом корпоративной стратегии.

Давайте рассмотрим далее понятие инвестиционного профиля проекта.

5.1.4. Инвестиционный профиль проекта

Инвестиционный профиль проекта – это достаточно специфическое понятие, объединяющее в себе набор характеристик инвестора, его своеобразный портрет, который описывает поведение на финансовом рынке.

Составление такого профиля помогает управляющему определить запросы клиента, его склонность к рискам и готовность их принимать. Инвестиционный профиль проекта иногда называют риск-профилем. В зависимости от определенных параметров риск-профиль помогает:

- 1) Подобрать эффективную стратегию инвестирования.
- 2) Собрать набор (пул) ценных бумаг и других активов для создания оптимального инвестиционного портфеля.
- 3) Разработать механизмы реакции на различные рыночные события.

Составление инвестиционного профиля – не прихоть управляющей компании и не зависит от желания клиента предоставлять запрошенную информацию.

Давайте рассмотрим виды риск-профилей.

- 1) Консервативный.
- 2) Умеренный.
- 3) Агрессивный.

Консервативный риск-профиль характеризуется тем, что инвестор вкладывает средства в основном для сохранения капитала и защиты его от обесценивания, даже от процессов инфляции. Сокращение производства для такого клиента практически неприемлемо или допустимо в незначительной мере. Соответственно, принимать он готов только минимальные риски и соглашается на доходность инвестиций, даже незначительно превышающую показатель инфляции.

Умеренный инвестиционный риск-профиль характеризует инвестора как специалиста, который интересуется стабильным накоплением капитала. Как правило, клиент склонен реинвестировать доход, полученный от вложенных средств, и собирается пополнять счет деньгами из других источников. Минимальная доходность, незначительно превышающая инфляцию, его не устраивает, прибыль от инвестиций должна быть значительно выше этого уровня, но при ограниченных уровнях рисков.

Другими словами, инвестор готов рисковать, но не согласен терять значительную часть накопленного капитала от сделок.

Агрессивный. Такой инвестор приходит не для создания источника пассивного дохода. Его интересует возможность заработать на проекте действительно крупное состояние, которое не просто удовлетворит запросы, но станет базой для дальнейшего прогресса, например, в собственном бизнесе. Он готов принимать значительные риски, если это обеспечит максимальную доходность инвестиций. При этом инвестора не останавливает даже возможность частичной потери инвесткапитала, если стратегия имеет потенциал для быстрой компенсации полученных убытков и дальнейшего значительного роста прибыли.

Следует заметить, что в этом плане управление ИТ-проектом сильно напоминает управление инвестиционным портфелем или проектом. Оба этих вида деятельности требуют внимательного обращения и грамотного управления во многих аспектах. Одним из подобных аспектов является управление рисками.

5.1.5. Риски проекта и методики управления рисками

Можно долго говорить о сходствах и различиях между различными видами проектов, - мы сейчас рассматриваем ИТ-проекты с их достоинствами и недостатками, а также характерными особенностями.

Источники рисков могут быть различными. Вообще говоря, об управлении рисками ИТ-проектов, крайне важно определить возможные источники таких рисков. Когда мы сможем классифицировать возможные ИТ-риски, их поиск и управление ими будет намного проще. Однако категории должны отражать характер конкретного проекта. В контексте управления ИТ-рисками мы можем перечислить три основных типа рисков, связанных с ИТ.

Это в первую очередь:

- 1) Внешние риски.
- 2) Внутренние риски.
- 3) Персональные риски.

В течение жизненного цикла проекта происходит много изменений – концепции, требования, количество задач в спринте и приоритеты - все это может измениться. Иногда клиент может передумать на полпути к выполнению поставленной задачи, что также увеличивает стоимость.

Последствия таких рисков могут быть различными.

- 1) Спринты могут быть перегружены или недогружены.
- 2) Задачи могут быть частично или полностью не завершены.
- 3) Изменения могут привести к полной или частичной переработке приложения.
- 4) Могут быть изменения в расписании.
- 5) Внезапная необходимость добавления большего количества людей в команду.

В качестве решения такой проблемы может быть **управление изменениями**.

Когда вы вносите изменения, жизненно важно проанализировать, какое влияние они оказывают на текущее состояние проекта, сколько усилий это потребует или существует ли риск задержек. Анализ позволит эффективно распределять задачи, обновлять приоритеты и предоставлять клиенту точную информацию о том, что может (и не может) быть фактически доставлено.

Отказ от ответа в данном случае был бы синонимом безоговорочного принятия их. Это обычная практика в процессе разработки программного обеспечения. Вот почему так важно, чтобы все заинтересованные стороны проекта осознали последствия изменений и совместно согласились на некоторые компромиссы, если они необходимы.

Еще бывает проблема **«отсутствия приверженности»**. Приверженность всех членов команды необходима для успеха каждого проекта. Вот почему важно, чтобы каждый член команды был привержен общей цели, понимал свою роль и поддерживал других участников.

Необходимо обратить внимание на других членов команды и попытаться выяснить, что может укрепить их приверженность. Они должны чувствовать себя комфортно, но в то же время уметь сосредоточиться на своей работе. Пусть они растут индивидуально, следует общаться с ними и мотивировать. Также следует предоставить членам команды как можно больше подробностей о проектах, чтобы они чувствовали себя важной частью чего-то большего.

Это важная часть системы управления проектами, так как недостаток коммуникаций в проекте может привести к провалу.

Регулярные встречи всех членов команды как для выполнения задач, так и для обмена знаниями, созданными в рамках проекта. Участники собрания должны чувствовать себя в безопасности и иметь каждому возможность высказаться. Никогда нельзя оставлять ничьи вопросы без ответа. Если даже сразу ответ неизвестен, то следует сообщить членам команды, что вы собираетесь попытаться найти его и вернуться с ним позже.

Очень важно, чтобы каждый понимал свою роль в проекте. Встреча вне рабочего контекста также может быть хорошим способом улучшить атмосферу в команде.

Другой проблемой и дополнительным риском в команде можно назвать **недостаток документации**.

Что такое проектная документация? Минимальный жизнеспособный продукт (MVP), описания задач в JIRA и пространство проекта в Confluence – все это очень важно для успеха проекта.

Недостаточная документация может привести к хаосу в команде, излишним затратам времени, повторному разбору вопросов, снижению общей квалификации и ротациям в команде.

Даже минимальная проектная документация имеет большое значение для предотвращения наихудших последствий.

Следует помнить, что согласно лучшим гибким практикам: **«работающее программное обеспечение важнее подробной документации»**.

Однако это не должно означать, что документация не важна. Как лучше всего решить эту проблему?

Следует обязательно выделить некоторое время для написания документации с самого начала. Следует использовать такие мощные инструменты, как JIRA, Confluence или QA Touch – они действительно упрощают работу. Существует также множество других более специализированных инструментов, которые помогут вам написать документацию результатов проекта. Определите, какая информация всегда должна быть доступна. Хорошее место для хранения документации в Confluence, но сгодится и папка в облаке типа Google Drive. В любом случае, это место, где можно найти всю основную информацию о проекте, членах команды и их ролях, паролях и другой информации о доступе к важным свойствам проекта и среде, описаниям пользователей или списку функций.

Последовательно следует использовать определенное соглашение для именования и описания задач. При этом следует помнить, что документация не должна быть громоздкой. Ее задача состоит в том, чтобы описать весь проект в ясной и понятной форме, чтобы было как можно меньше вопросов у членов команды.

Еще одним часто встречающимся риском ИТ-проекта можно назвать **незапланированное отсутствие члена команды**. Каждое незапланированное отсутствие члена команды является поводом для беспокойства. Например, в осенне-зимний период риск заболеть особенно велик.

Логичные последствия такого риска, это дезорганизация, задержки в выполнении задач, недостаточные знания о проекте, особенно если человек был ключевым участником (опять же, это показывает важность хорошей документации), — все это демотивирует команду.

Главное в избежание такого риска, чтобы все члены команды делились одними и теми же необходимыми знаниями о проекте. В зависимости от того, как долго длится отсутствие и в каком состоянии находится проект, менеджер проекта должен принять решение о необходимости замены. При наличии знаний о проекте и документации новичку будет легко приступить к работе.

Одной из самых сложных проблем в ИТ является **плохая коммуникация с заказчиком**. Коммуникация страдает если клиент не отвечает, допускает длинные паузы между вопросом и ответом, отвечает скомкано или неинформативно в силу разных причин. Эта же проблема может быть и внутри команды.

Этот риск ИТ-проекта приводит к задержкам в решении задачи и к демотивации команды.

Решением проблемы может быть демонстрация клиенту, как важно для вас поддерживать с ним хорошие отношения с первой встречи. Определите, какие решения всегда следует принимать сообща, а какие могут быть приняты разработчиками/руководителями самостоятельно. Когда вы отправляете электронное письмо с запросом клиенту, может быть хорошей идеей точно описать, почему задержка с ответом может быть проблематичной (например, трудности с соблюдением заданного срока). Если ни один из способов

связаться с клиентом не работает, руководитель проекта должен принять меры для улучшения коммуникации с клиентом.

Несвоевременная доставка результатов работы. К сожалению, такой риск в IT не редкость. Особенно если в проекте отсутствуют четко отлаженные механизмы непрерывной интеграции, развертывания и доставки. Важные части проекта должны доставляться незамедлительно. В серьезных проектах задержка в поставке даже небольших «фич» может привести к проблемам.

Когда вы планируете крайние сроки для проекта или / и спринта, примите во внимание все возможные факторы, подобные этому. Проанализируйте возможные риски и проинформируйте о них клиента. Всегда назначайте задачи с учетом количества доступных членов команды, а также их навыков, сильных и слабых сторон. Всегда сообщайте о своем прогрессе и устраняйте все проблемы / препятствия во время ежедневного митинга. Это лучший способ обеспечить качество разработки программного обеспечения.

Если сроки не могут быть соблюдены, крайне важно сообщить об этом клиенту как можно скорее. Хороший способ решить эту проблему - разделить большую задачу на пару более мелких. Лучше выполнить несколько небольших задач, чем вообще ничего не делать. Это также показывает клиенту вовлеченность команды и ее сильные стороны.

Риски IT-проекта необходимо предугадывать и выявлять заранее.

Собственно говоря, идентификация рисков может значительно повысить качество разработки программного обеспечения. Когда риска избегают, но в то же время учитывают при оценке, это может привести к тому, что задача будет выполнена даже раньше предполагаемых сроков – очень удобная ситуация для команды. Каждый член команды должен принимать участие в усилиях по выявлению рисков, быть осведомленным о последствиях возможных рисков и знать, как на них реагировать.

Когда вы должны начать идентификацию рисков? Естественно, с самого начала проекта. Следует начать с составления списка возможных рисков, которые следует учитывать в течение всего срока реализации проекта. Этот способ должен обновляться по мере продвижения.

Как же определить и идентифицировать риски.

- 1) Подробный анализ документации.
- 2) Подробный анализ целей проекта.
- 3) Составить контрольные списки, основанные на опыте предыдущих проектов.
- 4) Провести мозговой штурм – хотя бы простой разговор между всеми членами команды может иметь большое значение.

Давайте далее рассмотрим способы управления процессом исполнения IT-проектов.

5.1.6. Управление процессом исполнения IT-проектов

Выполнение проекта состоит из множества процессов, используемых для выполнения работ, определенных в плане проекта, для выполнения требований проекта. Процесс выполнения включает координацию людей и

ресурсов, а также интеграцию и выполнение мероприятий проекта в соответствии с планом проекта. Основным результатом выполнения проекта являются результаты проекта, и за их подготовку будет отвечать проектная группа, которая будет работать в соответствии с планом проекта.

Действия, выполняемые во время этого процесса, включают:

- 1) Выполнение мероприятий, необходимых для достижения целей проекта.
- 2) Получение и управление котировками, предложениями и предложениями по мере необходимости.
- 3) Управление проектной командой и управление другими ресурсами.
- 4) Сбор и анализ данных о производительности.
- 5) Генерация проектных данных для отчетов о состоянии и прогнозов.
- 6) Управление рисками.
- 7) Проведение контроля изменений и внедрение утвержденных изменений.
- 8) Создание и управление каналами связи проекта.
- 9) Сбор и документирование извлеченных уроков.

Мониторинг состоит в контроле процесса выполнения проекта. Это делается для того, чтобы можно было своевременно выявить потенциальные проблемы и при необходимости предпринять корректирующие действия. Мониторинг и контроль включают измерение текущей деятельности по проекту и переменных проекта (стоимость, усилия, объем и т.д.) в соответствии с планом проекта и базовыми показателями эффективности проекта. Затем в нем определяется, что необходимо сделать, чтобы вернуть проект в нужное русло.

В ходе любого проекта может измениться объем работ либо в результате необходимых изменений дизайна, различных условий на объекте, наличия материалов, изменений, запрошенных подрядчиком, разработки стоимости и воздействия третьих сторон, и это лишь некоторые из них.

Любое изменение должно быть задокументировано, чтобы показать, что было фактически сделано; это называется **управлением изменениями**.

Независимо от того, насколько тщательно был спланирован проект, по мере его продвижения необходимо будет вносить изменения. Это будет результатом как внешних воздействий, так и проблем, возникающих в среде проекта.

Четырьмя основными источниками изменений являются:

- 1) Организационные: Бизнес-решения высокого уровня могут изменить основное техническое задание проекта - например, могут быть внесены изменения в общий объем проекта.
- 2) Экологический: в результате изменений в законодательстве или изменений в государственной политике или изменений в бизнес-стратегии.
- 3) Технический: Новая технология может предложить лучшее решение по сравнению с первоначально запланированным. В качестве

альтернативы технические проблемы могут помешать продукту работать так, как предполагалось.

- 4) Конечный пользователь: результат изменений в требованиях заказчика. Также возможно, что отзывы, полученные во время обзора или тестирования продукта, могут показать, что он каким-то неожиданным образом непригоден для использования.

Любое лицо, связанное с проектом, должно иметь возможность в любое время высказать любую свою озабоченность. Озабоченность может быть связана с предполагаемой проблемой или предложением по улучшению какой-либо области работы, документации или организации проекта. Эти вопросы следует рассматривать на регулярных совещаниях. При рассмотрении вопроса возможны три исхода:

Изменение дизайна или характеристик продукта может быть согласовано. Это будет означать изменение способа указания продукта в планах и соответствующее обновление любых затрат и сроков. Также следует провести анализ воздействия. В этом процессе рассматриваются побочные эффекты изменений для других результатов, а также последствия, если изменения не будут реализованы. Цель анализа воздействия состоит в том, чтобы получить сбалансированное представление о влиянии предлагаемых изменений на способность проекта выполнять свой мандат. Это позволит руководству проекта решить, следует ли продолжать вносить изменения или нет.

Предлагаемое изменение отклоняется, поскольку оно не представляет серьезной проблемы.

Третий вариант необычен, но иногда случается, что результат не соответствует его спецификации, но изменение спецификации является лучшим решением, чем изменение результата.

В тех случаях, когда изменение результата считается наилучшим вариантом, руководитель проекта должен использовать анализ воздействия для оценки изменения с точки зрения его влияния на временные рамки, затраты, выгоды, качество, персонал и риски и решить, на каком уровне следует принимать решение о продолжении. Затем специалисты должны определить, является ли предлагаемое изменение достаточно значительным, чтобы его можно было вернуть спонсору или инвестору.

Управление процессом исполнения проекта также связано с процессами **управления командой проекта.**

В самом начале проекта процесс набора проектной команды связан с подтверждением наличия человеческих ресурсов и получением персонала, необходимого для выполнения проектных заданий. Привлечение проектной команды часто осложняется тем фактом, что руководитель проекта обычно не имеет прямого контроля над всеми, кого он хотел бы привлечь к проекту. Им может потребоваться провести переговоры с другими лицами, которые в состоянии предоставить нужное количество людей с соответствующим уровнем знаний, навыков и опыта.

Следующая цель - создать команду, которая сможет хорошо работать вместе. Этот процесс следует начинать на ранней стадии проекта, потому что формирование команды требует времени, а команда, которая хорошо работает вместе, окажет большое влияние на успех проекта. Эффективность работы команды можно повысить, используя открытую и эффективную коммуникацию, развивая доверие между членами команды, управляя конфликтами конструктивным образом и поощряя совместное решение проблем и принятие решений. Формирование команды - важная сложная тема, и начинающий руководитель проекта должен потратить время на изучение других материалов по этой теме. Например: Навыки повышения производительности, Лидерские качества, Навыки Оценки и Навыки проведения совещаний.

Как только начнется работа над проектом, руководителю проекта необходимо будет сосредоточиться на отслеживании производительности членов команды, предоставлении обратной связи, решении проблем и управлении изменениями для оптимизации производительности проекта. Управление людьми, вероятно, является одной из самых сложных областей, поскольку они могут быть непредсказуемыми, могут вступать в неожиданные конфликты друг с другом, их моральный дух может повышаться или понижаться, а некоторые могут даже решить покинуть проект до его завершения.

Следующий аспект связан с **руководством и управлением проектной работой**.

Этот процесс включает в себя руководство и выполнение работ, определенных в плане проекта, и внедрение утвержденных изменений для достижения целей проекта. Основными входными данными для этого процесса являются план управления проектом (подробно описанный в предыдущем разделе) и утвержденные запросы на изменения.

Другим немаловажным аспектом является **управление обеспечением качества проекта**.

Существует много путаницы в вопросах обеспечения качества и контроля качества. Разница заключается в том, что обеспечение качества является частью процесса выполнения и направлено на обеспечение достижения целей в области качества. Она направлена на совершенствование процессов. Этот процесс также обеспечивает основу для непрерывного совершенствования процессов, которое является итеративным средством повышения качества всех процессов с целью сокращения отходов и устранения видов деятельности, которые не повышают ценность.

Другим процессом является **управление коммуникацией с заинтересованными сторонами проекта**.

Цель этого процесса состоит в том, чтобы информировать различные заинтересованные стороны о ходе реализации проекта. Это делается в соответствии с планом коммуникаций, созданным в рамках процесса планирования коммуникаций.

Управление поставщиками для проекта также является важным для проекта.

Это процесс получения ответов поставщиков, выбора поставщика и заключения контракта. Это может потребоваться несколько раз, если имеется несколько контрактов, и для каждого случая это будет включать выдачу пакета предложений потенциальному поставщику, оценку предложений потенциальных поставщиков и, наконец, выбор победившего предложения.

Процесс выполнения работ включает в себя нечто большее, чем просто создание результатов проекта. В нем тесно переплетаются процессы мониторинга и контроля, а также дополнительные действия в сложно связанной команде.

5.1.7. Содержание ТЗ в соответствии с IEEE STD 830-1998

Далее рассмотрим один из основных современных стандартов на разработку программного обеспечения IEEE Recommended Practice for Software Requirements Specifications (SRS), то есть IEEE STD 830-1998.

Основной адрес размещения IEEE STD 830-1998 в сети интернет: [https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20\(RU\).md](https://github.com/maxvipon/IEEE-Std-830-1998-RU/blob/master/IEEE%20STD%20830-1998%20(RU).md).

Этот стандарт относится к группе стандартов, касающихся спецификации требований программного обеспечения, которые структурируют набор требований (функциональность, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам.

Данный стандарт предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком (или подрядчиками) о том, как должен функционировать программный продукт.

Может включать ряд пользовательских сценариев (англ. use cases), которые описывают варианты взаимодействия между пользователями и программным обеспечением.

Пользовательские сценарии являются средством представления функциональных требований. В дополнение к пользовательским сценариям спецификация также содержит нефункциональные требования, которые налагают ограничения на дизайн или реализацию (такие как требования производительности, стандарты качества, или проектные ограничения).

Документ описывает рекомендованный подход к разработке спецификаций требований к программному обеспечению. Они основываются на модели, в которой результатом процесса разработки спецификации требований к программному обеспечению является непротиворечивый и полный документ. Они призваны помочь:

- 1) Потребителям программного обеспечения — точно описать, что они желают получить.
- 2) Поставщикам программного обеспечения — в точности понять, что хочет потребитель.
- 3) Специалистам — выполнить следующие задачи:

- a. разработать план стандартной спецификации требований к программному обеспечению (SRS) для нужд вашей конкретной организации;
- b. определить формат и содержание своих собственных спецификаций требований к программному обеспечению;
- c. разработать собственные дополнительные инструменты поддержки, такие, как чек-листы качества SRS или справочники разработчиков SRS.

Содержание стандарта обеспечивает следующие преимущества для применения на практике:

- 1) можно установить основу для соглашения между потребителями и поставщиками о том, что должен делать программный продукт;
- 2) можно снизить трудоемкость разработки за счет четкого формирования требований и уменьшения времени на перепроектирование, перекодирование и перетестирование;
- 3) можно сформировать основу для оценки стоимости и сроков выполнения проекта;
- 4) можно создать основу для валидации и верификации приложений;
- 5) можно облегчить передачу программного обеспечения пользователям и заказчикам.
- 6) Можно создать основу для совершенствования программного обеспечения.

Стандарт содержит пять основных разделов:

- 1) Раздел 1 поясняет область применимости рекомендаций.
- 2) Раздел 2 перечисляет ссылки на другие стандарты.
- 3) В разделе 3 приведены определения основных используемых терминов.
- 4) Раздел 4 предоставляет дополнительную информацию по написанию хорошей SRS.
- 5) В разделе 5 обсуждаются все основные части SRS.

Следует помнить следующие особенности данного документа:

- 1) Данный документ представляет собой рекомендации по написанию спецификаций требований к программному обеспечению. Он описывает содержание и характеристики качественной спецификации требований к программному обеспечению (SRS) и представляет несколько примеров плана SRS.
- 2) Данные рекомендации ориентированы на специфицирование требований к разрабатываемому программному обеспечению, но могут также оказаться полезны при выборе самостоятельно разработанных и коммерческих продуктов. Однако их применение к уже разработанному программному обеспечению может отрицательно повлиять на продуктивность.
- 3) Если программное обеспечение встроено в некоторую большую систему, например, медицинское оборудование, то могут возникнуть проблемы, лежащие вне пределов применимости данных рекомендаций.

- 4) Данные рекомендации описывают процесс создания продукта и содержание продукта. Продуктом является SRS. Рекомендации могут быть непосредственно использованы для создания SRS или как модель для более специфического стандарта.
- 5) Данная рекомендация не предлагает никаких конкретных методов или инструментов для подготовки SRS.

В стандарте описываются следующие основные вопросы, которые должны решить между собой разработчики и заказчики:

- 1) Функциональность. Что должно по замыслу делать программное обеспечение?
- 2) Внешние интерфейсы. Как программное обеспечение взаимодействует с людьми, системным оборудованием, другим оборудованием и другими программами?
- 3) Производительность. Каковы скорость, доступность, время ответа, время восстановления различных программных функций и т.д.?
- 4) Атрибуты. Каковы переносимость, корректность, пригодность к поддержке, безопасность и т.д.?
- 5) Проектные ограничения, налагаемые на реализацию. Есть ли требования к действующим стандартам, языкам реализации, политикам поддержания целостности баз данных, ограничениям на ресурсы, операционной среде и т.д.?

В заключении отметим, что рекомендации по разработке программного обеспечения должны:

- 1) Корректно определять все требования к программному обеспечению (Требование к программному обеспечению может существовать благодаря природе решаемой задачи либо благодаря специфической особенности проекта).
- 2) Описывать общие характеристики программного обеспечения без конкретных деталей (детальное описание должно находиться в документации к проекту и формироваться ближе к завершению работы).
- 3) Убирать дополнительные ограничения на программное обеспечение (Они должны задаваться в других документах, например, плане обеспечения качества программного обеспечения).

В любом случае, рекомендации к программному обеспечению должны быть:

- 1) корректными;
- 2) непротиворечивыми;
- 3) полными;
- 4) целостными;
- 5) упорядоченными во важности и по стабильности системы;
- 6) верифицируемыми;
- 7) модифицируемыми;
- 8) трассируемыми.

Корректность выражается в удовлетворении ПО каждому установленному требованию.

Непротиворечивость выражается в единственной интерпретации каждого требования, термина и характеристики продукта. Это достаточно сложное качество для достижения, особенно в условиях работы с иностранными заказчиками. Требования часто пишутся на естественном языке (например, на английском), а естественные языки неоднозначны по своей природе (единственный однозначный язык сейчас – это латынь, но его в IT не используют). Поэтому нужно избегать неоднозначности понятий и особенно переводов на другие языки при помощи языков спецификаций требований или специальных подмножеств языков типа Gherkin.

Полнота требований определяется присутствием следующих специальных элементов:

- 1) Все значимые требования, имеющие отношение к функциональности, производительности, проектным ограничениям, атрибутам или внешним интерфейсам. В частности, следует принимать во внимание все внешние требования, налагаемые системной спецификацией.
- 2) Определение всех ответов программного обеспечения на все возможные классы входных данных во всех возможных классах ситуаций. Отметим, что важно задавать ответы как на допустимые, так и на недопустимые входные значения.
- 3) Все метки и ссылки на все рисунки, таблицы и диаграммы в SRS, а также определения всех терминов и единиц измерения.

Целостность, как понятие требований к ПО, относится в основном к внутренней целостности. Если какое-то требование к ПО не согласуется с каким-либо из документов высшего уровня, например, спецификацией требований к системе, вся система требований не является корректной.

Упорядоченность требований выражается в том, что каждое из них должно иметь идентификатор, который учитывает очередность, порядок или место в системе, ранг важности или стабильности. Как правило, не все требования, относящиеся к программному продукту, являются одинаково важными. Одни требования могут быть принципиально важными, особенно для жизненно-важных приложений, тогда как другие могут быть просто желательными.

Каждое требование в SRS должно быть идентифицировано с целью сделать эти различия отчетливыми и явными. Идентификация требований таким образом помогает:

- 1) потребителям — более тщательно рассмотреть каждое требование, что зачастую проясняет их неявные предположения.
- 2) разработчикам — принимать правильные проектные решения и соответственно распределять усилия, затрачиваемые на различные части программного продукта.

Верифицируемость требований выражается в том, что, если существует некий конечный достаточно эффективный процесс, согласно которому

человек либо машина может проверить, удовлетворяет ли программный продукт данному требованию. В общем случае никакое неоднозначное требование не является верифицируемым.

Неверифицируемые требования включают такие обороты, как «работает хорошо», «хороший человеко-машинный интерфейс» или «обычно должно происходить». Эти требования невозможно верифицировать, поскольку невозможно точно установить смысл терминов «хорошо», «хороши» или «обычно». Утверждение «программа никогда не должна входить в бесконечный цикл» не является верифицируемой, поскольку его тестирование невозможно даже теоретически.

Модифицируемость заключается в том, что при описании требований используется единая структура и стиль, так что изменения требований могут быть сделаны любым специалистом легко, полно и целостно без изменения структуры и стиля.

То есть, все элементы требований к ПО должны быть **ожидаемыми**. При этом, такое же требование к архитектуре и к принципам построения программного обеспечения также применимо.

Грубо говоря, модифицируемость документации и требований к ПО характеризуются:

- 1) связной и легкой в использовании организации документов с оглавлением, индексом и явными перекрестными ссылками.
- 2) отсутствием избыточности (то есть одно требование не должно упоминаться в двух местах);
- 3) раздельным представлений каждого требования и отсутствием смешивания.

Трассируемость выражается в том, что, если источник каждого требования ясен и если она в дальнейшем позволит ссылаться на каждое требование в документации разработки. Рекомендуются следующие два типа трассируемости:

- 1) Обратная трассируемость (т.е. к предыдущим этапам разработки). Она зависит от наличия в каждом требовании явной ссылки на его источник в предшествующей документации.
- 2) Прямая трассируемость (т.е. ко всем документам, порожденным SRS). Она зависит от наличия в SRS уникального имени или ссылочного номера для каждого требования.

Требования к ПО, согласно этому стандарту, подготавливаются совместно и этот процесс начинается с соглашения между поставщиком и потребителем программного обеспечения о том, что именно должно делать готовое программное обеспечение. Этот документ подготавливается с двух сторон и здесь особенно важно соблюдение чистоты требований, лаконичности и высокой совместной квалификации сторон.

Как правило, заказчиками являются представители бизнес-сообщества, а исполнители – представителями технической стороны. В этом случае, обычно, ни та, ни другая сторона не обладают поодиночке достаточными компетенциями для создания такого комплекта документов – требований.

Особая ситуация возникает, когда одновременно определяется система и ее программное обеспечение. В этом случае функциональность, интерфейсы, производительность, а также прочие атрибуты, и ограничения программного обеспечения не predetermined заранее на этапе инициации проекта, - они скорее определяются совместно в ходе работы и являются предметом обновляемых соглашений, изменений и компромиссов.

Давайте далее рассмотрим несколько другой стандарт в этой области деятельности, - с ISO/IEC/IEEE 29148-2011.

5.1.8. Содержание ТЗ в соответствии с ISO/IEC/IEEE 29148-2011

Стандарт ISO/IEC/IEEE 29148:2011 называется: «Программная и системная инженерия. Процессы жизненного цикла. Разработка требований».

Область применения стандарта:

- 1) определяет процессы, которые необходимо реализовать для разработки требований к системам и программным продуктам (включая сервисы) на протяжении всего жизненного цикла;
- 2) содержит руководства по применению требований и процессов, связанных с требованиями, описанных в стандартах ISO/IEC 12207:2008 (IEEE Std 12207-2008) и ISO/IEC 15288:2008 (IEEE Std 15288-2008);
- 3) определяет информационные единицы, которые должны быть произведены при реализации процессов;
- 4) определяет требуемое содержание требуемых единиц сведений и предоставляет руководство по определению формата требуемых и связанных единиц сведений

Стандарт представляет собой структурированный набор требований (функционал, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам. Предназначен для того, чтобы установить базу для соглашения между заказчиком и разработчиком о том, как должен функционировать программный продукт. Может включать ряд пользовательских сценариев (use cases), которые описывают варианты взаимодействия между пользователями и программным обеспечением.

Структура документа, согласно данному стандарту, может быть представлена в следующем виде:

- 1) Введение:
 - a. соглашение о терминах;
 - b. назначение;
 - c. объем проекта;
 - d. функции продукта.
- 2) Общее описание:
 - a. общее представление продукта;
 - b. классы и характеристики элементов продукта;
 - c. операционная среда:
 - операционная система;

- среда запуска приложения;
 - система управления базами данных;
 - интерфейсы обмена данными;
 - другое.
- d. ограничения дизайна и реализации;
 - e. документация, содержание и направленность.
- 3) Функциональность системы;
- a. функциональность отдельных сервисов системы;
 - b. объекты и их взаимодействие;
 - c. приоритеты системы;
 - d. действия и взаимодействия пользователей.
- 4) Требования к внешним интерфейсам:
- a. интерфейсы оборудования (если такие требуются) (НИ);
 - b. интерфейсы взаимодействия программных компонент (API);
 - c. интерфейсы передачи данных (ИПД).
- 5) Нефункциональные требования:
- a. требования к производительности (ТкП);
 - b. требования к сохранности данных (ТСД);
 - c. требования к защите данных (ТЗД);
 - d. требования к авторизации и к аутентификации;
 - e. требования к контролю;
 - f. требования к сегментированию;
 - g. требования к мониторингу;
 - h. другие требования.
- 6) Атрибуты качества и соответствия ПО:
- a. доступность;
 - b. безопасность;
 - c. производительность;
 - d. функциональность;
 - e. удобство;
 - f. поддерживаемость;
 - g. прочее.
- 7) История изменений.
- 8) Список согласования.
- 9) Прочее.
- 10) Приложения.

В любом случае, данный стандарт также является рекомендацией, которую можно принимать в проекте в полном объеме или изменять/дополнять в соответствии со спецификой проекта и по согласованию с заказчиком.

Тема № 6. Методы оценки эффективности проекта.

6.1. Основные анализа эффективности исполнения IT-проектов на всех этапах его жизненного цикла

Понятия эффективности и результативности обычно используются при оценке различных процессов. Поскольку управление IT-проектами может быть описано различными видами процессов, целью исследования эффективности является изучение концепций управления через призму управления качеством.

Поскольку проектные организации часто сталкиваются с проблемами баланса между временем, затратами и качеством, они заинтересованы в том, чтобы делать это как можно более эффективно и результативно. Однако существует большое разнообразие мнений об эффективности и результативности среди практиков и исследователей, что затрудняет применение этих концепций в условиях, основанных на концепции IT-проекта.

Зачастую термины «эффективность» и «результативность» используются без четких определений, а результаты исследований могут быть интерпретированы по-разному.

В этой связи, четкое понимание и толкование понятия эффективности поможет компании, разрабатывающей IT-проект, работать на совершенствование. Ясность подразумевает улучшение предварительных условий для измерения эффективности и эффективность и возможность разработки показателей, которые могут быть использованы для руководства организацией в желаемом направлении. Более четкое представление об эффективности и результативности проекта также может стать основой для внутренних улучшений с точки зрения времени, затрат и качества, а также внешних улучшений с точки зрения удовлетворенности клиентов.

В области управления проектами (PM) понятия эффективности и результативности обычно используются, но редко определяются четко. Некоторые исследователи применяют эти концепции при описании того, как улучшить какую-либо часть процесса управления проектом, некоторые применяют их при описании компетенций для выполнения проекта. В других моделях эффективность обычно указывается в качестве одной из причин измерения зрелости проекта.

Но достаточно редко в процессе управления проектом встречается разделение того, что подразумевается под эффективностью и результативностью, и поэтому эти понятия часто применяются взаимозаменяемо.

В области управления качеством (QM) эти концепции применяются более определенным образом. В QM эффективность относится к правильному выполнению задач, т.е. что бы ни выполнялось, оно выполняется наиболее подходящим способом, учитывая доступные ресурсы (высокая эффективность). Эффективность, с другой стороны, относится к тому, чтобы делать правильные вещи, то есть выбирать и фокусироваться на производстве продукции, на которую есть спрос. В повседневном употреблении слово

эффективный определяет то, что процесс «А» в сравнении с процессом «В» был реализован более оптимальным способом или был получен лучший результат при тех же ресурсах.

Но технология сравнения работает тогда, когда есть сопоставимые вещи, а в сфере разработки программного обеспечения, сам IT-проект понимается как нечто «уникальное», поэтому два разных проекта, выполненные в разное время и разными командами сложно сравнить.

В таких областях, как управление качеством и операциями, процессы занимают центральное место. Обширная литература в этих областях посвящена совершенствованию процессов, касающихся их результатов, самого процесса и его соответствия требованиям заказчика. В ней демонстрируют четкое применение концепций, связанных с теорией процессов на основе эффективности и результативности. Акцент на процессах в управлении проектами позволяет проводить сравнения между проектами и, таким образом, поддерживать непрерывные улучшения. Кроме того, представление процесса облегчает стратегическую увязку отдельных проектов с бизнес-контекстом организации и основными бизнес-целями. Таким образом, работа над улучшением процессов в управлении IT-проектами через призму эффективности и результативности должна быть сосредоточена на внутренних и внешних, а также краткосрочных и долгосрочных перспективах.

Если процессы могут быть идентифицированы в организации, основанной на проектах, и если концепции эффективности и результативности могут быть применены для оценки процессов, можно было бы оценить процесс управления проектами с течением времени с точки зрения производительности. Но для того, чтобы обсудить эффективность и результативность в управлении проектами, необходимо было выработать общую точку зрения между учеными и практиками.

Результатом совместной работы была выработка нескольких методов оценки эффективности проекта, которые можно применять на практике.

В целом существует даже классификация таких методов:

- 1) Методы статистические.
- 2) Методы, основанные на дисконтировании.

Статистические методы включают в себя:

- 1) Метод, основанный на расчете сроков окупаемости инвестиций.
- 2) Метод, основанный на определении нормы прибыли на капитал.
- 3) Метод, основанный на расчете разности между суммой доходов и инвестиционными издержками или единовременными затратами за весь срок использования инвестиционного проекта. Этот метод также иногда называется накопленным салдо денежного потока.
- 4) Метод сравнительной эффективности приведенных затрат на производство продукции.
- 5) Метод выбора вариантов капитальных вложений на основе сравнения массы прибыли или метод сравнения прибыли.

Перечисленные выше методы оценки эффективности проекта рационально применять при длительности проекта до 5 лет и при наличии равномерного распределения затрат по годам. При проведении анализа отрасли, в которой проект может занимать около 5-7 лет, статистические методы не подходят, так как появляется необходимость в учетывании фактора времени. Здесь нам больше подходят методы, которые входят во вторую группу, а именно:

- 1) Метод чистой приведенной стоимости или метод чистой дисконтированной стоимости или метод чистой текущей стоимости.
- 2) Метод внутренней нормы прибыли.
- 3) Дисконтированный срок окупаемости инвестиций.
- 4) Индекс доходности.

Кратко можно описать эти методы следующим образом.

Метод оценки эффективности инвестиционного проекта на основе чистой приведенной стоимости позволяет принять управленческое решение о целесообразности реализации проекта исходя из сравнения суммы будущих дисконтированных доходов с издержками, необходимыми для реализации проекта (капитальными вложениями).

Внутренняя норма прибыли представляет собой ту расчетную ставку процента (ставку дисконтирования), при которой сумма дисконтированных доходов за весь период реализации инвестиционного проекта становится равной сумме первоначальных затрат (инвестициям). Эту норму можно трактовать как максимальную ставку процент, под который фирма может взять кредит для финансирования проекта с помощью заемного капитала.

Дисконтированный срок окупаемости инвестиций - срок окупаемости инвестиций в текущих стоимостях.

Индекс доходности — это отношение приведенных денежных доходов к приведенным на начало реализации проекта инвестиционным расходам. Если индекс доходности больше 1, то проект принимается. При индексе доходности меньше 1 проект отклоняется.

Каждый из этих методов оценки эффективности проекта сводится к вычислениям математических формул, которые мы рассмотрим далее.

6.2. Математический подход к оценке эффективности проекта

Основным и, пожалуй, самым важным методом из второй группы считается метод **чистой приведенной стоимости**. Его основное достоинство состоит в том, что он позволяет сравнивать проекты с различной продолжительностью и разными капитальными вложениями.

Приведенная стоимость отсроченного дохода может быть рассчитана по следующей формуле:

$$(PV) = (DF) * C_1$$

где PV – приведенная стоимость, DF – коэффициент дисконтирования, C_1 – ожидаемый доход в период равный одному году.

Коэффициент дисконтирования – коэффициент, применяемый для дисконтирования, то есть приведения величины денежного потока на шаге t многошагового расчета эффективности инвестиционного проекта к моменту, называемому моментом приведения. Он определяется в результате деления единицы на норму доходности.

$$DF = \frac{1}{1 + r}$$

где r – это норма доходности.

Если подставить это выражение в формулу выше, мы получим.

$$(PV) = (DF) * C_1 = \frac{1}{1 + r} * C_1$$

А для того, чтобы вычислить чистую приведенную стоимость, необходимо увязать приведенную стоимость проекта с требуемыми на него инвестициями:

$$NPV = PV - \text{требуемые инвестиции}$$

Здесь NPV – это чистая приведенная стоимость, а PV – это просто приведенная стоимость.

Чистая приведенная стоимость, с другой стороны, может быть вычислена как сумма первоначальных вложений (инвестиций) и приведенной стоимости всех будущих денежных потоков проекта.

$$NVP = C_0 + \sum_{t=1}^n \frac{C_t}{(1 + r)^t}$$

где C_0 – это денежный поток нулевого периода (то есть стартовые, первоначальные инвестиции), C_t – инвестиции периода t , n – срок действия проекта, а r – ставка дисконтирования.

Ставка дисконтирования – это процентная ставка, используемая для пересчёта будущих потоков доходов в единую величину текущей стоимости.

Она определяется как переменная величина, зависящая от ряда факторов, например стоимость альтернативного вложения средств на данный период (например, ставка банковского процента по депозитам, ставка рефинансирования, средняя доходность уже имеющегося бизнеса и т. д.), оценка уровня инфляции на выбранный период, оценка стоимости риска обесценивания средств за период и так далее.

В большинстве примеров расчетов, ставка дисконтирования берется от 12% до 20%.

Справедливы следующие условные отношения для выбранных параметров NPV.

- 1) Если $NPV > 0$, то можно смело приступать к инвестированию в рассматриваемый проект;
- 2) Если $NPV=0$, то при наличии менее рискованных и энергоемких проектов, лучше отдать предпочтение им;
- 3) Если $NPV < 0$, то от такого проекта лучше отказаться и постараться найти более достойный объект вложения.

Стоит отметить, что представленные выше формулы актуальны лишь для расчета эффективности вложений на один год. В случае, если инвестиции будут осуществляться на период t , то формулы приобретают следующий вид.

$$PV = DF * C_1 = \sum \frac{1}{(1+r)^t} * C_t$$
$$NVP = C_0 + C_1/(1+r)$$

На основе формул формируется первое важное правило, которым необходимо руководствоваться при осуществлении инвестиционной деятельности: вкладывать деньги нужно только в проекты, которые имеют положительную чистую приведенную стоимость.

Второе правило распространяется на норму доходности, которая должна быть выше альтернативных издержек, направленных на реализацию проекта. Только в таком случае проект будет иметь свою целесообразность.

Альтернативным методом оценки эффективности проекта является метод внутренней нормы доходности или IRR.

Это такая норма дисконта, при которой приведенная стоимость планируемых проектных денежных поступлений равна приведенной стоимости связанных с проектом затрат. Этот показатель характеризует какое максимальное требование к годовому доходу на вложенные деньги инвестор может закладывать в свои расчеты, чтобы проект выглядел привлекательно.

Мы можем вычислить этот показатель по следующей формуле.

$$IRR = r1 + \frac{NPV * (R2 - R1)}{NPV1 - NPV2}$$

где $r1$ – это ставка дисконтирования, определенная методом подбора, которой соответствует положительное значение NPV1, $r2$ – это другая ставка дисконтирования, определенная методом подбора, которой соответствует положительное значение NPV2,

Также существует такой показатель как срока окупаемости проекта с учетом дисконтирования. Он рассчитывается как продолжительность периода до момента окупаемости с учетом дисконтирования. Моментом окупаемости с учетом дисконтирования считается более ранний момент времени в расчетном периоде, после которого накопленный чистый дисконтированный доход в дальнейшем остается неотрицательным.

Мы можем считать проект эффективным, если срок окупаемости с учетом дисконтирования существует и находится в пределах срока жизненного цикла проекта.

Доходность инвестиций, с другой стороны, рассчитывается как отношение прибыли к инвестициям.

$$\text{Доходность} = \text{Прибыль} / \text{Инвестиции}$$

Аналогичным образом можно рассчитать ожидаемую доходность компании и ожидаемую доходность проекта внутри компании, которые сопряжены с одинаковым риском. Затем сравнить полученные показатели и принять решение об объекте инвестирования. Данная формула актуальна для ситуации, которая зависит от общего состояния экономики и имеет оптимистичный, пессимистичный и наиболее вероятный прогнозы.

Подобные параметры могут оцениваться как в статике, так и в динамике.

Последний вариант наиболее реален, так как в ходе реализации проекта происходит движение денежных средств в виде поступлений и расходов, которое носит название «денежный поток». В случае, если исходные параметры для расчета эффективности проекта подчиняются определенному вероятностному закону, то можно применить вероятностный подход. Но зачастую в процессе моделирования реальных проектов статистика неактуальна или вовсе отсутствует, что не дает возможности его применять. Главная сложность заключается в наличие различных видов неопределенности, которые возникают в следствие отсутствия достаточного объема информации относительно неопределенных показателей проекта.

В этом случае нам помогают модели проекта.

6.3. Модель проекта

Модель проекта – это некоторое математическое «ядро», которое позволяет отслеживать состояние проекта в динамике и в условиях неопределенности.

Просчитав модель проекта с различными параметрами, мы можем определить возможные ситуации и поставить им в соответствие различные исходы.

Например, мы можем строить пессимистичный, оптимистичный и реалистичный прогнозы для проекта с разными параметрами. Можем строить «комфортный» прогноз и определять границы параметров, при нарушении которых, прогноз будет стремиться к оптимистичному или к пессимистичному.

Чему же стоит уделить особое внимание при расчете проекта.

В основном при оценке эффективности проектов в условиях неопределенности и риска уделяют внимание ставке дисконтирования, которая может быть реальной и номинальной. Если показатели проекта рассчитываются с учетом инфляции, то применяют номинальную ставку. Если меняется уровень риска, структура или стоимость капитала, необходимо

применять ставку дисконтирования по отдельности для каждого этапа проекта. Самым простым способом определения ставки дисконтирования, который чаще всего используется на практике – ее определение, исходя из требований инвестора, или экспертным путем.

Если проект создается в постоянных ценах, то необходимо номинальную ставку перевести в реальную по следующей формуле:

$$Dr = \frac{1 + dn}{1 + i}$$

где Dr – это реальная ставка дисконтирования, Dn – это номинальная ставка дисконтирования, i – это уровень инфляции.

Динамически меняя параметры у приведенных выше формул, мы можем построить модели развития IT-проекта по любому сценарию, причем в реальном времени просчитывать «будущее» проекта по разным сценариям.

Естественно, что процесс моделирования проекта и сам проект зависят от множества факторов.

6.4. Система факторов, влияющих на проект

IT-проект это сложная система, со своими внутренними взаимодействующими процессами и «контактами» с внешним миром. Поэтому на проект влияет как совокупность внутренних, так и внешних факторов.

Факторы внешнего окружения наименее прогнозируемые и среди них можно выделить следующие:

1) Политические факторы:

- a. политическая стабильность;
- b. поддержка проекта правительством; националистические проявления;
- c. уровень преступности;
- d. торговый баланс со странами-участниками;
- e. участие в военных союзах.

2) Экономические факторы:

- a. структура национального хозяйства;
- b. • виды ответственности и имущественные права, в том числе на землю;
- c. • тарифы и налоги;
- d. • страховые гарантии;
- e. • уровень инфляции и стабильность валюты;
- f. • развитость банковской системы;
- g. • источники инвестиций и капитальных вложений;
- h. • степень свободы предпринимательства и хозяйственной самостоятельности;
- i. • развитость рыночной инфраструктуры;
- j. • уровень цен;

k. • состояние рынка (сбыта, инвестиций, средств производства/ сырья и продуктов, рабочей силы и др.).

3) Общество.

- a. условия и уровень жизни;
- b. уровень образования;
- c. свобода перемещений, «въезд-выезд»;
- d. трудовое законодательство, запрещение забастовок;
- e. здравоохранение и медицина, условия отдыха;
- f. общественные организации, средства массовой информации;
- g. отношение местного населения к проекту.

4) Закон и право:

- a. права человека;
- b. права предпринимательства;
- c. права собственности;
- d. законы и нормативные акты о предоставлении гарантий и льгот. Наука и техника:
- e. уровень развития фундаментальных и прикладных наук;
- f. уровень информационных технологий и компьютеризации;
- g. уровень промышленных и производственных технологий;
- h. энергетические системы;
- i. транспортные системы;
- j. связь, коммуникации и др.

5) Природа и экологические факторы:

- a. естественно-климатические условия;
- b. природные ресурсы;
- c. законодательство в области окружающей среды;
- d. природные катастрофы.

Самое важное в плане управления проектами – это определить, какая группа факторов осуществляет наибольшее влияние.

Например, из поверхностного анализа таких факторов следует, что:

- 1) наибольшему влиянию внешнего окружения подвержены социальные и инвестиционные проекты, затем организационные, экономические, в меньшей степени — инновационные;
- 2) наибольшее влияние на проекты оказывают экономика, законы и право, затем культура, что несколько неожиданно, и только после этого политика и общество;
- 3) наименьшее влияние на проекты оказывают природа, экология и инфраструктура.

Тема № 8. Управление расписанием проекта. Риски проекта

8.1. Системы управления временем проекта

Планирование и управление расписанием в управлении проектами — это перечисление мероприятий, результатов и этапов в рамках проекта. Расписание также обычно включает запланированную дату начала и окончания, продолжительность и ресурсы, назначенные для каждого вида

деятельности. Эффективное планирование проекта является важнейшим компонентом успешного управления временем.

Когда люди обсуждают процессы построения расписания, они обычно имеют в виду первые шесть процессов управления временем:

- 1) Управление планами и расписаниями.
- 2) Определение проектной деятельности.
- 3) Последовательность действий.
- 4) Оценка ресурсов.
- 5) Оценка продолжительности.
- 6) Разработка графика проекта.

Как выполнять планирование в управлении проектами: существует три основных типа расписаний:

- 1) Основное расписание проекта: основное расписание, как правило, представляет собой упрощенный список задач с временной шкалой или календарем проекта.
- 2) График этапов или сводный график: Этот тип графика отслеживает основные этапы и ключевые результаты, но не все задачи, необходимые для завершения проекта.
- 3) Подробное расписание проекта: это наиболее детальное расписание проекта, поскольку оно определяет и отслеживает каждую проектную деятельность. Если у вас сложный, крупный или длительный проект, важно иметь подробный график проекта, чтобы помочь отслеживать все работы и события.

Наиболее распространенной формой представления расписания проекта является диаграмма Ганта. С его помощью можно показать и последовательность выполнения работ, и зависимость работ друг от друга.

Особо важна здесь как раз последняя деталь: часть работ требуют завершенности другой части работ, а еще какие-либо работы могут быть вообще блокирующими. В этом плане важно видеть и понимать критические точки в расписании проекта, а также отслеживать важные «вехи» и этапы.

Планирование и составление расписания проекта обеспечивает следующие преимущества:

- 1) Помогает отслеживать, сообщать и сообщать о прогрессе заказчику и сотрудникам.
- 2) Гарантирует, что все необходимые данные находятся на одном уровне представления, вместе с задачами, зависимостями и крайними сроками.
- 3) Освещает проблемы в выполнении задач и проблемы, связанные, например, с нехваткой ресурсов.
- 4) Определяет взаимосвязи задач и их последовательность.
- 5) Отслеживает прогресс и выявляет проблемы на ранней стадии.

Есть семь общепринятых советов по созданию четкого расписания проекта на основании процессов управления временем.

- 1) Получайте информацию от заинтересованных сторон: не создавайте свое расписание изолированно. Важно использовать свою команду и

других заинтересованных сторон для определения задач, ресурсов, зависимостей и продолжительности.

- 2) Просмотрите предыдущие или аналогичные проекты с подобными масштабами и требованиями, — это может помочь составить реалистичные оценки и убедиться, что вы не забыли ни о каких задачах.
- 3) Учитывайте риски, выявляйте и документируйте любые факторы, которые создают риск для соблюдения графика и расписания проекта. Это поможет вашим усилиям по управлению рисками.
- 4) Учитывайте любое нерабочее время: например, убедитесь, что отпуска и праздничные дни отражены в вашем расписании, чтобы вы не предполагали, что люди будут работать, когда они не работают.
- 5) Определите критический путь для вашего проекта: Определение критического пути вашего проекта позволяет вам расставлять приоритеты и распределять ресурсы для наиболее важных задач в проекте.
- 6) Запишите предположения о планировании: Запишите логику, лежащую в основе ваших прогнозов планирования. Например, если вы предполагаете, что выполнение задачи займет всего 10 часов, потому что у вас есть старший разработчик. Затем, если вы окажетесь в ситуации, когда доступен только младший разработчик, вы сможете понять и объяснить, почему это заняло в два раза больше времени, чем планировалось.
- 7) Включите основные этапы проекта в расписание. Основные этапы — это события или маркеры, обозначающие важный момент в вашем проекте. Они полезны для составления сводного графика, отчетности перед руководителями и раннего выявления проблем. Вот несколько важных примеров:
 - a. начало проекта;
 - b. утверждение проекта;
 - c. выполнение требований;
 - d. внедрение продукта;
 - e. закрытие проекта.

Управление временем проекта в западной культуре также тесно связано с понятием «хронологии». Чтобы представить план или рабочий процесс проекта визуально в виде классической «западной» хронологии, следует выполнить следующие действия:

- 1) Перечислите этапы проекта и установите чёткие сроки их выполнения.
- 2) Разбейте (декомпозируйте) проект на задачи с определёнными исполнителями.
- 3) Зафиксируйте даты начала и сроки выполнения, чтобы можно было понять, сколько времени займёт каждый этап.
- 4) Определите порядок работы над задачами.

- 5) Распределите приоритеты выполнения задач. (Какие задачи нужно выполнить в первую очередь? Есть ли такие задачи, к которым нельзя приступить, пока не будут завершены другие задачи? Что идёт последним? Ответив на эти вопросы, вы сможете понять, в каком порядке расставить задачи).
- 6) Поделитесь хронологией с командой и получите отзывы и предложения. Предоставьте хронологию проекта коллегам. Таким образом все смогут ознакомиться с планами проекта и будут знать, кто за что отвечает. Это также обеспечит вовлечённость заинтересованных лиц.
- 7) Отслеживайте ход выполнения проекта.
- 8) Контролируйте работу своего коллектива и следите за тем, чтобы всё делалось своевременно в соответствии с целями и сроками проекта. Отправляйте обновления статуса при достижении вех, а также держите коллег в курсе ведущейся работы и изменений хронологии.

8.2. Учет индивидуального времени работы сотрудников проекта

Как только работа над проектом начата, необходимо отслеживать время, затрачиваемое каждым разработчиком на выполнение задач.

Для этого существуют специальные методы и инструменты.

Только когда вы будете знать, что для вас важно, во время работы над проектом, вы сможете заранее расставить приоритеты. Чтобы расставить приоритеты в чем-то, вам придется научиться говорить «нет» чему-то другому.

Хотя это может показаться очень трудным навыком для освоения, помните, что так или иначе вы всегда говорите «нет» чему-то. Разница в том, что, когда ваши приоритеты предельно ясны, вы можете обдумать, чему сказать «нет», а чему говорить «да». В противном случае кто-то другой определит ваши приоритеты за вас, и вы будете пассивно принимать их и реагировать на чужие приоритеты, работая менее эффективно.

Для того, чтобы работать более комфортно, нужно уметь работать со своим расписанием, вести учет рабочего времени и правильно расставлять приоритеты.

Для каждого разработчика и участника команды в мировой практике существует несколько важных советов, как повысить эффективность затрачиваемого рабочего времени.

- 1) Составьте план, чтобы начать день. Подумайте о том, как вы начинаете свой день и что вы можете сделать, чтобы начать его в хорошем настроении и продуктивно. Просто убедитесь, что вы реалистичны и учитываете обязанности, которые у вас есть в это время суток. Это не должно усложнять работу, но должно сделать ее эффективнее.
- 2) Создайте расписание «особых» дней работы над проектом. На любом проекте есть ряд долгосрочных целей или общих задач. Выделение небольшого количества времени в определенный день каждую

неделю помогает гарантировать, что вы их достигнете. Это не позволит откладывать важные цели, о которых вам нужно позаботиться. Это также поможет вам составить лучшее расписание. Примером такого «особого дня» или нескольких дней может быть «день интеграционного тестирования», «день повторного ревью кода» и так далее. Более того, в больших компаниях принято изначально учитывать такие моменты в расписании проекта и воспринимать их как фазы жизненного цикла. Например: 3 дня пишем код, 1 день проводим повторное ревью, 1 день делаем рефакторинг кода.

- 3) Делегируйте обязанности. Это трудно для большинства людей, но это очень важно. Вы знаете, что у вас много обязанностей, и если вам немного помогут с делами, вы сможете лучше распоряжаться своим временем. Просто будьте внимательны к тому, кому вы делегируете задачи. Вы должны быть уверены, что они будут выполнены. Этот подход особенно эффективен, когда вам достается задача, для выполнения которой больше подходит другой специалист, - можно делегировать часть обязанностей ему.
- 4) Расставляйте приоритеты в своих задачах. Знайте, что нужно делать каждый день, и назначайте каждой задаче определенный уровень важности. Это поможет убедиться, что вы делаете самые важные вещи в первую очередь, чтобы они были выполнены вовремя. Это также помогает убедиться, что у вас достаточно времени для выполнения самых важных задач. Многие разработчики стараются сначала выполнить более простые или интересные задачи, откладывая сложные, рутинные или непонятные задачи «на потом», — это «порочная» практика.
- 5) Контролируйте свои перерывы. Перерывы - самый «болезненный» фактор, из-за которого люди не могут управлять своим временем. Думайте о вещах, которые отвлекают и прерывают вас каждый день, и работайте над тем, чтобы держать их под контролем. Это может означать такие вещи, как выключение телефона в определенные моменты дня, чтобы сосредоточиться на своих задачах.
- 6) Создайте несколько систем поддержки. Каждый мог бы использовать инструменты, которые помогают ему двигаться в верном направлении. Вам нужно все организовать и найти способ, чтобы ваше расписание было на первом плане. Эти системы помогут вам оставаться организованными, чтобы ваши задачи выполнялись по установленному вами графику.
- 7) Учитесь лучше общаться. Умение эффективно общаться экономит вам много времени. Даже на устранение простейших недоразумений могут уйти часы, а иногда и дни. Работайте над тем, чтобы ваше общение было прямым, чтобы ничего не потерялось в беспорядке.
- 8) Следите за своими контактами. Будут моменты, когда вам нужно быстро связаться с кем-то, и вы не хотите тратить время на поиск их

контактной информации. Создайте систему для поддержания всех ваших контактов в порядке и доступности.

- 9) Работайте в команде. Если командная работа позволит выполнить задачу быстрее, воспользуйтесь этим преимуществом. Просто убедитесь, что работа правильно распределена и что все движутся «в одном направлении».
- 10) Используйте Технологии, экономящие время. Есть ряд вещей, которые могут помочь вам оставаться организованными и идти по намеченному пути. Ознакомьтесь с различными доступными приложениями и программами и выберите несколько, которые улучшат ваши навыки управления временем. Мы будем рассматривать такие инструменты дальше в наших лекционном и практическом курсах.

Теперь давайте рассмотрим какими инструментами сейчас пользуются для учета личного рабочего времени сотрудники компаний и индивидуальные разработчики.

Основная функциональность любой системы отслеживания времени обычно довольно последовательна по всем направлениям. Приложения для отслеживания времени позволяют записывать рабочее время с помощью таймера, который вы можете запускать и останавливать. Вы также можете вносить незначительные изменения вручную в журналы времени — это позволяет обновлять записи о времени, когда вы забываете запустить или остановить таймер. Также системы могут выполнять отслеживание времени и автоматически исключит случаи, когда ваш компьютер простаивает или неактивен в течение определенного промежутка времени.

Лучшие приложения для отслеживания времени обладают большей функциональностью и специализированными функциями, которые облегчают управление вашей работой. Ниже приведены некоторые функции, которые важно учитывать при выборе лучшего приложения в данной области.

- 1) Стоимость. найдите систему, соответствующую вашему бюджету, даже если этот бюджет ограничивает вас бесплатными приложениями. Нет причин платить за дополнительные функции, которые вам не нужны и которыми вы не будете пользоваться. Кроме того, многие системы в этом списке предлагают бесплатную опцию или бесплатный пробный период.
- 2) Простота использования и быстрое освоение: если вы новичок в отслеживании времени, крайне важно найти систему, которая была бы интуитивно понятной в использовании и простой в освоении.
- 3) Выставление счетов и бухгалтерский учет. Наиболее распространенная причина, по которой, например, фрилансер должен отслеживать рабочее время, — это создание счетов для клиентов и получение оплаты. Отслеживание рабочего времени позволяет точно выставить счета клиентам за количество времени, затраченного на их проекты. Некоторые системы отслеживания времени предлагают

функции, связанные с бухгалтерским учетом, такие как отслеживание расходов.

- 4) Данные, отчетность и отслеживание. Когда вы знаете, как тратится ваше время, вы можете проанализировать свои привычки, чтобы принимать более разумные бизнес-решения. Кроме того, при отслеживании времени вы можете точно выставить счета по проектам, и в будущем вы будете знать количество времени, необходимое для аналогичных проектов. Некоторые системы предлагают подробный мониторинг, который позволяет вам разбивать почасовую оплату с помощью снимков экрана с отметками времени и расширенного отслеживания использования приложений.
- 5) Гибкость. Некоторые системы могут работать только в сети Интернет, в то время как другие предоставляют программное обеспечение для настольных и мобильных устройств. Независимо от платформы, гибкость вашей системы должна соответствовать вашим потребностям. Вы должны иметь возможность отслеживать свое время, даже если это означает работу с разных устройств.
- 6) Управление проектами. Эффективное управление проектами и отслеживание времени идут рука об руку. Вот почему некоторые системы позволяют вам управлять своими проектами в приложении для отслеживания времени. Некоторые даже имеют возможность интегрироваться со сторонними системами управления проектами.
- 7) Командное сотрудничество и коммуникация. Нужно ли вам делиться работой в команде? Нужно ли вам сотрудничать с другими людьми для реализации проектов? Как вы учитываете отзывы клиентов и получаете обратную связь на протяжении всего вашего рабочего процесса? Эти факторы следует учитывать при принятии решения о том, какое программное обеспечение для отслеживания времени будет наиболее эффективным для вашего бизнеса.
- 8) Управление несколькими сотрудниками, клиентами и контрактами. Так приложения для отслеживания времени могут помочь вам определить, где команды тратят свое время, чтобы определить, есть ли ненужные узкие места или дублирование. Если у вас много активных контрактов или несколько клиентов, вам может понадобиться система, которая может классифицировать ваше время. Система также должна связать его с нужным проектом по умолчанию.

Приведем дальше список самых популярных современных решений в этой области.

- 1) Toggle.
- 2) Freshbooks.
- 3) RescueTime.
- 4) Harvest.
- 5) Timely by Memory.
- 6) Timing.

- 7) Everhout.
- 8) Timeneye.
- 9) Tmetric.
- 10) Hubstaff.
- 11) Upwork time tracker.

Мы рассмотрим некоторые приложения более подробно на практической части нашего курса.

8.3. Потокосые диаграммы для развития проекта

Потоковые диаграммы или диаграммы потоков данных служат для представления процесса развития IT-проекта в динамике.

Подобный тип диаграмм развивался еще с 1920-х гг. инженерами и экспертами на производствах различного характера.

Тогда было две основные формы диаграммного представления процессов (они также актуальны и сейчас):

- 1) Диаграмма бизнес-процессов.
- 2) Диаграмма потоков данных.

Диаграмма бизнес-процессов (Business Process Diagrams или Business Process Mapping) занимается отображением бизнес-процессов и детализирует шаги, которые предпринимает бизнес для завершения процесса. Например, разработки дизайна, прототипа приложения и так далее. Они показывают «кто», «что», «когда», «где» и «как» для этих шагов и помогают проанализировать «почему». Эти карты также называются диаграммами бизнес-процессов и диаграммами бизнес-процессов. Как и другие типы диаграмм, на этих картах используются определенные символы, такие как круги, прямоугольники, ромбы и стрелки, для отображения деловой активности.

Отображение бизнес-процессов можно использовать для документирования текущего процесса и моделирования нового. Его цель состоит в том, чтобы получить подробное представление о процессе, людях, входных данных, средствах управления и результатах, а затем потенциально упростить все это, сделать его более эффективным и/или улучшить результаты процесса. Для проведения такого картирования требуется время и дисциплина, но со временем отдача может быть значительной. Построение таких диаграмм стало обычным явлением в деловом мире для стандартизации процедур, повышения эффективности, соответствия требованиям аудита и получения конкурентных преимуществ.

Диаграмма потоков данных (Data Flow Diagram) отображает поток информации для любого процесса или системы. Он использует определенные символы, такие как прямоугольники, круги и стрелки, а также короткие текстовые метки, для отображения входных и выходных данных, точек хранения и маршрутов между каждым пунктом назначения. Блок-схемы данных могут варьироваться от простых, даже нарисованных от руки обзоров процессов до подробных многоуровневых DFD, которые постепенно углубляются в то, как обрабатываются данные. Они могут быть использованы

для анализа существующей системы или моделирования новой. Как и все подобные диаграммы, DFD часто может визуально «показать» вещи, которые было бы трудно объяснить словами, и это работает как для технической, так и для нетехнической аудитории.

Вот почему DFDS остаются такими популярными даже сейчас. Хотя они хорошо работают для программного обеспечения и систем для передачи данных, в настоящее время они менее применимы для визуализации интерактивного программного обеспечения или систем, ориентированных на работу в реальном времени или базы данных.

Цель и преимущества потоковых диаграмм различной направленности.

- 1) Документировать и визуализировать процесс для лучшего понимания, контроля качества и обучения сотрудников.
- 2) Стандартизировать процесс и его представление для обеспечения оптимальной эффективности и повторяемости.
- 3) Изучить процесс для повышения эффективности и совершенствования. Это помогает показать ненужные шаги, узкие места и другие недостатки.
- 4) Смоделировать лучший процесс или создать совершенно новый процесс.
- 5) Для общения и совместной работы с диаграммами, которые соответствуют различным ролям в организации или за ее пределами.

В общем случае потоковые диаграммы содержат некоторый «словарь» графических элементов, составляющий базу для создания формы представления.

Обычно такой «словарь» состоит из следующих элементов.

- 1) Основной элемент – крупная инвентарная единица, объект или понятие.
- 2) Связи и направления передачи данных. Показывают направление перемещения крупных потоков информации.
- 3) Направление технологического потока.
- 4) Регулирующие клапаны и затворы, интерфейсы и буферные элементы.
- 5) Запасные и рециркуляционные системы.
- 6) Эксплуатационные данные. Значения часто включают минимальное, нормальное и максимальное значение.
- 7) Состав и структура информации.
- 8) Имена элементов, потоков и процессов.
- 9) Места соединения с другими системами.

Как видно из такого перечня, при помощи такого набора элементов можно описать как техническую систему, так и механическую. А еще их применяют для описания процессов и программных систем, также состоящих из потоков данных, ключевых блоков, подсистем, программных агентов, серверов, клиентов, библиотек и так далее.

В отечественной «нотации» такие схемы могут также соответствовать различным ГОСТ на оформление документации по проекту и выражаться,

например, как «схема электрическая структурная» или «схема электрическая функциональная», и даже «структура базы данных».

8.4. Диаграммы Ганнта

Это еще один пример структурного представления информации о проекте с точки зрения его времени исполнения и совокупности работ.

Другими словами, диаграмма Ганнта, - это ленточная диаграмма, график Ганта, календарный график. Это популярный формат столбчатых диаграмм (или гистограмм), который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов.

Часто диаграмма Ганнта используется в приложениях по управлению проектами и является одним из основных инструментов для контроля и мониторинга IT-проектов.

По своей сути, диаграмма Ганнта представляет совокупность полос, ориентированных вдоль оси времени, слева на право. Начало и конец каждой полосы представляют момент времени начала и завершения работы, а длина полосы отражает длительность работы.

Соответственно, по оси X на диаграмме Ганнта откладываются моменты времени, а по оси Y откладываются работы или задачи по проекту.

Кроме того, на диаграмме могут быть отражены совокупные задачи, процентные метрики завершения, различные указатели, последовательности, зависимости работы, метки, вежи проекта и так далее.

Для регулирования течения проекта есть несколько ключевых понятий, например «веха». Веха – это метка одного из наиболее значимых переходов между группами задач. Например, вехой может быть момент синхронизации работ, просто календарная дата релиза, допуски и так далее.

Сдвиг вехи может привести к сдвигу всего проекта, так что диаграмма Ганнта может считаться весьма приближенным календарным планом или графиком работ, но не является чистым календарным планом.

Единственное, что классическая диаграмма Ганнта не отражает, это значимость или ресурсоемкость работы, а также не отображает сущность работы. Хотя она может быть модифицирована для того, чтобы, при помощи дополнительных связей провести соответствие какой-либо работы с ресурсом. Ну или же мы можем при помощи цветовой дифференциации полос диаграммы отмечать важность или приоритет работ.

Для больших проектов диаграммы Ганнта могут быть очень тяжеловесными и громоздкими, что затрудняет их применение на практике, - они теряют наглядность.

Но, несмотря на недостатки, диаграммы Ганнта все еще являются стандартом де-факто в теории управления проектами и применяются, по крайней мере, для отображения структуры перечня работ.

Есть несколько общих рекомендаций для составления диаграмм Ганнта.

- 1) Максимально детализируйте проект на этапе начальной оценки.
- 2) Установить актуальный производственный календарь.

- 3) Проверить все исключения.
- 4) Сформировать все необходимые этапы проекта.
- 5) Распределить деятельность по видам работ и по этапам.
- 6) Занести все данные и виды деятельности на диаграмму.
- 7) Составить карту рисков проекта.
- 8) Занести на диаграмму все контрольные точки по представлению информации от владельца бизнеса или заказчика.
- 9) Занести все требуемые для выполнения проекта ресурсы.
- 10) Занести информацию об исполнителях проекта.
- 11) Отметить критические пути проекта.

При этом следует помнить, что диаграмма Ганнта – это всего лишь средство. Она применяется совместно с дорожной картой проекта и другими дополнительными инструментами.

Давайте далее поговорим о диаграмме сгорания.

8.5. Диаграммы сгорания

Диаграмма сгорания или Burndown chart применяется как графический инструмент представления оставшейся работы в зависимости от времени.

Невыполненная работа (или отставание) часто находится на вертикальной оси, а время выполнения - на горизонтальной, так что диаграмма сгорания — это по своей сути график выполнения работы.

Этот инструмент полезен для прогнозирования того, когда вся работа на проект или один конкретный период будет завершена. Он часто используется в гибких методологиях разработки программного обеспечения, таких как Scrum. Однако диаграммы выгорания могут быть применены к любому проекту, содержащему измеримый прогресс с течением времени.

В основе построения диаграмм сгорания лежит линейное представление оставшихся работ слева на право по убыванию. Так что левая крайняя точка основной линии соответствует максимальному числу задач и начальному (нулевому) времени, а крайняя правая точка соответствует нулевому числу оставшихся задач и крайней точки по времени (момент закрытия работы).

После формирования этой основной линии необходимо в ходе работы отмечать каждую завершенную задачу по его метке времени, так что точки второй линии будут лежать ниже или выше основной линии, как бы аппроксимируя ее с определенной точностью.

Если вторая линия лежит ниже первой, то проект идет на опережение. Если вторая линия лежит выше первой линии, то проект идет с опозданием.

По соотношению линий также считается эффективность работы команды. Например, нам по оценкам, предстоит выполнить проект за 28 дней работы, и над проектом работают два разработчика, которые работают с эффективностью 70%. Следовательно, работа должна быть завершена за $(28 \div 2) \div 0,7 = 20$ дней, то есть пойти на опережение.

Конечно, первая линия может быть названа «идеальной», поскольку, как только она пересечет ось X, работ по проекту не останется и проект будет завершен. Однако многие специалисты говорят, что следовать этому правилу

неверно. Эта линия представляет собой математический расчет, основанный на оценках, и оценки с большей вероятностью будут ошибочными для реальных работ.

Цель диаграммы выгорания состоит в том, чтобы отобразить прогресс на пути к завершению и дать оценку вероятности своевременного завершения.

В этой ситуации как раз-таки вторая линия показывает фактическую оставшуюся работу. В начальной точке фактическая оставшаяся работа совпадает с идеальной оставшейся работой, но с течением времени фактическая линия работы колеблется выше и ниже идеальной линии в зависимости от этого несоответствия между оценками и тем, насколько эффективна команда. В общем, каждый день проекта в эту строку добавляется новый пункт. Каждый день сумма оценок времени или точки истории для недавно выполненной работы вычитается из последней точки в строке для определения следующей точки.

Одна из проблем, которая может быть замечена в графиках выгорания, заключается в том, где конкретно может находиться вторая(рабочая) линия: выше или ниже первой (идеальной) рабочей линии.

Это во многом зависит от того, насколько были точны первоначальные оценки времени:

- 1) Если команда постоянно переоценивает временные требования, прогресс всегда будет появляться с опережением графика.
- 2) Если они постоянно недооценивают временные требования, они всегда будут отставать от графика.

Эта проблема может быть частично устранена путем включения коэффициента полезного действия в диаграмму выгорания. После первой итерации проекта коэффициент эффективности может быть пересчитан, чтобы обеспечить более точные оценки во время следующей итерации. Некоторые шаблоны автоматически рассчитывают эффективность по мере продвижения проекта. Это может быть использовано для определения областей / этапов, где постоянно возникают неточные оценки.

Также следует отметить, что существуют разные варианты построения и представления подобных диаграмм, например связанные с решением задач, где линии идут слева снизу – вверх направо, и так далее.

В любом случае, диаграммы сгорания также являются инструментом динамического отслеживания IT-проекта, но иногда следует «остановиться» и посмотреть на детали проекта более детально. Этому помогает принцип «среза» проекта.

8.6. Принцип «среза» проекта

Срез проекта – это одна из форм отчета, выполненного за определенный период времени и представленного в виде отдельного проекта. В срезе проекта отображены все данные и представления проекта на текущий момент, а также данные по расходам стоимостных составляющих и материалам, длительности, объему работ. В этот момент также автоматически выполняется расчет затрат за заданный период времени. На диаграмме Гантта, например, срезы

представлены в виде подмножества тех операций, которые были запланированы или выполнялись в этот период.

На основе среза проекта можно выполнить переоценку времени и стоимости проекта, также ориентируясь на дополнительные инструменты типа диаграмм сгорания, календарный план, план работ и так далее.

8.7. Методики динамического переоценивания времени проекта

Ценность проекта и его ожидаемый эффект со временем меняются. Это связано, по крайней мере, с тремя обстоятельствами:

- 1) С поступлением новой информации о проекте и соответственно с уточнением его базовых характеристик.
- 2) С поступлением новой информации о состоянии финансовых и товарных рынков, что позволяет уточнить уровни доходности, рыночные прогнозы объемов продаж, цен, тенденции в динамике спроса и предложения на продукт проекта и факторы производства;
- 3) С течением времени. Смещения на оси времени точек, в которых находятся денежные потоки, приводит к изменению результатов оценки. Уже пройденные стадии проекта, сделанные ранее затраты и полученные доходы становятся нерелевантными для оценщика. Его в большинстве случаев не интересуют ранее достигнутые успехи или понесенные потери. Его взгляд устремлен в будущее. Поэтому ценность проекта со временем меняется.

И все же оценщику бывает целесообразно производить расчет эффекта проекта в приведении не только к текущему моменту времени, но и к моменту начала проекта. Это позволяет выяснить, насколько точно сбываются прогнозы, используемые при принятии инвестиционных решений, и контролировать, как меняются значения критериев оценки проекта после его запуска.

Если характеристики большинства проектов компании меняются в сторону ухудшения по мере исполнения этих проектов, что бывает довольно часто, можно сделать вывод о том, что прогнозы, лежащие в основе финансовой модели, имеют смещение, или систематическую ошибку. Соответственно в будущем это надо иметь в виду и внести необходимые корректировки в последующие обоснования проектов.

Приведение денежных потоков к текущему моменту времени помимо контроля экономической ценности можно использовать для принятия решения о том, надо ли продолжить проект или целесообразно от него отказаться.

Если сумма ликвидационной стоимости существенно превышает ценность действующего проекта, приведенную ценность его денежных потоков, то стоит задуматься, есть ли смысл его продолжать.

В эти моменты времени менеджер проекта, играющий роль оценщика может воспользоваться описываемыми методиками для того, чтобы «пересчитать формулы» и заново взглянуть на проводимые расчеты.

Также можно отобразить получаемые результаты на графиках, расположив внизу ось времени, а по оси У значения параметра. Это позволит отслеживать оценки проекта в динамике.

МОДУЛЬ 3. ПРАКТИЧЕСКИЕ АСПЕКТЫ УПРАВЛЕНИЯ IT-ПРОЕКТАМИ

Тема № 9. Оценка стоимости проекта по различным критериям

9.1. Мировая практика в оценке стоимости проекта

Оценка стоимости — процесс определения всех затрат, необходимых для успешной и полной реализации проекта.

Оценка стоимости — итеративный процесс получения примерных данных о стоимости работ и ресурсов. Оценки могут уточняться по ходу проекта. Допустимая погрешность оценок зависит от назначения получаемых данных и от фазы проекта.

Таким образом, любая оценка стоимости в проекте приблизительная. Но все же приблизительно — это от слова «близко». По ходу проекта оценки должны уточняться и становиться более реальными.

Рассмотрим далее типы оценок стоимости проекта.

- 1) грубый порядок величины — стоимостные ожидания проекта, находящегося на фазе замысла или идеи;
- 2) порядок величины — предположения стоимости проекта, рассчитанные в бизнес-плане или аналогичном документе;
- 3) бюджетная оценка — оценка стоимости проекта, полученная на основе данных, предоставленных поставщиками и исполнителями работ;
- 4) точная — оценка стоимости, включаемая в бюджет при определении окончательной плановой стоимости проекта перед переходом к фазе реализации.

Для любых вариантов оценок есть диапазоны погрешностей, а в качестве исходных данных для оценки стоимости менеджеру проекта необходима информация о содержании проекта. На первоначальных этапах это может быть Устав проекта, а на последующих потребуются его подробный календарный план. Все зависит от номера итерации по оценке стоимости. Чтобы получить оценку грубого порядка величины стоимости проекта, иногда достаточно лишь идеи и замысла проекта. Для более точных оценок требуется более точная информация.

Также оценка стоимости зависит от многих составляющих, например:

- 1) материалы и комплектующие;
- 2) закупаемое оборудование, транспорт;
- 3) арендные платежи (площади, оборудование, транспорт);
- 4) затраты на лизинг (покупка, взятие в аренду, лизинг);
- 5) производственные мощности;
- 6) стоимость труда персонала;
- 7) затраты на расходные материалы;

- 8) затраты на обучение и стажировки;
- 9) затраты на проведение мероприятий (конференции, семинары);
- 10) командировочные расходы;
- 11) затраты на логистику;
- 12) представительские расходы.

По своей сути выделяют следующие методы оценки стоимости проекта:

- 1) Параметрическая оценка — метод, при котором для стоимостной оценки используется статистическая зависимость между стоимостью операции и другими переменными (параметрами), полученная на основе анализа исторических данных (например, величина площади конструкции в строительстве, число строк в коде программы, количество часов рабочего времени). Опытным путем рассчитывается стоимость одной единицы объема работ. При расчете стоимости используются различные формулы, а для расчета стоимости полного объема работ исходят из стоимости отдельной единицы объема работ.
- 2) Оценка по аналогам — метод оценки стоимости по аналогии со сходными работами, выполнявшимися в этом или других проектах. Метод оценки по аналогам может относиться ко всему пакету работ целиком или использоваться в комплексе с параметрической оценкой, когда имеется информация о выполнении аналогичных работ, но другого объема или в других условиях. Достоинством метода является возможность получить более точную оценку. Причина тому — наличие информации не только о плановой стоимости анализируемых работ, но и о ее фактической стоимости. Разница в плановой оценке и в фактической стоимости может дать менеджеру проекта дополнительную информацию для размышления.
- 3) Оценка «снизу вверх» — технология оценки больших объемов работ суммированием оценок, полученных для более мелких составляющих данной работы. Чем более подробно и точно разработана ИСР проекта, тем точнее и корректнее могут быть получены стоимостные оценки по проекту. Метод «снизу вверх» по праву считается одним из самых точных.
- 4) Метод оценки «сверху вниз» считается значительно менее точным по сравнению с методом «снизу вверх». Он применяется в условиях отсутствия детальной ИСР, нехватки информации о ресурсах и материалах, необходимых для реализации работ. Технология оценки предполагает ровно обратные шаги по отношению к методу «снизу вверх». Сначала дается укрупненная оценка всего пакета работ, а затем она детализируется и декомпозируется на отдельные элементы (по работам, исполнителям и др.). Метод имеет право на жизнь на ранних этапах проекта, когда выполняется оценка его жизнеспособности и непонятно, следует ли расходовать ресурсы на более детальное планирование и оценку.

- 5) Анализ предложений исполнителей — очень простой метод при условии наличия исполнителей и подрядных организаций, желающих выполнить данный объем работ. Техническое задание, тендерная или иная документация рассылается по исполнителям-претендентам с просьбой предоставить свои оценки стоимости (а зачастую — и продолжительности) выполнения данных работ.

Оценка стоимости проекта также должна выполняться с соответствующим качеством и уровнем достоверности. Для повышения качества этого процесса рекомендуется придерживаться следующих принципов.

- 1) Принцип оптимального ответственного — лучше всех оценит стоимость задачи тот, кто ее лучше всех понимает. Часто таким человеком является непосредственный исполнитель задачи. Его оценки наиболее точны, они обоснованы его опытом и экспертными знаниями. Кроме того, привлекая исполнителей к процессу планирования, менеджер проекта тем самым повышает их мотивацию и ответственность за результат при выполнении задания.
- 2) Принцип независимости — оценку стоимости операций и работ необходимо вести независимо от оценок связанных с ними работ. Каждая работа рассматривается как независимая от других работ. Взаимосвязь между работами, связанные риски и сопутствующие этому отклонения будут учтены при агрегировании полученной информации на более высоком уровне планирования.
- 3) Принцип адекватности условий — при оценке и расчетах работ эксперт должен руководствоваться предположением, что у него адекватные условия реализации, достаточное количество ресурсов, ему доступны эффективные методы выполнения работ. Конечно же, жизнь сложнее предположений о ней, поэтому расчеты, полученные таким способом, грешат излишним оптимизмом. Для получения более точных оценок эксперт в своих предположениях должен адекватно учитывать ограничения проекта. Скорее всего, это приведет к увеличению оценок, но оно будет обоснованное и адекватное.
- 4) Принцип признания наличия рисков — в оценках, вводимых в плановые документы, должны быть учтены непредвиденные обстоятельства и риски, которые могут повлиять на стоимость и сроки работ. Абсолютно нормально вводить в оценку стоимости рисковые резервы. Однако их величина должна быть результатом адекватного, осмысленного анализа и расчета. Введение в стоимость работы резерва «на всякий случай» чаще всего является неэффективным. Он превращается в элементарную «перестраховку», планирование «на всякий пожарный», не обоснованное ни объективными обстоятельствами, ни реальной ситуацией в проекте.
- 5) Принцип права на ошибку — любая оценка есть предположение. Любое предположение содержит погрешность. Задача эксперта —

сделать эту погрешность минимальной. Введение наказаний за ошибку, санкций за любое отклонение реальных данных от прогнозных приведет к «перезакладам» и «перестраховкам». Исполнители и эксперты будут предусматривать необоснованные резервы, чтобы избежать наказания. Все оценки будут содержать исключительно пессимистические значения. При этом надежды на то, что при таких условиях фактические и плановые затраты будут совпадать или появится экономия, практически никакой. Доверие экспертов значительно повысит точность оценок.

Процесс оценки стоимости проекта должен быть выражен в виде **сметы**. Смета является документом, получаемым как результат процесса структуризации и систематизации стоимостных оценок проекта. По своему содержанию, смета проекта содержит список затрат, полученных на основе объема работ, ресурсов и цен, структурированных по отдельным статьям.

Следует напомнить, что затраты могут быть разными:

- 1) прямые (расходы);
- 2) накладные (косвенные);
- 3) общие и административные накладные расходы.

Сметы классифицируются следующим образом:

- 1) локальные;
- 2) объектные;
- 3) сметы на отдельные виды затрат;
- 4) сводные сметы (сводный сметный расчет).

Для расчета смет можно привести следующее содержание:

- 1) прямые затраты:
 - a. заработная плата;
 - b. стоимость материалов;
 - c. стоимость оборудования;
 - d. стоимость комплектующих;
 - e. эксплуатационные расходы;
- 2) накладные расходы:
 - a. административные;
 - b. хозяйственные;
 - c. расходы на управление и организацию работ;
 - d. расходы на обслуживание сотрудников проекта;
 - e. командировочные расходы;
 - f. представительские расходы;
 - g. транспортные расходы;
 - h. расходы на связь;
- 3) сметная прибыль:
 - a. чистый доход подрядчика, например, как процент от суммы прямых затрат и накладных расходов;
- 4) сметная стоимость:
 - a. сумма прямых затрат, накладных расходов и сметной прибыли.

Также бывают другие виды смет. Например, объектная смета.

Объектная смета — документ, содержащий расчеты и оценки стоимости по объекту (объектам) в целом в базисных или текущих ценах.

- 1) Цена базисная — цена товара стандартного качества, на основе которой устанавливается цена товара более высокого и низкого качества, например в случае, когда свойства фактически поставленного товара отличаются от оговоренных в контракте.
- 2) Цена текущая — цена или тариф, действующие в данный период времени (могут быть оптовые, закупочные, розничные, цены и расценки в строительстве, тарифы и цены на услуги, оказанные предприятиям, организациям, населению).

Еще один пример сметной документации – это **сметы** на отдельные виды затрат, то есть документы, содержащие расчеты и оценки стоимости по затратам, не учтенные сметными нормативами.

К отдельным видам затрат можно отнести:

- 1) премирование за досрочное завершение проекта;
- 2) оплату консультационных, аудиторских услуг;
- 3) выплаты льгот и компенсаций;
- 4) оплату непредвиденных командировок, деловых поездок;
- 5) оплату транспорта для сотрудников (доставку к месту работы);
- 6) расходы на рекламу;
- 7) страховые взносы на добровольное страхование;
- 8) оплату услуг операторов мобильной связи, интернет-провайдеров;
- 9) другое.

Также для управления стоимостью проекта используется сводный сметный расчет.

На самом деле, это основной документ для определения стоимости проекта, при помощи которого происходит обобщение данных из локальных и объектных смет и смет на отдельные виды затрат. При этом расчет производится как в базисных, так и в текущих ценах. Также возможно применение прогнозных цен на будущие периоды.

В сводном сметном расчете происходит суммирование и сведение воедино данных локальных и объектных смет до уровня всего проекта. В итоговый сметный расчет включаются данные смет на отдельные виды затрат.

К **сводному сметному расчету** (сводной смете) обычно прилагается пояснительная записка, которая содержит сопутствующую информацию, необходимую для понимания документа и облегчения работы с ним.

При всем этом сметные расчеты могут делиться на:

- 1) предварительные (для определения порядка величины предполагаемых затрат в начале проекта).
- 2) первичные (для сравнения планируемых затрат на проект с возможными ограничениями).
- 3) факторные (аналогичны первичным);
- 4) приближенные (для принятия окончательного решения по поводу инвестиций, запуске или отказе от проекта);

5) сводные (предназначены для окончательной фиксации стоимости проекта).

На основе сметной документации и по совокупности финансовых документов выполняется бюджетирование проекта.

По определению, бюджет проекта – это обособленный документ, разработка которого служит отправной точкой для финансирования.

Разработка абсолютно точного, полного и реального бюджета практически невозможно с первой попытки. Однако от указанных характеристик полноты, связности и отношения к реальной ситуации во многом зависит успех проекта.

Бюджет проекта может уточняться по мере работы над проектом.

В зависимости от этапа проекта специалисты выделяют несколько видов бюджета.

Процесс разработки бюджета проекта называется **бюджетированием**.

Бюджетирование – это процесс структуризации расходов проекта согласно плану счетов стоимостного учета конкретного проекта.

Структурировать бюджет можно по следующим критериям:

- 1) по видам работ;
- 2) статьям затрат;
- 3) отчетным периодам;
- 4) рискам;
- 5) структуре.

Можно также рассматривать бюджетирование как работу над планированием стоимости. Результат бюджетирования или план затрат является ответом на вопросы: когда, сколько и на что будут расходоваться денежные средства.

Для бюджетов разных видов также существует наиболее удобное время для разработки и допустимые погрешности.

- 1) Разработка концепции проекта (предварительное планирование, допустимая погрешность 25 % - 40%).
- 2) Обоснование инвестиций проекта (обоснование статей затрат и привлечения инвестиций, допустимая погрешность 15 % - 20%).
- 3) Тендеры и заключение договоров (планирование расходов и расчетов с подрядчиками и субподрядчиками, допустимая погрешность до 10%).
- 4) Разработка рабочей документации (привлечение и использование ресурсов, допустимая погрешность 5% - 8%).
- 5) Реализация проекта (учет и контроль текущих расходов, контроль показателей проекта, мониторинг ресурсов, допустимая погрешность до 5%).
- 6) Завершение проекта (фактически законченный бюджет, анализ и архивирование проекта, анализ фактической стоимости).

Бюджет проекта нужно уметь рассчитывать с учетом специфики проекта и внешних факторов.

Например, в случае, когда проект не предусматривает постинвестиционной фазы, т. е. фазы получения прибыли, доходная часть бюджета не разрабатывается. Если же разрабатывается бюджет затрат, то при наличии корректных смет процесс превращения сметы в бюджет становится технической функцией. Смета — документ, который отвечает на вопросы, сколько и за что предстоит заплатить в проекте. Бюджет к этим вопросам добавляет ответ на вопрос, когда это будет.

Если доходная часть предусмотрена, она разрабатывается на основе данных маркетологов о планируемых объемах продаж продукта проекта и о прогнозных ценах.

Смета накладывается на ось времени, и все платежи распределяются по времени. При использовании информационных систем бюджетирования и назначении затрат конкретной работе система все остальное делает автоматически.

Бюджет может быть представлен в различном виде. Наиболее распространена форма представления бюджета в виде таблицы. Иногда для наглядности используются гистограммы или круговые диаграммы.

Все перечисленные детали сводятся к тому, что нужно учитывать все возможные критерии и особенности расчета стоимости IT-проекта.

9.2. Критерии и особенности расчета стоимости IT-проекта

Разработать бюджет проекта — это половина дела. Выполнить работы проекта и не превысить бюджета — вторая половина задачи.

Менеджер проекта должен выстроить эффективную систему контроля стоимости проекта и обеспечить своевременное выполнение корректирующих действий для минимизации отклонений от бюджета. Выделяют две основные задачи контроля стоимости:

- 1) учет фактических затрат;
- 2) прогноз будущих затрат.

Применение традиционных методов контроля стоимости часто позволяет решать только первую задачу контроля — учетную.

Традиционный контроль стоимости проекта позволяет провести анализ «план-факт» стоимости проекта.

Обычно такой расчет проводится с учетом двух основных значений:

- 1) плановая стоимость выполненных работ или освоенный объем (EV, Earned Variance).
- 2) Фактическая стоимость выполненных работ (AC, Actual Cost).

Разница между этими показателями называется отклонением по стоимости (CV, Cost Variance)

Если фактическая стоимость выполненных работ AC больше плановой стоимости работ EV, то наблюдается перерасход средств, если наоборот — то получается экономия денежных средств на проекте.

Между двумя этими величинами также может существовать отклонение

$$CV = EV - AC$$

В традиционном способе контроля отслеживаются только стоимостные показатели выполнения работ. В нем отсутствует возможность контроля объемных показателей проекта. Имеющейся информации недостаточно для прогнозирования хода выполнения работ.

Для принятия верных решений менеджер проекта должен обладать большим количеством информации:

- 1) сколько работ выполнено относительно плана;
- 2) отстает проект от графика или опережает;
- 3) сделано ли то, что должно быть выполнено к отчетной дате;
- 4) есть ли отклонения от плана работ по объемным показателям;
- 5) являются ли отклонения от графика случайными или это обоснованная тенденция.

Вся эта информация требуется менеджеру проекта для понимания дальнейшего хода выполнения проекта и расчета прогнозной стоимости всего проекта в изменившихся условиях.

Возможность решить перечисленные задачи дает использование метода освоенного объема (Earned Value Analysis), который часто применяется при контроле проекта.

В этом обычно помогает **метод освоенного объема**.

Метод освоенного объема — это совокупность инструментов, позволяющих измерить, проанализировать и спрогнозировать значения основных показателей проекта по стоимости, продолжительности и содержанию проекта. Основными показателями метода являются:

EV — плановая стоимость выполненных работ. Это стоимость работ, которые выполнены на момент анализа по плану согласно утвержденному бюджету. Эта цифра заложена в базовый бюджет в качестве плановой стоимости данного (фактически выполненного) объема работ;

AC — фактическая стоимость выполненных работ. Это стоимость работ, которые выполнены на момент анализа по факту, реально затраченные деньги на выполнение работ, которые уже фактически выполнены на данный момент;

PV (плановый объем, Planned Value) — плановая стоимость запланированных работ. Это стоимость работ, которые должны быть выполнены на момент анализа согласно утвержденному бюджету. Эта величина заложена в базовый бюджет в качестве стоимости того объема работ, который должен быть выполнен к данному моменту.

Сравнивая освоенный объем и фактическую стоимость, т. е. стоимость по бюджету и стоимость по факту тех работ, которые уже выполнены, можно определить, нет ли перерасхода средств в проекте.

CV (отклонение по стоимости, Cost Variance) — это разность между освоенным объемом и фактической стоимостью:

$$CV = EV - AC$$

- 1) Если $CV < 0$, в проекте имеет место перерасход средств.

2) Если $CV > 0$, в проекте имеет место экономия бюджета.

Физический смысл расчета показателя CV — сравнение реально выполненных работ в плановых (бюджетных) и фактических деньгах.

По отдельным показателям будет видна финансовая картина только с одной стороны. Даже если мы определили перерасход, то мы не сможем узнать, как выполняется расписание проекта. Может быть, перерасход получился из-за увеличенной скорости выполнения проекта. Мы не знаем выполнены ли все работы, а может быть выполнено работ меньше, чем запланировано.

Ответом на этот вопрос может быть показатель **планового объема (PV)**.

В принципе, имея этот показатель можно ответить на указанные выше вопросы.

Также учитывается SV (отклонение по расписанию, Schedule Variance) — это разность между освоенным объемом и плановым объемом:

$$SV = EV - PV$$

Работа с этим показателем происходит по следующей методике:

- 1) Если $SV < 0$, в проекте имеет место отставание от графика выполнения работ.
- 2) Если $SV > 0$, в проекте имеет место опережение графика выполнения работ.

Физический смысл расчета показателя SV — сравнение в плановых (бюджетных) деньгах объема работ, который реально выполнен (EV), и объема работ, который должен быть выполнен согласно графику работ.

С учетом этого показателя можно определить, например, что при положительном перерасходе (это, вроде бы как неприятный момент), имеется также положительное отклонение по расписанию, то есть работ было выполнено больше, чем изначально планировалось. В этом случае можно сказать, что высокая скорость работы приоритетнее перерасхода, хоть за это и пришлось заплатить.

9.3. Синхронизация стоимостных и качественных показателей проекта

На основе синхронизации параметров, а также стоимостных и качественных показателей продукта, можно выполнять прогнозирование стоимости всего проекта.

Следует помнить, что этот процесс проводится в динамике, то есть в ходе выполнения проекта во всех контрольных точках.

Этот процесс инициируется менеджерами проекта после адекватного анализа ситуации, фиксации текущего состояния проекта и анализа остатков.

Абсолютные величины показателей CV и SV дают возможность сделать выводы о текущем состоянии проекта, то есть провести его **срез**.

Также в расчете учитываются относительные показатели выполнения проекта.

- 1) CPI (Cost Performance Index) — индекс выполнения стоимости.

2) SPI (Schedule Performance Index) — индекс выполнения расписания.

Индекс выполнения стоимости характеризует эффективность расходования денежных средств в проекте и определяется отношением значений освоенного объема и фактической стоимости проекта.

$$CPI = EV / AC$$

Индекс выполнения расписания определяет степень достижения показателей проекта по объемам работ и выполнения расписания проекта. Он рассчитывается как отношение значения освоенного объема и планового объема.

$$SPI = EV / PV$$

К этим показателям применяется следующая методика расчета.

Индекс выполнения стоимости проекта характеризует стоимостные параметры проекта:

- 1) если $CPI < 1$, в проекте имеет место перерасход средств;
- 2) если $CPI > 1$, в проекте имеет место экономия бюджета.

Индекс выполнения расписания характеризует параметры выполнения расписания и объемов выполненных работ:

- 1) если $SPI < 1$, в проекте имеет место отставание по срокам;
- 2) если $SPI > 1$, в проекте имеет место опережение графика.

Индексы помогут менеджеру проекта в расчете прогнозов дальнейшего хода выполнения проекта. Для прогнозирования будущих стоимостных показателей проекта, в частности расчета оценки по завершении проекта, менеджеру проекта понадобится:

- 1) понимание, сколько денег уже израсходовано;
- 2) прогноз, сколько денег еще предстоит израсходовать.

Именно прогноз оставшейся стоимости проекта и является главной задачей команды управления стоимостью. Метод освоенного объема дает возможность рассчитать его с учетом реальной ситуации в проекте.

Для того, чтобы рассчитать бюджет проекта нужно определить еще несколько понятий.

- 1) Бюджет по завершении (BAC, Budget At complete).
- 2) Оценка по завершении (EAC, Estimate At Complete).

BAC определяет плановую стоимость выполнения всех работ проекта, которая зафиксирована в базовом варианте бюджета проекта.

EAC определяет расчетную или прогнозную стоимость выполнения работ проекта. Она вычисляется на базе имеющейся фактической информации о ходе проекта и его стоимостных показателей.

AC (Actual Cost) определяет фактическую стоимость проекта.

Для того, чтобы вычислить EAC нужно знать фактические затраты еще на начальном моменте проекта, когда фактически еще не было потрачено

ничего, а оценка по завершению будет определена фактически в завершении проекта.

Еще один интересный факт, это то, что $EAC = BAC$ в момент фиксации базового бюджета.

Также существует оценка до завершения проекта ETC (Estimate To Complete). Это прогнозное значение стоимости выполнения оставшихся работ проекта от момента выполнения анализа до окончания проекта.

Обладание этими значениями делает расчеты чисто технической задачей.

$$EAC = AC + ETC$$

Метод освоенного объема позволяет вычислить оценку до завершения с учетом складывающейся ситуации в проекте. Для этого анализируется существующая фактическая информация и выбирается способ расчета прогнозных показателей

Итак, мы можем определить следующую матрицу отношений между этими показателями.

Бюджет по завершении BAC		Рассчитывается в начале проекта и фиксируется в базовом бюджете
Оценка по завершении EAC		Рассчитывается в ходе проекта по фактической информации по прогнозным оценкам
		$EAC = AC + ETC$
Фактическая стоимость AC		Оценка до завершения ETC
Момент анализа проекта (текущий момент)		
<i>Прошлое проекта</i>	Сейчас	Оставшаяся часть проекта

Как видно из приведенных формул, индекс выполнения стоимости CPI вводится в расчетную формулу для учета тенденции выполнения стоимости в проекте.

- 1) Если в проекте имеется перерасход средств, введение в формулу значения $CPI < 1$ в знаменатель увеличивает значение оценки до завершения. Это логично. Раз есть перерасход сейчас, то сохранение тенденции приведет к общему перерасходу по итогам проекта.
- 2) Если в проекте на данный момент наблюдается экономия средств, то введение в формулу значения $CPI > 1$ в знаменатель увеличивает значение оценки до завершения. Раз есть экономия сейчас, сохранение тенденции приведет к экономии всего бюджета.

Применение оценок для построения прогнозов можно оценить или описать следующим образом.

Оценки	Условия применения	Оценка по завершении ЕАС
<p>Оценка до завершения, основанная на новой оценке</p> <p>Формула:</p> <p><i>новый прогноз</i></p>	<p>Применяется в случае наличия отклонений по стоимости. При этом фактическая информация о стоимостных параметрах проекта позволяет сделать вывод о значительных принципиальных ошибках при расчете бюджета по завершении (ВАС) в ходе планирования проекта. Команда проекта принимает решение провести новый расчет стоимости в изменившихся условиях</p>	<p>Оценка по завершении с использованием новой оценки. Она равна фактической стоимости работ на дату анализа плюс новый прогноз ЕТС, представленный командой проекта:</p> $EAC = AC + ETC$
<p>Оценка до завершения, основанная на нетипичных отклонениях</p> $ETC = BAC - EV$	<p>Применяется в случае наличия отклонений по стоимости. При этом команда проекта принимает решение, что подобное отклонение (перерасход или экономия бюджета) является случайным и предположительно больше не повторится. Тенденция неустойчивая</p>	<p>Оценка по завершении, основанная на нетипичных отклонениях.</p> $EAC = AC + (BAC - EV)$
<p>Оценка до завершения, основанная на типичных отклонениях</p> $ETC = (BAC - EV) / CPI$	<p>Применяется в случае наличия отклонений по стоимости. При этом команда проекта принимает решение, что подобное отклонение (перерасход или экономия бюджета) является не случайным и предположительно будет повторяться в дальнейшем. Тенденция устойчивая, необходимо использовать индекс выполнения стоимости CPI</p>	<p>Оценка по завершении, основанная на типичных отклонениях.</p> $EAC = AC + (BAC - EV) / CPI$

Для повышения точности прогнозов необходимо организовать регулярный сбор фактической информации о стоимостных показателях и пересчет прогнозных оценок. На основе полученных оценок следует составлять графики и анализировать существующие тренды и тенденции.

Если три-четыре отчетных периода расчетные показатели основных оценок имеют одинаковое значение, например «перерасход бюджета», доверие к такой оценке значительно возрастает. Если же три-четыре отчетных периода оценки показывают различные значения (то «перерасход», то

«экономия»)), возможно, это связано с особенностями стоимостного учета в проекте или с другими причинами.

Тема № 10. Методология управление стоимостью проекта. Бизнес-аналитика в управлении проектами

10.1. Методы управления стоимостью проекта

Стоимость проекта определяется суммой стоимостей ресурсов проекта, то есть стоимостями и временем выполнения работ. Общую стоимость проекта можно также рассчитать, как оценку всех затрат по проекту.

Соответственно, процесс управления стоимостью проекта включает в себя другие процессы, работа которых обеспечивает выполнение проекта в рамках утвержденного бюджета. В контексте нашего курса управление стоимостью и управление затратами проекта являются практически тождественными понятиями.

Цель процесса **управления стоимостью** заключается в разработке особой политики, а также процедур и методов, которые позволяют осуществить планирование и своевременный контроль затрат по проекту.

Мы можем назвать следующие процессы, которые входят в процедуру управления стоимостью или (другими словами) управления затратами проекта:

- 1) Оценка стоимости проекта.
- 2) Бюджетирование и установление целевых показателей затрат на реализацию проекта.
- 3) Контроль затрат проекта (оценка фактических затрат, периодическое сравнение с запланированным бюджетом, выработка мероприятий для предупреждения возможности превышения бюджета).

Естественно, что в процессе управления стоимостью проекта мы оперируем **бюджетом** проекта во всех его проявлениях. Мы можем описать бюджет как некоторый директивный документ с реестром расходов и доходов, с распределением по статьям и по периодам времени.

Бюджет проекта помогает понять ресурсные и временные ограничения проекта, поэтому при управлении стоимостью на первый план выходит список затрат, который мы также можем называть **сметой**.

Под сметой проекта мы будем понимать документ, который содержит список фактических стоимостей различных видов работ и ресурсов, различных контрактов и так далее. Обычно смета составляется на основе объемов работ проекта, ресурсов и цен.

Так как в основе этого процесса лежат затраты, то нам нужен способ управления затратами, например использование структуры счетов затрат или планов счетов.

Счета составляются по классическим правилам бухгалтерии: слева дебетовая сторона, а справа кредитовая сторона. Принцип управления счетом: выполнение в любой момент равенства между дебетом и кредитом (уравнивание баланса, нулевое сальдо).

Также часто используют уравнение Активы = Пассивы.

Активы отражают стоимость имущества и имущественных прав организации на определенную дату.

Пассивы указывают на источники возникновения активов.

Для описания процесса используют двойную запись, а каждая двойная запись называется **проводкой**. Каждая проводка изменяет обе части уравнения, оставляя баланс неизменным.

Так, увеличение активов увеличивает дебет счета, а увеличение пассива – кредит. И тут действует закон сокращения: сумма дебетов всех счетов должна быть равной сумме кредитов, обеспечивая нулевое сальдо.

Такой метод двойной записи позволяет отследить источники получения и направления расходов средств. Примером может служить ситуация, когда проводится оплата по кредиту «денежные средства» с одновременным сокращением в дебете счета «кредиторская задолженность» или с увеличением суммы «выданного аванса».

Проще говоря, «бухгалтерский» подход к управлению стоимостью IT-проекта помогает показать сводное финансовое положение компании.

Следует помнить, что в реальной ситуации у компании много работ и много смет. Для того, чтобы не «потеряться» в этом море документов и показателей, применяют иерархическую структуру работ WBS. На начальной стадии формирования бюджета работы все сметы составляются по требуемым ресурсам, а ресурсы списываются на разные статьи затрат.

Раз мы применяем иерархический подход, значит структура счетов затрат разрабатывается по принципу декомпозиции с агрегированием информации со счетов нижних уровней структуры «наверх».

Таким образом мы получаем расшифровку расходов на требуемом уровне детализации: общие потоки – на верхнем уровне декомпозиции, более подробные расходы – на нижних уровнях.

Нужно учитывать, что управление стоимостью проекта осуществляется на протяжении всего жизненного цикла проекта. Это находит отражение в современной концепции управления стоимостью проекта, которая называется **управление стоимостью на протяжении проекта, LLC *life-Cycle Costing**).

По сути, мы можем представить этот процесс в виде следующих потоков или процессов.

Выше всего интенсивность процесса оценки стоимости проекта наблюдается в моменты основных работ по проекту, так что этот процесс больше похож на трехфазную фигуру:

- 1) Плавный рост интенсивности.
- 2) Максимальная интенсивность.
- 3) Резкое снижение интенсивности.

Концепция проекта	Обоснование проекта	Планирование проекта	Реализация проекта	Завершение проекта
Укрупненная оценка стоимости				
	Детальная оценка стоимости			
		Бюджетирование		
			Завершающая оценка проекта	

Давайте далее рассмотрим, как можно скорректировать стоимость проекта без учета рисков и с учетом возможных рисков.

10.2. Методы коррекции стоимости проекта в связи с рисками

Оценка стоимости проекта определяется ресурсами, которые необходимы для выполнения проекта. В их число входят:

- 1) Расходы на оборудование.
- 2) Расходы на приспособления, устройства и другие производственные мощности.
- 3) Оплата труда работников (штатных и нанятых по контракту).
- 4) «Расходники».
- 5) Материалы.
- 6) Обучение персонала, повышение квалификации, семинары и т.д.
- 7) Субконтракты.
- 8) Перевозка, оплата транспорта.
- 9) Оплата связи и так далее.

В общем случае стоимость проекта может корректироваться по ходу выполнения и в определенных пределах.

Можно описать этот процесс так.

- 1) Разработка концепции проекта, - предварительная оценка (оценка жизнеспособности) с погрешностью 25% - 40%.
- 2) Обоснование инвестиций, - факторизация, сопоставление плана затрат с лимитами с погрешностью 20 % - 30 %.
- 3) Технико-экономическое обоснование проекта, - принятие окончательного инвестиционного решения с погрешностью в 15 % - 20 %.
- 4) Разработка рабочей документации, - основные расчеты для управления стоимостью проекта с погрешностью 3 % - 5%.
- 5) Реализация проекта, - оценка стоимости производимых работ с погрешностью около нуля.
- 6) Сдача проекта в эксплуатацию, - обработка двух оценок: фактической и прогнозной с погрешностью 3 % - 5 %.
- 7) Эксплуатация, опять же обработка фактической и прогнозной стоимостей с погрешностью 3 % - 5 %.

8) Завершение проекта с полной переоценкой стоимости проекта, с нулевой погрешностью.

Погрешности и допуски при расчетах помогают учитывать возможные риски проекта. То есть, по самому простому понятию, если в бюджет заложили 10 % превышение на «возможные» риски, но рисков не случилось – это хорошо. Если все-таки произошло, то «это было так запланировано».

Как с учетом рисков, так и без их учета, мы можем классифицировать затраты как:

- 1) прямые и накладные;
- 2) повторяющиеся и единовременные;
- 3) постоянные и переменные;
- 4) сверхурочные оплаты.

При всем этом мы также можем определять три вида затрат проекта:

- 1) обязательства;
- 2) бюджетные затраты;
- 3) фактические затраты.

С учетом этих особенностей бюджета проекта, мы можем сказать, что стоимость изменяется или «мигрирует», и этим процессом можно управлять.

10.3. Миграция стоимости проекта

Управление изменениями бюджета проекта представляет собой процесс планирования будущих изменений, фиксация всех потенциальных (возможных) изменений в содержании проекта, в его сетевом графике. В спецификациях и так далее. Это делается для детального изучения, оценки последствий, а также для организации мониторинга и коррекции возможных изменений в проекте.

По результатам анализа возможен перенос стоимости проекта на будущие периоды.

Под изменениями мы будем понимать замещение одного «решения» другим под действием внешних и внутренних факторов и рисков.

Изменения могут вноситься в различные разделы проекта. Инициировать изменения может заказчик, инвестор, менеджер, подрядчик или другой участник проекта, обладающий соответствующими компетенциями и правами.

В целом, можно привести следующий примерный список изменений:

- 1) цена проекта;
- 2) план проекта;
- 3) ресурсы проекта;
- 4) контракты;
- 5) обязательства;
- 6) стандарты и нормативы, используемые «здесь и сейчас»;
- 7) география размещения объектов.

Все множество изменений можно разделить на следующие виды:

- 1) осознанные (желательные) изменения;
- 2) вынужденные изменения.

Вынужденные изменения должны быть вовремя распознаны и реализованы с наименьшими убытками. Возможность же выполнения желаемых изменений должна быть дифференцирована и реализована с выгодой для проекта.

Неконтролируемые изменения, производящиеся в процессе реализации проекта, могут носить разрушающий характер для всего процесса управления.

Вынужденные изменения должны быть вовремя распознаны и реализованы с наименьшими убытками. Возможность же выполнения желаемых изменений должна быть дифференцирована и реализована с выгодой для проекта.

Неконтролируемые изменения, производящиеся в процессе реализации проекта, могут носить разрушающий характер для всего процесса управления.

10.4. Капитализация стоимости проекта

Рассмотрим классическое определение понятия «капитализация».

Капитализация (или рыночная капитализация) — стоимость актива (компании, ценной бумаги, криптовалюты и т. д.), рассчитанная на основе текущих биржевых котировок.

В сфере IT-проектов термин капитализации можно соотнести с «капитализацией бизнеса». Конечно, обычно слышать термины из банковской сферы вроде «капитализация вклада» или «капитализация процентов».

С точки зрения банковского бизнеса, капитализация процентов или капитализация вклада означает суммирование накопленных по вкладу за определенный период времени процентов с общей суммой вклада в конце этого периода (например, месяца или года). Тогда следующие проценты будут уже рассчитываться и начисляться исходя из новой увеличенной суммы вклада. Получается график, похожий на «лестницу».

Однако нас здесь более интересует применение термина капитализация с точки зрения реального бизнеса, например с точки зрения компании, производящей программное обеспечение.

В таком случае, под капитализацией мы будем понимать рыночную стоимость компании, определенную в какой-либо определенный момент времени.

Капитализация считается по «рыночной цене», то есть по той цене, какая бы была у компании, если бы у нее существовали акции и они могли бы продаваться свободно на рынке ценных бумаг.

С другой стороны, если мы все-таки говорим о рыночной капитализации акционерной компании, акции которой прошли первичное размещение на фондовой бирже и теперь свободно возвращаются на открытом рынке, то рыночную капитализацию компании посчитать очень просто — нам нужно умножить количество акций компании на текущую рыночную стоимость одной акции. Сама стоимость акций постоянно колеблется и является выражением мнения огромного количества покупателей и продавцов — именно так и достигается мнение свободного конкурентного рынка.

Естественно, что, если дела компании идут не слишком хорошо, стоимость ее акций падает и падает рыночная капитализация бизнеса. Если компания показывает хорошие результаты, ее выручка и прибыль стабильно растут, то рынок, а именно люди, инвесторы, оценивают положительные тенденции и формируют спрос на акции, - то есть капитализация компании растет.

Существует множество компаний, которые не имеют своих акций и которые не известны публично и не «продаются» на свободном рынке. Для таких компаний также можно рассчитать рыночную капитализацию.

Рыночную капитализацию такой компании можно рассчитать одним из следующих способов:

- 1) по текущей EBITDA компании;
- 2) по финансовой модели компании.

EBITDA – это аналитический показатель, равный объему прибыли до вычета расходов по выплате процентов, налогов, износа и начисленной амортизации.

Показатель рассчитывается на основании финансовой отчетности компании и служит для оценки того, насколько прибыльна деятельность компании без учёта амортизационных отчислений.

Алгоритм расчета показателя следующий.

Чистая прибыль

+ Начисленный налог на прибыль

- Возмещенный налог на прибыль

(+ Чрезвычайные расходы)

(- Чрезвычайные доходы)

+ Проценты уплаченные

- Проценты полученные

= EBIT

+ Амортизационные отчисления по материальным и нематериальным активам

- Дооценка активов

+ Обесценение активов

= EBITDA

EBITDA по текущей компании должен быть умножен на ее рыночный мультипликатор. Берется EBITDA за последний отчетный год и умножаем на бенчмаркинг-мультипликатор рыночной мультипликатор.

Например, наша розничная сеть заработала в прошлом году выручку 100 млн. руб., ее EBITDA составила 15 млн. руб. и мы знаем, что на рынке были похожие сделки с мультипликатором 10 – мы умножаем 15 млн. руб. на 10 и получаем первичную рыночную оценку нашей сети – 150 млн. руб.;

По финансовой модели компании, мы проводим расчет, в котором учитываем будущий свободный ежегодный денежный поток компании (чистые деньги, которые бизнес может получать в обозримом будущем).

После этого мы прогнозируем и доказываем себе и потенциальному покупателю, что в будущем 10 лет мы суммарно получим 150 млн. руб.

свободных (чистых) денег, которые можно будет безболезненно изъять из компании – это и будет являться его стоимостью или рыночной капитализацией.

Тема № 11. Структуризация проекта и разработка документации

11.1. Особенности структуризации ИТ проектов

Структуризация проекта служит для того, чтобы можно было эффективно управлять проектом.

В терминах управления проектами структура проекта представляет собой «дерево» ориентированных на продукт компонентов, представленных оборудованием, работами, услугами и информацией, полученными в результате реализации проекта.

С точки зрения теории систем, структура проекта, — это его граф, то есть совокупность элементов и связей между ними.

Вообще говоря, такой граф строится как иерархическая декомпозиция проекта с разделением на составные части.

Структура проекта призвана определить продукцию, которую необходимо разработать или произвести, и связать элементы работ, которые предстоит выполнить, как между собой, так и с конечной целью проекта.

Мы уже упоминали основной термин процесса структуризации проекта WBS (Work Breakdown Structure). WBS состоит в следующем:

- 1) разделяются цели проекта на меньшие блоки;
- 2) проект делится на поддающиеся измерению, описанию и управлению элементы работ;
- 3) для элементов определяются затраты;
- 4) строятся графики исполнения.

Должным образом подготовленная и составленная структура проекта должна удовлетворять требованиям менеджера проекта и заказчика. Структуризация проекта помогает менеджеру наделить участников проекта ответственностью за выполнение конкретных технических заданий. Она также позволяет создать простую систему отслеживания хода реализации проекта.

Применительно к реальным проектам структура разбивки проекта должна сочетать разделение на:

- 1) компоненты продукции;
- 2) этапы жизненного цикла проекта;
- 3) элементы организационной структуры.

Кроме этого, процесс структуризации проекта является неотъемлемой частью общего процесса планирования проекта и определения его целей, а также подготовки сводного (генерального) плана проекта и матрицы распределения ответственности и обязанностей.

Осуществление этого процесса может проводиться по следующим подходам:

- 1) продуктовый подход;
- 2) подход по жизненному циклу;
- 3) смешанный подход.

Давайте рассмотрим основные задачи структуризации проекта:

- 1) определение комплексов работ проекта;
- 2) переход от общих, не всегда конкретно выраженных, целей к определенным заданиям;
- 3) увязка работ по проекту с системами бухгалтерского и управленческого учета;
- 4) создание единой базы данных для планирования, составления смет и контроля над затратами;
- 5) точная оценка необходимых затрат времени, финансовых средств и материальных ресурсов;
- 6) распределение ответственности за различные элементы проекта и увязка работ с наличными ресурсами и структурой организации;
- 7) разбивка проекта на поддающиеся управлению блоки.

При структуризации проекта возможно допущение следующих ошибок:

- 1) возможен «пропуск» некоторых «неопределенных» или «неосязаемых» конечных продуктов (к которым обычно относятся услуги, а также информационное и программное обеспечение);
- 2) возможно получение результатов, которые невозможно обработать;
- 3) возможна излишняя детализация структуры проекта;
- 4) возможна недостаточная детализация структуры проекта;
- 5) возможно отсутствие «синхронизации» структуры проекта с системой ведения бухгалтерии;
- 6) возможно повторение одних и тех же элементов структуры проекта;
- 7) возможно «зацикливание» связей в структуре проекта;
- 8) возможна рассинхронизация структуры проекта с жизненным циклом;
- 9) возможен простой пропуск стадии структуризации проекта и переход сразу к анализу, без детализации и декомпозиции.

Как правильно осуществлять структуризацию проекта? Главная задача при структуризации проекта — найти вещественные компоненты проекта. Это напоминает задачу разбивки книги на главы, земельной площади — на участки, компьютерной программы — на модули. Это — то, что называют подпродуктовой структурой.

Безусловно, разбивка проекта на составные части должна учитывать все конечные продукты проекта. Однако в процессе структуризации должны учитываться также и этапы жизненного цикла проекта (структура процесса), такие как, например, планирование, сборка и т. д.

11.2. Системы контроля версий

Контроль версий иногда называют «управлением» исходным кодом, — это общепринятая практика отслеживания изменений программного кода.

Система контроля версий (Version Control Systems, VCS) позволяет сохранять изменения в файлах программ, прятать их, переключаться между ними. При этом можно отследить кто из разработчиков, когда и какие изменения в код проекта. Можно переключаться между версиями одного и

того же файла, принимая наилучший вариант, соответствующий техническому заданию.

Современные системы контроля версий окружены специальным программным обеспечением, которое носит общее название GIT. В эту систему включены инструменты командной строки, графические оболочки, сервера и веб-приложения.

Программное обеспечение контроля версий отслеживает все вносимые в код изменения в специальной базе данных. При обнаружении ошибки разработчики могут вернуться назад и выполнить сравнение с более ранними версиями кода для исправления ошибок, сводя к минимуму проблемы для всех участников команды.

Практически во всех программных проектах исходный код является сокровищем: это ценный ресурс, который необходимо беречь. Для большинства команд разработчиков программного обеспечения исходный код — это репозиторий бесценных знаний и понимания проблемной области, которые они скрупулезно собирали и совершенствовали. Контроль версий защищает исходный код от катастрофических сбоев, от случайных ухудшений, вызванных человеческим фактором, а также от непредвиденных последствий.

Разработчики программного обеспечения, работающие в командах, постоянно пишут новый исходный код и изменяют существующий. Код проекта, приложения или программного компонента обычно организован в виде структуры папок или «дерева файлов». Один разработчик в команде может работать над новой возможностью, а другой в это же время изменять код для исправления несвязанной ошибки, т. е. каждый разработчик может вносить свои изменения в несколько частей дерева файлов.

Контроль версий помогает командам решать подобные проблемы путем отслеживания каждого изменения, внесенного каждым участником, и предотвращать возникновение конфликтов при параллельной работе. Изменения, внесенные в одну часть программного обеспечения, могут быть не совместимы с изменениями, внесенными другим разработчиком, работавшим параллельно. Такая проблема должна быть обнаружена и решена согласно регламенту, не создавая препятствий для работы остальной части команды.

Современный GIT является неотъемлемой частью рабочего процесса любой команды и индивидуальных разработчиков. Сейчас есть несколько наиболее популярных «провайдеров» контроля версий:

- 1) Github.
- 2) Gitlab.
- 3) Bitbucket.

Также есть системы, основанные на Subversion и Mercurial, но они не так широко распространены.

Независимо от того, какой вариант системы контроля версий используется, ее основные преимущества заключаются в следующем.

- 1) Полная история изменений каждого файла за длительный период. Это касается всех изменений, внесенных огромным количеством

людей за долгие годы. Изменением считается создание и удаление файлов, а также редактирование их содержимого.

- 2) Ветвление и слияние. Эта возможность полезна не только при одновременной работе участников команды: отдельные люди также могут извлечь из нее пользу и работать над несколькими независимыми направлениями. Создание «веток» в инструментах VCS позволяет иметь несколько независимых друг от друга направлений разработки, а также выполнять их слияние, чтобы разработчики могли проверить, что изменения, внесенные в каждую из веток, не конфликтуют друг с другом.
- 3) Отслеживаемость. Возможность отслеживать каждое изменение, внесенное в программное обеспечение, и связывать его с ПО для управления проектами и отслеживания ошибок, например Jira, а также оставлять к каждому изменению комментарий с описанием цели и назначения изменения может помочь не только при анализе основных причин возникновения ошибок, но и при проведении другого анализа. История с комментариями во время чтения кода помогает понять, что этот код делает и почему действие реализовано именно таким образом. Благодаря этому разработчики могут вносить корректные и совместимые изменения в соответствии с долгосрочным планом разработки системы.

Современная теория управления IT-проектами немислима без систем контроля версий, так как существует понятие Git-flow.

Git-flow — альтернативная модель ветвления Git, в которой используются функциональные ветки и несколько основных веток. В Git-flow создается множество веток с использованием нескольких правил:

- 1) Главная ветка main или master.
- 2) Основная ветка разработки development.
- 3) Ветка текущих релизов release.
- 4) Ветки возможностей (фичи) feature/*.
- 5) Ветки исправлений (баги) fix/* и hotfix/*.
- 6) Ветки для тестирования test/*.

Это не абсолютная или предопределенная структура, а лишь рекомендация. Многие разработчики формируют свой собственный поток управления состоянием и ветвлением. Следует обратить внимание, что ветки main и release меняются медленнее всего, но содержат более крупные изменения, рабочая ветка development имеет средний темп изменений и является самой актуальной в любой момент. Ветки с исправлениями содержат меньше всего изменений и меняются быстрее всего.

В классическом варианте Git-flow, функциональные ветки (фичи) содержат новые возможности, добавленные в проект. В основном функциональные ветки создаются из development, то есть содержат самые последние изменения, включая возможности из соседних веток feature/*.

Релизная ветка release объединяет изменения в тот момент, когда команда понимает, что ветка development накопила достаточно изменений для

очередного соединения с master и нужно делать релиз, то есть выпускать очередную версию программного обеспечения.

Благодаря тому, что для подготовки выпусков используется специальная ветка, одна команда может дорабатывать текущий выпуск, в то время как другая команда продолжает работу над функциями для следующего.

Когда подготовка к поставке завершается, релиз сливается с ветками main и develop, а ветка release удаляется. Важно слить ее с веткой develop, поскольку в ветку release могли добавить критические обновления, которые должны быть доступны для новых функций.

Ветки исправлений fix/*, hotfix/* служат для сопровождения и используются для быстрого внесения исправлений в релизы. Согласно классическому Git-flow, ветки исправлений создаются из master, а не из development. Благодаря специальной ветке для исправления ошибок команда может устранять проблемы, не прерывая остальную часть рабочего процесса и не дожидаясь следующего цикла релиза.

Как уже говорилось выше, это рекомендованная структура Git-flow и многие компании могут вносить в нее изменения, главное помнить, что данная модель отлично подходит для составления регламента и систематизации рабочего процесса компании.

Наличие Git-flow в компании помогает упорядочить процесс разработки программного обеспечения и синхронизировать его с системой управления задачами.

11.3. Документирование и автодокументирование кода

Документирование кода является важным элементом сопровождения проекта и обеспечения его жизнеспособности.

Со временем объем кодовой базы растет, - некоторые современные программные продукты насчитывают миллионы строк кода и сотни модулей. Естественно, что без документации кодовой базы разобраться в таком объеме просто невозможно.

Для программного кода можно выделить две надежные системы документирования:

- 1) Документация и автодокументация.
- 2) Самодокументируемы код.

При все этом, документирование кода, - это не просто комментарии, а именно строки документации или отдельные файлы, в которых описано что делает конкретный модуль, класс, функция или блок кода.

Основное назначение комментариев – пояснить что делает код, как он работает. Основное назначение строк документации – кратко описать в целом для чего предназначен объект, какие аргументы принимает, и что возвращает.

Такие строки с описанием называются dockstring.

Существуют отдельные правила для составления dockstring, то есть правила описания и правила содержания этих строк.

Строки документации могут содержать различные тэги, по которым можно форматировать отображаемое описание при помощи средств автодокументирования.

Кроме того, строки документации помогают разработчикам в интегрированных средах разработки легче понимать, что делает функция, которую он вызывает и какие правила вызова существуют. Например, в сплывающих окнах подсказок видно краткое описание функции, а также названия входных параметров функций и их типы.

Теперь поговорим о системах автодокументирования. Самой известной из них является Doxygen.

Doxygen требует оформления комментариев функций для документирования согласно определенного стандарта. Общий принцип работы Doxygen следующий: на вход такого генератора поступает специальным образом комментированный исходный код, а иногда и другие компоненты программы, а на выходе создаётся готовая документация для распространения и использования.

Doxygen позволяет генерировать на основе исходного кода, содержащего комментарии специального вида, красивую и удобную документацию, содержащую в себе ссылки, диаграммы классов, вызовов и т.п. в различных форматах: HTML, LaTeX, CHM, RTF, PostScript, PDF, man-страницы.

Так как в строки документации функций можно вставлять даже HTML-тэги, то страницы документации получаются сверстанные как самые современные сайты. Там поддерживается навигация, ссылки, меню, формулы и так далее.

Doxygen также содержит команды вызова и принимает определенные параметры вызова. Вообще, любая команда в Doxygen представляет собой слово на английском языке, предваренное символом "\" или "@" (обе записи тождественны) и таких команд очень много, порядка двухсот. Например:

- 1) \authors – указывает авторов кода.
- 2) \version – используется для указания версии.
- 3) \date – используется для указания даты разработки или релиза.
- 4) \bug – приводит список известных ошибок.
- 5) \warning – приводит предупреждения для использования.
- 6) \example – добавляет комментарий для указания ссылки на исходный код с примером.
- 7) \todo – используется для описания изменений, которые необходимо будет выполнить.

Можно описать несколько подходов к составлению документации.

- 1) Документирование на основе исходного кода.
- 2) Документирование файла.
- 3) Документирование функций и методов.
- 4) Документирование классов.
- 5) Документирование перечислений.

Иногда команда разработки не тратит время на формирование отдельной документации на проект, а используют подход самодокументирующегося кода.

11.4. Самодокументирующийся программный код

Самодокументирующийся код, — это концепция разработки программного обеспечения, в которой применяются определенные соглашения об именах и содержимому функций программного кода.

Обычно сформулированные цели для самодокументирующих систем включают:

- 1) Упростить чтение и понимание исходного кода.
- 2) Свести к минимуму усилия, необходимые для поддержки или расширения унаследованных систем.
- 3) Уменьшить потребность пользователей и разработчиков систем в обращении к вторичным источникам документации, таким как комментарии к коду или руководства по программному обеспечению.

Другими словами, написанный по определенным правилам код не нуждается в отдельной документации, так как из названий функций, классов и переменных можно понять, что делает тот или иной блок кода.

Такой код можно писать только если соблюдать:

- 1) единообразии в соглашении об именах;
- 2) единообразии в написании кода;
- 3) принципы разграничения ответственности блоков кода;
- 4) принципы ограничения обязанностей блоков кода.

Также существует критика подхода самодокументирующегося кода. Говорят, что такой код не может сам по себе объяснить причины написания и почему он так реализован. Он лишь описывает смысл кода и принцип его работы.

11.5. Методика ведения справочных систем

Документирование кода является частью методики ведения справочных систем или системы управления нормативной и справочной документации (НСИ).

Большая система имеет определенные недостатки, которые становятся только «хуже» с ростом масштаба проекта. Например:

- 1) Децентрализация информационной системы.
- 2) Разноплатформенность информационных систем.
- 3) Низкое качество информации в системах документирования и сопровождения проекта.
- 4) Проблемы в ведении и поддержке справочных данных.
- 5) Проблемы доступа к документации и поиска информации и так далее.

Для того, чтобы создать правильный и жизнеспособный проект, нужно использовать системы НСИ.

НСИ, разработанная по правилам, делает следующие вещи:

- 1) Обеспечивает единообразие документации и целостность данных.

- 2) Унифицирует механизмы взаимодействия бизнес-процессов системы.
- 3) Создает возможность построения многоуровневой системы контроля информационной системы.
- 4) Предупреждает возникновение проблем с качеством и достоверностью информации.
- 5) Повышает прозрачность бизнес-процессов и обоснованность принимаемых решений.
- 6) Обеспечивает повышение эффективности рабочего процесса команд разработчиков.

При внедрении системы нормативно-справочной информации проводятся следующие мероприятия.

- 1) Проводится исследование проекта на предмет возможности внедрения системы НСИ:
 - a. выполняется анализ организационно-методических документов, регламентирующих введение НСИ;
 - b. проводится описание бизнес-процессов системы;
 - c. выполняется описание справочников и классификаторов программного обеспечения;
 - d. выполняется аудит готовых справочников и классификаторов.
- 2) Разрабатывается и внедряется подсистема методического и организационного обеспечения:
 - a. создается организационная структура ведения НСИ;
 - b. проводится оптимизация бизнес-процессов;
 - c. параллельно с созданием самого программного обеспечения разрабатываются компоненты программного обеспечения, а также дополнительные регламенты, методики и инструкции по ведению НСИ.
- 3) Выполняется нормализация данных:
 - a. проводится нормализация и чистка справочников и подсистем документации;
 - b. разрабатываются классификаторы.
- 4) Настраивается и перенастраивается НСИ, согласно жизненному циклу разработки программного обеспечения, НСИ развивается параллельно исходному коду проекта.

11.6. Документальное оформление проекта. Основные типы и формы обязательных документов.

Мы только что обсудили формы документирования исходного кода в IT-проектах, - давайте посмотрим методологию разработки технологической документации на проект.

На разработку документации проекта применяются определенные стандарты, например Стандарт Единой технологической документации ЕСТД, исходная информация на проектирование, производственные требования, требования к нормоконтролю и так далее.

Согласно ЕСТД, технологическая документация (ТД) – это комплекс текстовых и графических документов, определяющих в отдельности или в совокупности производства программного продукта.

Зачет грамотно оформлять документацию на проект?

Техническая документация позволяет оценить стоимость разработки и согласовать функциональность будущей системы. При возникновении споров о стоимости и сроках разработки той или иной фичи она может стать определенной гарантией для заказчика. С другой стороны, если возникнет потребность в развитии приложения, документация облегчит процесс доработки и даст четкое понимание, возможно ли встроить новую функциональность в существующую систему.

Обычно разработкой ТЗ занимается аналитик. Именно он общается с заказчиком / заинтересованным лицом и выявляет у него требования. В сложных случаях к процессу можно привлечь менеджера проекта, разработчиков, тестировщиков и других специалистов. Чем сложнее проект — тем больше требований к документации. Серьезные заказчики из крупных корпораций и госсектора требуют составлять документацию в соответствии с ГОСТами, а это задача очень трудоемкая, требующая внимательности к деталям и постоянной вовлеченности в процесс. И это не только ТЗ, но и технический проект, полностью описывающий все используемые в разработке решения, подходы и методологии. На него также распространяются ГОСТы. Подготовкой такой документации должен заниматься специалист — технический писатель.

Разработка решений для крупных заказчиков может длиться год и более, это довольно долгий срок. За это время возникает достаточное количество изменений, провоцирующих обновление документации. Кроме того, любая долгосрочная разработка ПО предполагает развитие. В этом случае актуальная техническая документация позволит снизить риски, закладываемые в работу исполнителей.

Обычно документацией занимается специальный человек в компании – технический писатель. Строго сформулированного перечня должностных обязанностей технического писателя не существует: каждая компания формирует его сама, а иногда возлагает на такого специалиста и дополнительные задачи. Поэтому иногда бывает сложно даже определить род деятельности технического писателя.

Для этого он собирает информацию и материалы от участников проекта и документирует эти данные согласно требованиям заказчика, в том числе и в соответствии с ГОСТами. Речь идет о ГОСТ 19 «Единая система программной документации» и ГОСТ 34 «Информационная технология. Комплекс стандартов на автоматизированные системы». Также технический писатель следит за актуальностью технической информации, если это необходимо на длительных и сложных проектах.

Разработка технической документации начинается с оформления требований заказчика и заинтересованных лиц.

На основании этих требований оформляется содержимое документа «Техническое задание».

Техническое задание — это документ, регламентирующий бизнес-цели, общее описание системы, объем работ, границы проекта, а также порядок разработки, оценки и приемки. Данный документ отвечает нам на вопрос «что нужно сделать?» и фактически является постановкой задачи.

Аналитик составляет ТЗ и согласует его со всеми заинтересованными сторонами. После того как собраны и утверждены все требования, можно приступать к созданию прототипов будущей системы и разработке программного обеспечения.

На этом этапе начинается разработка макетов, сценариев, архитектуры и др. Раз уж мы говорим об эталонном процессе разработки, то все это должно быть описано в техническом проекте, который также использует часть информации из ТЗ.

Технический проект — это совокупность документов, описывающих и обосновывающих все подходы, методы, архитектурные и технические решения, применяемые для создания системы. Например, в технический проект включают макеты интерфейсов, описание протоколов для интеграции со смежными системами и оборудованием, пользовательские сценарии, описание алгоритма и их формирование, структура серверов и баз данных, а также другие требования к системе и ее взаимодействию с другими внешними системами.

Это далеко не все: существует много стандартов для написания технической документации, и для каждой страны они свои. В отечественных стандартах есть отдельно ГОСТ на создание автоматизированных работ и на программное изделие. Например, технический проект по ГОСТу 19 «Единая система программной документации» может включать в себя следующий перечень работ:

- 1) уточнение структуры входных и выходных данных;
- 2) разработка алгоритма решения задачи;
- 3) определение формы представления входных и выходных данных;
- 4) определение семантики и синтаксиса языка;
- 5) разработка структуры программы;
- 6) окончательное определение конфигурации технических средств;
- 7) разработка плана мероприятий по разработке и внедрению программ;
- 8) разработка пояснительной записки;
- 9) согласование и утверждение технического проекта.

Есть проекты, в которых важно иметь полную документацию до начала работ. Это касается решений с высокими требованиями к безопасности, соблюдению законодательства и т. д. Например, мобильные приложения для банков. В этом случае важно сначала продумать все детали системы (информационная безопасность клиентов и самого банка, соответствие законам). На это уйдет больше времени, но позволит избежать финансовых и репутационных рисков.

Если разработка идет по методологии Agile, то нет смысла прописывать всю документацию до старта работ. Если заказчику важно работать по спринтам и контролировать ход разработки, документацию можно дописывать параллельно основному процессу.

Если в компании возможны оба варианта — то нужно просто адаптироваться под условия и пожелания заказчика.

Большинство комплектов технической документации является собственностью компаний и подлежат политики о неразглашении, но можно привести пример оформления функциональных требований, так как они могут отличаться от рекомендаций ГОСТов и больше соответствуют продуктам из коммерческой сферы, чем принятым в госсекторе.

Для начала давайте разберемся, что такое функциональные требования.

Функциональные требования — это постановка задачи разработчику. Все, что не указано в требованиях, делается на усмотрение разработчика, что часто расходится с представлением продакт-менеджер об ожидаемом результате. Поэтому требования должны содержать ответы на все возможные вопросы по задаче.

Функциональные требования, как правило, состоят из:

- 1) User story — показывает, чего вы ожидаете от команды разработки.
- 2) Use cases — показывают сценарии использования фичи.
- 3) Wireframes — средство визуализации своей идеи.

User story описывает, что делает пользователь определенной роли для достижения результата, и что нужно сделать разработчику, чтобы воплотить эту задачу в жизнь.

Как правило используется шаблон:

As a/an <Название роли>, I want to <Цель, Действие>, so that <Ожидаемый результат>, to do <Что нужно сделать разработчику>

Use cases описывает поведение пользователя по шагам при взаимодействии с разрабатываемым продуктом.

Задача пользователя — это то, что делает пользователь для достижения краткосрочных целей.

Если пользователь решает задачу на разрабатываемой странице несколькими путями, то на каждое решение должен быть написан свой use case. Например, если доступ к затрагиваемому функционалу находится на нескольких страницах, нужно написать отдельный use case на каждый способ перехода пользователя к функционалу.

Для use case применяется правило оформления Целей и Задач с точки зрения пользователей.

Для каждой задачи можно написать свой use case — описание того, как пользователь взаимодействует с интерфейсом.

Пример Use Case: Задача загрузки изображений в личном кабинете клиента:

- 1) Клиент заходит в свой личный кабинет.
- 2) Клиент открывает раздел «Галерея».

- 3) Клиент загружает изображения через drag&drop или с помощью клика по кнопке «Выбрать файлы».
- 4) Изображения загружаются.
- 5) Пользователь видит уведомление об успешной загрузке изображений.

Другим примером может служить задача Удаления файлов.

- 1) Пользователь кликает на изображение.
- 2) Изображение выделяется.
- 3) Выделение можно снять при помощи клика на область за пределами выделенного изображения.
- 4) Пользователь нажимает на иконку «три точки».
- 5) Появляется контекстное меню.
- 6) Пользователь выбирает в нем ссылку «Удалить файл». Если было выделено несколько изображений, то удалятся все.
- 7) Изображение удаляется.

Почему же функциональные требования так важны с точки зрения современной разработки программного обеспечения?

Обычно постановка задачи разработчикам рождает у них множество вопросов, от ответов на которые зависит сложность и срок реализации. Для уточнения деталей им приходится тратить время на коммуникацию вместо своей прямой работы — создания классных фич и улучшения продукта. И даже в процессе коммуникации не всегда выясняются все тонкости, если постановщик задачи только отвечает на возникающие вопросы, но не проходит путь пользователя сам.

На приведенном выше примере можно также проиллюстрировать эту проблему.

Например, если бы задача звучала просто как «удалить картинку в личном кабинете», то у разработчиков сразу появилось бы множество вопросов:

- 1) Нужно ли удаление файла вообще?
- 2) Будет ли это ручное удаление или автоматически стираются самые старые файлы при загрузке новых, если превышен лимит пространства для хранения?
- 3) удаление происходит из списка файлов или нужно открыть файл?
- 4) файл удаляется навсегда или есть корзина для файлов, где они хранятся какое-то время? если нужна корзина, то сколько файлы в ней хранятся?
- 5) должно ли быть пакетное удаление файлов или можно удалять только одному?
- 6) файл удаляется с помощью отдельной иконки (как выглядит эта иконка?) или через пункт меню (как он будет называться? на каком месте в списке действий расположен?)
- 7) и т.д.

Это был всего один пункт в функциональном описании, а мы получили такое количество вопросов и на выяснение каждого ответа требуется время как со стороны исполнителей, так и со стороны заказчиков.

Функциональные требования помогают менеджеру проекта продумать и четко сформулировать все сценарии взаимодействия пользователя с интерфейсов в рамках задачи.

Чем точнее поставлена задача и чем больше деталей есть у разработчиков до начала работы, тем эффективнее идет работа. Не тратится время на долгую и подчас бессмысленную коммуникацию. В этом случае все стороны в выигрыше: разработчики получают четкое понимание, что и как нужно сделать, а поставщик задачи получает выполненную работу именно в том виде, в каком он ее себе представлял.

Тема № 12. Подготовка, техническое обеспечение и управление параметрами проекта

12.1. Подготовка проекта

Подготовка проекта к «старту» - достаточно сложная задача, состоящая из нескольких этапов. Обычно это:

- 1) Инициация проекта.
- 2) Разработка концепции проекта.
- 3) Первые исследования и прединвестиционный анализ.
- 4) Проектный анализ.
- 5) Анализ финансовой реализуемости проекта.
- 6) Техничко-экономическое обоснование.
- 7) Составление бизнес-плана.
- 8) Анализ ресурсной обеспеченности проекта.
- 9) Планирование.
- 10) Презентация.
- 11) Формирование портфеля проекта.
- 12) Начало работ.
- 13) Следование жизненному циклу.
- 14) Завершение проекта или сопровождение проекта.

Это минимальные шаги, которые позволяют начать проект более или менее грамотно.

Все, фактически, начинается с идеи проекта, исследования рынка и формирования концепции проекта.

Если с идеей все понятно, то нужно поговорить о том, чем же «концепция проекта» отличается от нее, почему это два разных понятия.

Идея – это мысль или смысл проекта. Источником идею могут быть внутренние и внешние источники.

К внешним источникам идей мы можем отнести:

- 1) Новые технологии и новые знания.
- 2) Изменения рыночной ситуации (рынок созрел).
- 3) Изменение законодательства, норм и правил.

4) Изменение интересов кредиторов.

К внутренним источникам идей мы также можем отнести:

1) Избыточные ресурсы.

2) Инициатива предпринимателей.

3) Собственные исследования, разработки, инициативы (имеется в виду коллектива разработчиков).

4) Интересы акционеров и собственников.

5) Производственная необходимость.

Идей всегда великое множество. Но перед тем, как начинать работу, нужно провести предварительную экспертизу.

Задача предварительной экспертизы - исключение из дальнейшего рассмотрения заведомо неприемлемых идей. Среди критериев оценки затраты, риск, доступность необходимых ресурсов (в том числе и интеллектуальных).

Методов экспертной оценки тоже много, но особо распространена экспертная оценка осуществимости проекта (метод весовых коэффициентов).

1) эксперт выделяет факторы, существенно влияющие на осуществимость проекта;

2) ранжирует эти факторы (располагает в порядке убывания их важности);

3) оценивает весомость (ранг) каждого фактора. Сумма рангов должна равняться 1;

4) Оценивает каждый фактор каждого инвестиционного замысла в баллах (от 0 до 100);

5) Результат оценки – это сумма произведений веса каждого фактора на его оценку.

Например, в результате экспертной оценки мы можем получить следующую картину проекта.

№	Фактор влияния	Весовой коэффициент	Оценка фактора для инвестиционного замысла №			Интегральная оценка для инвестиционного замысла №		
			1	2	3	1	2	3
1	Спрос	0,33	90	80	60	30	26,7	20
2	Ресурсы	0,27	65	75	100	17,3	20	26,7
3	Интеллектуальные права	0,17	20	80	100	3,3	13,3	16,7
4	Лицензирование	0,13	0	0	100	0	0	13,3
5	Затраты	0,1	100	60	10	10	6	1
	Всего:	1				60,7	66	77,7

Весовые коэффициенты можно получить, используя балльную оценку факторов, оценив каждый фактор по 10 балльной системе и поделив оценку на сумму всех баллов.

Тогда, сложив все и умножив на коэффициенты, мы получим сводную таблицу.

Спрос	10	0,33
Ресурсы	8	0,27
Интеллектуальные права	5	0,17
Лицензирование	4	0,13
Затраты	3	0,1
Сумма	30	1

Другим важным объектом внимания экспертов перед стартом проекта является совокупность целей и задач проекта. Например:

- 1) Наличие альтернативных решений и их популярность.
- 2) Спрос на продукт у аналогов.
- 3) Возможная продолжительность проекта.
- 4) Прогнозируемая цена на продукт.
- 5) Перспектива распространения (модель распространения) и охвата аудитории.
- 6) Степень сложности и реализуемость проекта.
- 7) Наличие юридических возможностей.
- 8) «Инвестиционный климат».
- 9) Соотношение прогнозируемых затрат и прогнозируемых результатов.

Все сгенерированные идеи и детали замысла оформляют в отдельный документ, который называется **декларация о намерениях**.

С этим документом должны ознакомиться все заинтересованные лица, включая инвесторов, заказчика и основных исполнителей.

Другим важным этапом подготовки проекта являются прединвестиционные исследования. По сути, это анализ факторов, влияющих на проект, которые также могут изменяться с течением времени. Диапазон возможных изменений этих факторов может быть чрезвычайно широк. Для уменьшения неопределённости прибегают к прогнозированию.

В состав прединвестиционных исследований входят различные области деятельности. Это могут быть прогнозы социально-экономического развития города, области, региона или страны. Могут быть прогнозы развития отрасли, прогноз демографический, прогноз деловой активности, покупательской способности населения и так далее.

При прогнозировании основываются на Декларации о намерениях и после прединвестиционных исследований формируют **обоснования инвестиций**.

Главная цель такого обоснования – это предоставление информации, достаточной для принятия инвесторами решения о целесообразности дальнейшей работы над идеей.

Напомним, что на этом этапе самого проекта еще не существует, - он находится на стадии формирования идеи и ряда исследований экономического и политического характера.

Обычно обоснования инвестиций содержат:

- 1) Резюме проекта.
- 2) Общую характеристику отрасли.
- 3) Исходные данные прогнозов.
- 4) Анализ рынка.
- 5) Процедуру управления данным проектом и основные тезисы этого процесса.
- 6) Анализ эффективности.

После формирования обоснований инвестиций проводят **проектный анализ**.

Проектный анализ имеет одну цель – определить ценность вариантов реализации проекта.

При этом используют простую формулу:

$$\text{Ценность} = \text{Изменение выгоды проекта} - \text{Изменение затрат}$$

При этом результат проекта равен произведению прироста объема реализации на цену единицы продукции, а затраты за год равны приросту объема требуемых ресурсов, умноженному на издержки на единицу продукции.

Экономисты различают семь основных видов проектного анализа.

- 1) Технический (с точки зрения реализуемости проекта, его масштаба, расположения и доступности ресурсов).
- 2) Финансовый (с точки зрения движений финансовых потоков).
- 3) Коммерческий (с точки зрения конечного потребителя).
- 4) Экологический (с точки зрения влияния на окружающую среду).
- 5) Организационный (с точки зрения организационной структуры, планирования и работы с персоналом).
- 6) Социальный (с точки зрения социокультурных и демографических последствий внедрения, организации населения, а также приемлемости для населения, культуры и религии).
- 7) Экономический (влияние на экономику разных уровней).

Также при работе, эксперты рассматривают **финансовую реализуемость проекта**, то есть определенный показатель («да» или «нет»), который характеризует наличие финансовых возможностей осуществления проекта.

Для любого коммерческого проекта показатель финансовой реализуемости является важным, так как на его основе можно сделать вывод о том, есть ли на общих основаниях «смысл» заниматься проектом или нет. Также требование финансовой реализуемости автоматически определяет необходимый объем финансирования проекта.

При этом, понятие реализуемости и объем финансирования становятся зависимыми понятиями и показателями.

При выявлении финансовой нереализуемости, схема финансирования, отдельные элементы экономического и организационного механизма могут быть изменены, если есть другие «критерии» необходимости реализации проекта.

Финансовая реализуемость проверяется для совокупного капитала всех участников проекта. Денежные потоки, поступающие от каждого участника в проект, являются в этом случае притоками (и берутся со знаком «плюс»), а потоки, поступающие к каждому участнику из проекта, — оттоками (со знаком «минус»).

Помимо этого, рассматривается денежный поток самого проекта (в данном случае сумма потоков от выручки и прочих доходов — это притоки, записываемые со знаком «плюс», плюс инвестиционные и производственные затраты, не считая налогов, — оттоки, записываемые со знаком «минус»).

Например, поток «выручка от реализации продукции» (с учетом НДС, акциз, пошлин, наценок) отображается положительным числом, а поток «производственные затраты» (со всеми сопутствующими статьями), — отрицательным.

В результате получается совокупный денежный поток, анализ которого говорит о том, стоит ли заниматься проектом или нет, а также оценивать его жизнеспособность вообще.

Для оценки жизнеспособности проекта сравнивают варианты проекта с точки зрения их стоимости, сроков реализации и прибыльности. В результате такой оценки инвестор (заказчик) должен быть уверен, что на продукцию, являющуюся результатом проекта, в течение всего жизненного цикла будет держаться стабильный спрос, достаточный для назначения такой цены, которая обеспечивала бы покрытие расходов на эксплуатацию и обслуживание объектов проекта, выплату задолженностей и удовлетворительную окупаемость капиталовложений.

Работа по оценке жизнеспособности проекта обычно проводится в 2 этапа:

- 1) из альтернативных вариантов проекта выбирается наиболее жизнеспособный;
- 2) для выбранного варианта проекта подбираются методы финансирования и структура инвестиций, обеспечивающие максимальную жизнеспособность проекта.

Как уже говорилось выше, на стартовом этапе проекта эксперты также занимаются **техничко-экономическим обоснованием** проекта (ТЭО). Мы можем охарактеризовать технико-экономическое обоснование как объект (документ, портфель документов), по которому принимаются основные решения: технологические, объемно-планировочные, конструктивные, природоохранные. По ТЭО достоверно оценивается экологическая, санитарно-эпидемиологическая и эксплуатационная безопасность проекта, а также его экономическая эффективность и социальные последствия.

ТЭО обычно состоит из следующих элементов или разделов:

- 1) пояснительная записка;
- 2) генеральный план;
- 3) описание технологических решений;
- 4) детали управления производством;

- 5) архитектурные решения;
- 6) описание необходимого оборудования;
- 7) организация процесса разработки или производства;
- 8) вопросы охраны окружающей среды (а также безопасность жизнедеятельности и анализ условий производства);
- 9) вопросы предупреждения чрезвычайных ситуаций;
- 10) проектная и сметная документация;
- 11) анализ эффективности инвестиционной деятельности;
- 12) и так далее.

Разработка ТЭО может выполняться собственными силами или специальными организациями, которые являются профессионалами в этой области (это достаточно длительный и трудоемкий процесс). В любом случае, если в результате выполненного в ТЭО инвестиций анализа выявлена нецелесообразность инвестирования средств в строительство намечаемого объекта, стоимость разработки ТЭО списывается на убытки заказчика в установленном порядке. После утверждения ТЭО и принятия инвестиционного решения заказчик обращается в орган местного самоуправления, обладающий правом изъятия и предоставления земельных участков, с ходатайством об изъятии предварительно согласованного земельного участка и предоставлении его для строительства объекта.

Если ТЭО «дает добро» на дальнейшую работу, то составляется **бизнес-план** проекта.

Бизнес-план — это подробный, структурированный и тщательно подготовленный документ, описывающий цели и задачи, которые необходимо решить предприятию (компании), способы достижения поставленных целей и технико-экономические показатели проекта в результате их достижения. В нем содержится оценка текущего момента, сильных и слабых сторон проекта, анализ рынка и информация о потребителях продукции или услуг.

Бизнес-план в основном предназначен для менеджеров проекта, собственников и кредиторов (инвесторов). Для последней категории наличие бизнес-плана вообще является одним из основных критериев «серьезности» проекта.

Состав бизнес-плана и степень его детализации зависят от размеров будущего проекта и сферы, к которой он относится. Если предполагается наладить производство нового вида какой-либо продукции, то должен быть разработан весьма подробный план, диктуемый сложностью самого продукта и сложностью рынка этого продукта. То есть бизнес-план зависит от продукта и от рынка сбыта.

Примерный состав бизнес-плана следующий:

- 1) Вводная часть.
- 2) Анализ положения дел в отрасли.
- 3) Сущность проекта.
- 4) Анализ рынка.
- 5) Маркетинговый план.
- 6) Производственный план.

- 7) Организационный план и план управления персоналом.
- 8) Оценка рисков проекта.
- 9) Финансовый план проекта.
- 10) Приложения.

Хорошо. С «началом» проекта мы определились, давайте теперь рассмотрим основные методы технического обеспечения проекта, тем более что для успешной реализации проектов, - он должен быть обеспечен в достаточном объеме нужными ресурсами.

12.2. Методы технического обеспечения проекта

Для нормального функционирования проектируемого производства важным условием является обеспечение оборудованием, материалами, сырьем и другими ресурсами.

Говоря общими словами из экономической теории, материально-техническое обеспечение инвестиционного проекта — это сложный процесс, который включает: систему материально-технического обеспечения проекта, определение потребности в ресурсах, организации сбалансированности их объемов с производством, размещение заказов на поставки и осуществления их в сроки, подчинены непрерывности инвестиционного процесса.

Для IT-проекта это тоже важный вопрос, несмотря на то что основными ресурсами для такого типа инвестиционных проектов являются человеческие ресурсы. Нужны еще программы, лицензии, оборудование, связь и так далее...

Материально-техническое обеспечение инвестиционного проекта осуществляется через систему закупок и поставок. Во закупками и поставками понимают мероприятия направлены на обеспечение проектов ресурсами - есть имуществом (товарами), выполнением работ (услуг), передачей результатов интеллектуального капитала в связи с реализацией конкретного инвестиционного проекта.

Даже если IT-проект не содержит «закупок» и «поставок» напрямую, они все-равно существуют их аналоги, хотя бы в «цифровом» виде, так что мы можем свободно оперировать этими терминами и в нашем случае.

Закупки мы можем разделять на:

- 1) закупки работ (подряды и субподряды);
- 2) закупки материалов;
- 3) закупки оборудования;
- 4) закупки дополнительных услуг;
- 5) закупки услуг консультантов.

Есть прямые закупки, при которых создаются правовые связи между двумя субъектами закупок через заключение договора поставок.

Есть посреднические закупки, при которых разработчики самостоятельно или через посредников (может быть на биржах), заключают косвенные контракты на поставку требуемых ресурсов.

На стадии разработки проекта (планирование) создается модель технологической комплектации. В составе планов проекта разрабатывается унифицированная нормативно-технологическая документация (УНТД) - это

комплекс документов, который является нормативной базой производственно-технологической комплектации.

Важным в процессе обеспечения материально-техническими ресурсами проектов является определение потребности в материально-технических ресурсах для реализации инвестиционных проектов, который проводится по следующим направлениям:

- 1) определяются виды и количество «сырья», которое «перерабатываются» во время работы над проектом;
- 2) изучается рынок материалов и сырья, а также комплектующих изделий, подбираются условия поставки, находятся лучшие поставщики по разным критериям;
- 3) предполагаются альтернативные варианты обеспечения проекта по каждому виду ресурсов;
- 4) рассчитываются возможные потери сырья или материалов и так далее.

Для разработки программного обеспечения «материалами» или «сырьем» может считаться любой расходник или оборудование, которое имеет ограниченный срок эксплуатации, включая канцелярию. Например, для разработки мобильных приложений нужно постоянное пополнение «тестового стенда» мобильных устройств различных фирм, годов выпуска, версий операционной системы, разрешения экрана и так далее. Их приходится менять как минимум раз в год, так как старые устройства уходят с рынка и замещаются новыми. Такие устройства мы точно можем отнести к категории расходных материалов, как и кабели/зарядки к ним.

Потребности в материально-технических ресурсах для реализации инвестиционных проектов определяются, анализируются и уточняются на стадии подготовки ТЭО как в количественном, так и в качественном виде. При выполнении этой работы необходимо рассматривать социально-экономические, финансовые и технические факторы, которые могут сильно повлиять на виды, количество и качество ресурсов. К ним, в частности, относятся:

- 1) социально-экономические факторы: социальная и культурная среда, социально-экономическую политику и регулирование, инфраструктурная сеть, система транспорта и коммуникаций;
- 2) коммерческие и финансовые (экономические) факторы: размер проекта, квалификация и производительность труда персонала, требования рынка к качеству продукта, материалов, сырья и т.д.;
- 3) технические факторы: сфера промышленности, технология и технологический процесс, тип оборудования, производственные или вычислительные мощности и расчетный объем (скорость) производства.

Материально-техническое снабжение инвестиционных проектов осуществляется по графику, четко связанному со сроками выполнения работ.

Процесс обеспечения проекта нужными ресурсами и материально-технической базой поддается расчету.

12.3. Методы расчета обеспеченности проекта

Расчет обеспеченности проекта является частью процесса **управления ресурсами проекта**. Включает процессы планирования, закупок, поставок, распределения, учета и контроля ресурсов (трудовых и материально-технических).

В принципе понятие ресурс в методологии управления проектами трактуется широко: все, чем располагает проект, — в том числе трудовые, финансовые и материально-технические ресурсы, команда проекта, время (продолжительности, сроки ограничения), информация, знания и технологии — является взаимосвязанными ресурсами IT-проекта. И основная задача управления ресурсами — обеспечить их оптимальное использование для достижения конечной цели управления проектом — результата проекта с запланированными показателями.

Мы можем выделить два основных типа ресурсов проекта:

- 1) невозпроизводимые, которые расходуются во время выполнения проекта и не допускают повторного использования;
- 2) воспроизводимые, которые сохраняют свою натурально-вещественную форму в ходе работы над проектом и могут быть задействованы на других работах.

Понятие ресурсов взаимосвязано с понятием «работа», поскольку ресурсы соотносятся не с проектом в целом, а с определенными работами, выполняемыми в запланированной последовательности, соответствующей календарному плану работ по проекту.

В основе планирования и учета ресурсов в ходе выполнения проекта лежит принцип необходимости сопоставления функций наличия и потребности в ресурсах.

В каждый текущий момент времени ресурсы проекта ограничены, и потому основными задачами управления обеспеченностью проекта ресурсами являются:

- 1) оптимальное планирование ресурсов;
- 2) управление материально-техническим обеспечением;
- 3) управление закупками ресурсов;
- 4) управление снабжением;
- 5) управление поставками ресурсов;
- 6) управление запасами ресурсов;
- 7) управление распределением ресурсов по работам проекта.

В общем случае получается следующая структура материально-технического обеспечения проекта.



Для расчета обеспеченности проекта применяют следующие понятия.

Под закупками понимают деятельность, направленную на обеспечение проектов следующими сущностями:

- 1) ресурсами (имуществом);
- 2) выполнением работ (услуг);
- 3) передачей результатов интеллектуального творчества в связи с конкретным проектом.

Закупки и поставки взаимосвязаны и, по сути, являются двумя сторонами процессов материально-технического обеспечения проекта.

Закупками «заведует» подсистема предприятия **управление закупками**. Она включает в себя процессы приобретения товаров, услуг, продукции от внешних поставщиков. Подпроцессы управления закупками:

- 1) планирование;
- 2) выбор поставщика;
- 3) заключение контракта;
- 4) проведение закупки;
- 5) обеспечение поставки;
- 6) завершение контракта.

Также с управлением закупками связаны следующие подсистемы компании:

- 1) планирование поставок;
- 2) бухгалтерский учет;
- 3) доставка;
- 4) приемка и постановка на учет;
- 5) хранение (и обеспечение сохранности);
- 6) учет и контроль;
- 7) списание;
- 8) утилизация.

В общем говоря, мы можем выделить следующие основные принципы обеспечения проекта необходимыми ресурсами:

- 1) планирование ресурсной обеспеченности при ограничении во времени;
- 2) планирование при ограниченных ресурсах.

Первый подход предполагает фиксированную дату окончания проекта и назначение на проект дополнительных ресурсов на периоды перегрузок.

Второй подход предполагает, что первоначально заданное количество доступных ресурсов не может быть изменено и является основным ограничением проекта.

Также в компаниях существует множество связанных процессов, но они больше относятся к сферам экономики, бухгалтерского учета, менеджмента и маркетинга. Нам важно понять, что невозможно создавать IT-проекты в «изолированной» среде от перечисленных выше направлений деятельности, именно поэтому мы применяем экономические методы и категории в теории управления IT-проектов.

Здесь хотелось бы отметить еще несколько интересных моментов связанных с тем, что основная деятельность по производству программного обеспечения связана с эксплуатацией вычислительной техники и цифровых ресурсов.

Поэтому дальше мы рассмотрим актуальный вопрос, связанный с амортизацией и устареванием оборудования, что очень хорошо заметно при работе в самых передовых и современных сферах.

12.4. Амортизация и устаревание оборудования

По экономической теории, в бухгалтерском учете срок полезного использования (СПИ) основных средств (ОС) устанавливается организацией самостоятельно с учетом следующих факторов:

- 1) ожидаемый срок использования в соответствии с ожидаемой производительностью или мощностью;
- 2) ожидаемый физический износ, который зависит от режима эксплуатации, естественных условий, влияния агрессивной среды и системы проведения ремонта;
- 3) нормативно-правовые и другие ограничения использования объекта.

В первую очередь для IT-проектов амортизация касается вычислительной техники и мобильных устройств.

Учитывая, что амортизация начисляется в течение срока полезного использования, то можно сказать, что срок амортизации компьютера — свыше 2 лет до 3 лет включительно.

При этом такой срок может быть установлен не только для компьютеров, но и иного аналогичного оборудования. Это значит, что этот же срок амортизации компьютерной техники может быть установлен и для принтеров, серверов, сетевого оборудования, локальных вычислительных сетей и т.д.

Напомним, что если стоимость компьютера в бухучете не более 40 000 рублей, он может вообще не признаваться объектом ОС и, соответственно, не амортизироваться, а единовременно списываться на расходы проекта.

В налоговом учете объекты не дороже 100 000 рублей должны учитываться как материалы. Это обязанность, а не право организации.

Приведем пример расчета амортизации ноутбука при использовании наиболее распространенного метода в бухгалтерском и налоговом учете – линейного.

Так, срок амортизации ноутбука первоначальной стоимостью 138 000 рублей установлен равным 25 месяцев. Учитывая такой срок эксплуатации ноутбука, при начислении амортизации ежемесячно на расходы организации будут относиться 4% его первоначальной стоимости ($1 / 25 * 100\%$).

Поэтому на систематической основе в учете организации будет признаваться ежемесячная амортизация в размере 5 520 рублей ($138\ 000 * 0,04$).

Есть, конечно, случаи, когда ЭВМ работает гораздо больше предложенного срока амортизации. Например, в случае качественного и дорогого оборудования. Например, в 2022 году «в руках» дизайнеров прекрасно работают компьютеры Mac производства 2011 г. Однако они не функциональны для системного программиста или для DevOps или iOS-разработчика, так как его процессор не поддерживает нужные инструкции.

Те же проблемы должны быть решены и для программного обеспечения: оно тоже «устареваает» и теряет актуальность.

12.5. Лицензии и истечение лицензий на программное обеспечение

Лицензия на программное обеспечение — это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом. Обычно лицензия на программное обеспечение разрешает получателю использовать одну или несколько копий программы, причём без лицензии такое использование рассматривалось бы в рамках закона как нарушение авторских прав издателя. По сути, лицензия выступает гарантией того, что издатель ПО, которому принадлежат исключительные права на программу, не подаст в суд на того, кто ею пользуется.

Есть свободные и проприетарные лицензии.

Свободные лицензии позволяют разрабатывать свободное (бесплатное, с открытым исходным кодом) и коммерческое программное обеспечение (ПО). Для многих компаний такое ПО просто необходимо, так как затраты на приобретение программ могут быть существенными.

Например, лицензия на ОС Windows Pro стоит порядка 300\$, а лицензия на комплект разработчика JetBrains – около 130 \$ (на каждого пользователя в 1 год использования). Так что «на старте» компании со штатом в 10 человек, в которой разрабатывается программное обеспечение для ОС Windows, в первый год нужно вложить около 4 500 \$ - 5 000 \$ только на программное обеспечение. Плюс еще нужны:

- 1) аренда серверов для размещения и сборки проекта;
- 2) аренда серверов с базами данных;
- 3) «продвинутые» аккаунты на таких системах, как Atlassian JIRA и Gitlab;
- 4) облачные системы;

5) системы коммуникации типа Slack и так далее.

Тогда сумма выплат за программное обеспечение и лицензии в первый же год может достигать 10 000 \$ для нашей небольшой гипотетической компании. Крупные компании платят сотни тысяч долларов в год или даже в месяц за те же самые продукты, но за большие масштабы.

Естественно, что компании стараются снизить такие расходы, используя открытое программное обеспечение, если это возможно.

Например, вместо покупки ОС Windows, веб-разработчики используют бесплатный Linux, а вместо JIRA используют Trello, разворачивают системы сборки контейнеров приложений на своих машинах, а не в Gitlab и пользуются бесплатными программами типа Microsoft Visual Studio Code вместо JetBrains WebStorm.

Лицензия программного обеспечения играет большую роль в этом процессе. Их существует множество видов, но давайте рассмотрим одну из самых популярных открытых лицензий, - GNU GPL.

GNU GPL или GNU General Public License, лицензия на свободное программное обеспечение, созданная в рамках проекта GNU в 1988 г., по которой автор передаёт программное обеспечение в общественную собственность.

Цель GNU GPL — предоставить пользователю права копировать, модифицировать и распространять (в том числе на коммерческой основе) программы, а также гарантировать, что и пользователи всех производных программ получат вышеперечисленные права.

GPL предоставляет получателям компьютерных программ следующие права, или «свободы»:

- 1) свободу запуска программы с любой целью;
- 2) свободу изучения того, как программа работает, и её модификации (предварительным условием для этого является доступ к исходному коду);
- 3) свободу распространения копий как исходного, так и исполняемого кода;
- 4) свободу улучшения программы и выпуска улучшений в публичный доступ (предварительным условием для этого является доступ к исходному коду).

В общем случае распространитель программы, полученной на условиях GPL, либо программы, основанной на таковой, обязан предоставить получателю возможность получить соответствующий исходный код.

12.6. Облачные системы и способы размещения

Ну и естественно, мы не можем обойти такое современное направление в разработке программного обеспечения, как облачные системы.

Под облачными системами или сервисами мы будем понимать модели удаленного доступа к инфраструктуре, платформам и программному обеспечению. С помощью облачных сервисов компании высвобождают капитал, оптимизируют обслуживание ИТ-сред, модернизируют и

масштабируют бизнес-подходы, повышают безопасность и гибкость услуг и решений, помогают заказчикам, используя новые пути и варианты, и развивают свой бизнес в постоянно меняющихся рыночных условиях.

«Облака» делятся на следующие типы:

- 1) Частное облако.
- 2) Общедоступное облако.
- 3) Гибридное облако.

В модели частного облака рассматриваются ресурсы облачных сервисов, используемые исключительно одним предприятием или организацией. Частное облако может располагаться в локальном центре обработки данных или в облачной среде поставщика услуг.

Когда организация разворачивает модель общедоступного облака, все оборудование, программное обеспечение и вспомогательная инфраструктура принадлежат поставщику услуг и управляются им, а услуги предоставляются исключительно через Интернет. Для доступа к этим сервисам и управления своей учетной записью используется веб-браузер.

Модель гибридных облачных сервисов объединяет общедоступные и частные облака для совместного использования данных и приложений. Гибридные облака связывают инфраструктуру и приложения в облачных ресурсах с существующими ресурсами, не расположенными в облачной среде. Такой подход дает предприятию больше гибкости и возможностей развертывания.

Сейчас облака используются для решения множества задач, например:

- 1) Организация хранения данных с разграничением доступа.
- 2) Системы документооборота.
- 3) Системы высокопроизводительных вычислений.
- 4) Системы массового обслуживания.
- 5) «Фасады», за которыми скрываются сложные системы управления.
- 6) Инструменты для совместной работы.

В облачную среду все больше компаний перемещает свои активы в виде данных и программного обеспечения.

Существуют даже редакторы исходного кода и среды разработки ПО на базе облачных систем, - они позволяют работать с одним проектом сразу нескольким разработчикам с разделением ресурсов, фиксацией изменений и даже содержат в себе площадки для сборки и запуска программ. При этом, на машине разработчика может не быть даже редактора кода – нужен только современный браузер и Интернет.

Ну и, конечно, современные облачные системы сами по себе содержат множество полезных инфраструктурных объектов и сервисов. Например, сейчас распространено несколько типов облаков:

- 1) Amazon AWS.
- 2) IBM Cloud.
- 3) Azure.
- 4) Google Cloud.

Также есть отечественные облачные решения от Mail.ru, Яндекс и так далее.

AWS, например, содержит в себе множество готовых решений типа баз данных, серверов приложений, систем масштабирования и управления контейнерами виртуальных машин, платформ для бессерверных вычислений и так далее. Все они либо бесплатны, либо работают по модели «платишь только за те ресурсы, которые потребляешь».

Так что процесс перевода программного обеспечения в облака – только начинается и многие компании уже сделали свой выбор.

Тема № 16. Организация и формирование проектной команды

16.1. Командные стратегии

Когда компания работает над IT-проектом, в центре активности всегда находится команда. Команда разработки ставит и решает задачи, разбирается с проблемами, приходит к определенным решениям, делает выводы, приобретает компетенции и так далее. Подбор правильной команды под особенности проекта – уже половина успеха.

Компании заинтересованы в хороших командах, которым комфортно работать вместе. Такие команды нужно сохранять и развивать различными способами: от организации личностного роста каждого члена команды, до вовлечения команд в развитие самой организации.

Этот процесс называется стратегическим развитием команды.

Давайте рассмотрим командные цели развития.

Участие потенциальных членов команды в разработке стратегии организации, впрочем, как и иная совместная деятельность членов группы позволяет выстроить индивидуальные векторы целей в одном направлении (в направлении развития организации) и обрести большую сплоченность в совместном поиске решения стратегических проблем. Однако стратегия организации и стратегия развития команды, хотя и связаны между собой и не должны противоречить друг другу, но они отличаются друг от друга тем, что у них разные задачи, вытекающие из соответствующих целей.

Цели командной работы объясняют основу и направленность ее деятельности и должны быть декомпозированы до уровня задач конкретных и понятных членам команды. При этом члены команды должны не только одинаково понимать постановку задач, но и понимать изменение этих задач в динамике.

Установление целей команды, таким образом, служит своеобразным связующим звеном между стратегическими целями организации и индивидуальными целями ее работников, составляющих команду управления. Кроме того, принятие членом команды корпоративных целей и корректировка своих собственных устремлений снижает вероятность конфликта между целями личными и коллективными в силу большей осведомленности в причинах принятых командой целевых установок и в силу возможности аргументированного отстаивания своей позиции в процессе обсуждения.

Синергический эффект принятия командных (совместных) решений способствует генерированию оригинальных целей и идей, но реализовать указанную возможность дано не любой группе, а только команде.

Давайте рассмотрим стратегии создания и развития эффективной команда.

Как говорится в исследованиях американских специалистов, каждой команде, которая планирует функционировать какое-то время, требуется прочный фундамент, состоящий из двух важных элементов — декларации цели деятельности и директивы команды.

Под декларацией цели команды понимают формулирование конкретной цели команды и причину ее существования. Причем, цель формулируется не столько в виде образа идеального состояния команды и даже не в форме миссии, сколько декларируется короткая и понятная членам команды настолько, чтобы каждый ее мог сформулировать, не задумываясь в любое время и в любой обстановке. При этом декларацию цели следует разрабатывать вместе с командой, а не вместо команды.

Декларация цели — это состоящее из одного предложения определение причины создания и цели деятельности команды.

Что касается директив команды, то они должны подробно разъяснять ожидания членов команды, связанные с ее созданием и деятельностью. Директивы сродни правилам хорошего тона и определяют поведение и образ действий членов команды во время совместной деятельности. Директив не должно быть много (не больше десяти) поскольку они определяют скорее рамки, ограничивающие поведение членов команды, чем являются предписаниями на все случаи жизни команды.

Давайте рассмотрим пример директивы целевой команды по разработке совместных стратегий:

- 1) Члены команды обязуются посещать все собрания команды вовремя и с надлежащей подготовкой.
- 2) Обязуются выполнять все обязательства и доводить работу до конца.
- 3) Обязуются принимать активное участие в собраниях.
- 4) Обязуются помогать командам выполнять коллективную работу.
- 5) Обязуются поддерживать командные решения.
- 6) Обязуются относиться друг к другу уважительно и на профессиональном уровне.
- 7) Обязуются сохранять конфиденциальность и этику проекта.

Общими четами всех директив является позитивный тон, описание наблюдаемого поведения, значимого для всех членов команды, ориентация на действие и лаконичность. Директивы формализуют ожидания, усиливают самоорганизацию и самоуправление, помогают новым членам команды влиться в проект.

Сама по себе цель подразумевает достижение результата согласно выдвинутым критериям.

Любая директива должна содержать следующие критерии:

- 1) Ясность и четкость, корректность формулировки.

- 2) Измеримость и возможность оценки.
- 3) Ориентацию на действие.
- 4) Ориентация на результат.
- 5) Привязанность к конкретному сроку.

После того как команда определилась с целями, которые ей предстоит достичь, задача состоит в том, чтобы помочь ей разработать план выполнения работ. К разработке плана возможны два подхода: план шагов для приближения к цели или план движения от цели к исходному состоянию.

Следующим шагом будет распределение ролей и обязанностей между членами команды, дабы избежать дублирования функций и обеспечить распределение обязанностей сообразно умениям и предрасположенности членов команды. Также необходимо учитывать перспективы команды.

Для создания сплоченности следует применить стратегии объединения людей, поэтому члены команды должны постоянно встречаться для установления прочных связей между ними. Чаще всего это происходит на собраниях, на которых осуществляется не только обмен информацией, но и рассматриваются проблемы, представляющие общий интерес. Совместное обсуждение проблем и последующее выполнение заданий имеет большое значение для снятия напряженности и изменения стиля работы «в одиночку» на стиль взаимопомощи и поддержки.

Также для сплоченности команды и укрепления стратегии можно использовать несколько мероприятий, например:

- 1) Мастер-классы и семинары.
- 2) Теоретические форумы.
- 3) Тренинги.
- 4) Перекрестное обучение.
- 5) Индивидуальное обучение.

Это мероприятия, нацеленные на развитие командной стратегии в компании и повышение эффективности командной работы.

Однако, единство и сплоченность команды не следует путать с единомыслием, нивелирующим разнообразие подходов, точек зрения и аргументов в процессе поиска решений стратегических и текущих проблем команды.

Сплоченность отличается от конформизма тем, что проявляется в общности мыслей и дел членов команды, а не в преднамеренном замалчивании, игнорировании, нежелании слышать и видеть реальные события и воспринимать творческие, нестандартные идеи.

Сплоченность коллектива в командной стратегии развития может формироваться с помощью обычного живого общения за чашкой кофе, во время совместных обедов, перерывов на работе, совместных походов и корпоративов, а также спортивных событий.

В любом случае, повышение эффективности работы команды положительно сказывается на всей компании и помогает более эффективно осуществлять управление проектной командой.

16.2. Управление проектной командой

Управление проектной командой – это сложный и многогранный процесс. Недостаточно сплоченные команды напрасно тратят энергию и рабочее время. Каждый член команды может работать «на износ», но общие результаты не будут хороши. Слаженность команды – это необходимое условие для роста компании, при этом потенциал компании кроется в синергии.

Взаимодействие в группе создает атмосферу творческого напряжения и обеспечивает лучшие условия для системного мышления. А создание интеллектуальной модели взаимодействия, которая формирует и направляет восприятия и действия членов команды, обеспечивает постоянное взаимное проникновение действительности и сознания членов команды.

Управление проектной командой частично реализуется через управление внутренней интеграцией. Внутренняя интеграция, это процессы, которые сопровождают команду, отражая ее внутренние проблемы, с которыми команда сталкивается на разных этапах своего жизненного цикла.

В этот процесс входят:

- 1) Выработка языка команды и концептуальных понятий группы.
- 2) Определение групповых границ и критериев для включения и исключения новых членов группы.
- 3) Распределение служебных полномочий.
- 4) Разработка норм доверительных и дружеских отношений.
- 5) Определение и распределение поощрений и наказаний.
- 6) Создание идеологии команды.

Посредством диалогов и совместных мероприятий, повышается эффективность процесса внутренней интеграции команды.

Выработка общего языка команды важна, потому как если участники группы не смогут общаться и понимать друг друга, ее создание становится невозможным по определению.

Определение групповых границ важно так как группа должна как-то определить самое себя. Здесь решаются вопросы: кто входит и кто не входит в нее, в чем состоит критерий принадлежности к ней.

Распределение полномочий важно потому как каждая группа должна разработать собственную иерархию, а также критерии и правила, занятия и потери определенных иерархических позиций и прав работниками. Консенсус в этом вопросе способствует снижению возможных агрессивных настроений в коллективе.

Разработка норм отношений проводится в каждой группе по-своему между сотрудниками, стоящими на одной или на соседних иерархических ступенях. Организационные задачи должны решаться в атмосфере открытости и взаимного доверия.

По поводу определения и распределения поощрений и наказаний, можно сказать, что каждый сотрудник должен быть в курсе как они назначаются и в чем состоят.

Идеология команды сродни некоторой «религии». Каждая группа, подобно любому сообществу, может сталкиваться с необъяснимыми явлениями, которые должны быть как-то объяснены ее членам, дабы у них не возникло ощущения, что они «имеют дело с чем-то непостижимым и неподконтрольным».

По мере развития и формирования специфической истории команды, у ее членов возникает ее собственный язык, состоящий из слов со специальным значением, своеобразный жаргон, который, с одной стороны, облегчает членам команды быстрое понимание друг друга, а с другой стороны, позволяет команде представлять себя внешней среде.

Часто нормы доверительных отношений в команде формируются под воздействием тех моделей поведения, на которых воспитан лидер команды, задающий тон в этом процессе.

Кроме того, нормы отношений в команде восходят к таким феноменам, как привязанность, преданность, любовь и сексуальность, которые постепенно складываются в группе и являются отражением основополагающих представлений влиятельных ее членов, а также ее реального опыта командного взаимодействия.

Таким образом, возможность нормальной совместной работы и сосредоточения на решении основной задачи определяется наличием в команде консенсуса по обсуждаемым выше вопросам.

16.3. Гибкие методологии

За всю историю проектного управления было создано множество различных методов управления проектами под практически любые нужды.

Главное понять, что самое важное для Вашего проекта – дедлайны, ресурсы, соблюдение процесса, или сразу несколько факторов – а затем выбрать метод управления проектом, ориентированный на достижение этого показателя.

Прежде чем приступить к рассмотрению самых популярных методов, определим некоторые ключевые термины.

Agile – гибкий итеративный и инкрементальный подход к управлению проектами и продуктами. Подход ориентирован на работу с требованиями заказчика и их реализацию в результате постоянного взаимодействия на базе самоорганизующихся групп из специалистов разного профиля. Термином Agile названа одна из самых известных гибких методологий разработки программного обеспечения. Также существуют такие альтернативы как Scrum и Kanban.

Критический путь – последовательность событий от начала и до конца, требующая наибольшего времени для ее выполнения. По критическому пути определяется цепь наихудшего сценария развития проекта.

Цепочка событий или процессов (ЕРС-диаграмма) – это диаграмма, которая отражает последовательность реализации работ проекта на основе анализа доступности ресурсов.

Резерв времени – это время, на которое может быть отложено выполнение тех или иных работ без существенного ущерба для проекта. На критическом пути такого резерва нет, так как это самый длинный путь развития.

Ресурсы проекта – это необходимые компоненты для реализации проекта: время, оборудование, материалы, устройства, сотрудники, вычислительные мощности и так далее.

Содержание проекта (Scope) – это описание работ, которые нужно выполнить, чтобы сдать проект.

Спринт (Sprint) – это период времени, интервал, рабочий цикл, длящийся от недели до месяца, в ходе которого создается очередная рабочая версия продукта, с определенными возможностями (фичами), представляющими интерес для заказчика.

«Гибкие» методологии родились из классического подхода, основанного на разбиении процесса работ на последовательные этапы. На такой линейной структуре основывается традиционное проектное управление:

- 1) Инициация.
- 2) Планирование.
- 3) Разработка.
- 4) Реализация и тестирование.
- 5) Мониторинг и завершение.

На этапе инициации руководитель и команда определяют требования к проекту. Часто это решается через совещания и «мозговые штурмы», на которых определяется, что должен представлять из себя проект.

На этапе планирования команда решает, как она будет достигать целей, поставленных на предыдущем этапе. Здесь формируется календарный план проекта, а также состав работ по нему.

На этапе разработки определяется конфигурация будущего проекта или продукта, а также технические способы достижения результата.

На этапе реализации и тестирования происходит основная работа по проекту: написание кода, его тестирование и формирование документации. Тестирование должно быть непрерывным и циклическим, так как через этот процесс продукт проверяется на соответствие требованиям заказчика и других заинтересованных сторон.

На этапе мониторинга и завершения проекта выполняется передача кода продукта заказчику и, возможно, его поддержка и обслуживание.

Эти этапы – база, на которой строятся различные методы управления проектами. Разным проектам нужны различные фазы реализации – некоторым достаточно и трёх фаз, другим гораздо больше. Иногда используется так называемый «итеративный водопад», в котором каждый этап представляет собой некий подпроект, в ходе которого задачи реализуются по фиксированным итерациям. Но суть остаётся одна – проект разбит на этапы, которые исполняются в строго определённой последовательности.

Благодаря тому, что классический проектный менеджмент строго привязан ко времени исполнения задач, как правило, заранее определённому

на этапе планирования, для реализации проектов в рамках данного подхода отлично подходят инструменты календарно-сетевое планирования.

Давайте рассмотрим несколько характеристик гибких методологий разработки проектов.

Первыми гибкими методами были: RAD (с ориентиром на качество при минимальном бюджете и ограниченном сроке), XP (экстремальное программирование с коллективным владением кодом), SCRUM (где каждый участник команды несет ответственность за результат), Kanban (визуализация этапов разработки на доске), Agile (гибкая и быстрая методология).

Agile говорит нам, что люди в команде и взаимодействие между ними важнее процессов, взаимодействие с заказчиком важнее согласований, работающий продукт должен быть на первом месте, а готовность быстро реагировать на изменения важнее утвержденного плана.

Waterfall гласит, что в такой модели разработки каждый этап – это отдельный подпроект, документация важнее продукта, ответственность за результат лежит в целом на команде, заказчик, как и владелец продукта не участвует в рабочем процессе.

Scrum говорит о том, что каждый из участников команды ответственен за результат, а самоорганизация становится приоритетной для формирования рабочего процесса и нужно быть готовым к максимальной гибкости.

Kanban требует полной визуализации рабочего процесса с поэтапным перемещением задач от стадии «принято в работу», к стадии «готово» через промежуточные стадии.

Следует отметить, что, хотя Scrum и Kanban являются гибким подходом к управлению проектами, но Kanban может считаться самым гибким потому, что он допускает внезапное поступление новых задач с переключением между ними, коллективная ответственность за результат повышает эффективность работы, все незапланированные задачи будут иметь специальный статус (попадают в бэклог) и есть место для экспериментов и неопределенности в проектах и задачах. Kanban позволяет быстро адаптироваться и продолжить работы над задачей не нарушая рабочий процесс.

Также в современных условиях часто соединяют несколько методологий в один процесс, например используют Agile как основную методологию, Scrum используют для организации системы общения, а подход Kanban для визуализации процесса работы над задачами при помощи карточек задач.

16.4. Системы карточек заданий

Система карточек задач или заданий помогает визуализировать рабочий процесс. Эта технология впервые официально появилась в методологии Kanban. Как уже говорилось выше, Kanban проводит визуализацию и обеспечивает поэтапное перемещение задачи от «Принято в работу» (например) до «Готово». Между этими двумя станциями может быть еще несколько: «Разработка», «Тестирование», «Оптимизация» и т.д. Kanban визуально представляет собой доску, по которой мы перетягиваем однотипные

задачи со станции на станцию. И когда задача приходит на конечную станцию «Готово» — она завершена.

Как правило, при использовании Kanban метода, нет жестких ограничений по времени для каждого отдельного этапа. Но команда ориентируется на среднее время работы над каждым отдельным этапом. Чем быстрее — тем выше эффективность в целом.

Весь процесс разработки прозрачен для всех членов команды. Задачи по мере поступления заносятся в отдельный список, откуда каждый разработчик может извлечь требуемую задачу.

Kanban — наглядная система разработки, показывающая, что необходимо создать, когда и сколько. Давайте посмотрим, что такое карточка Kanban.

Каждая Kanban-карточка представляет собой отдельную рабочую задачу, проходящую по различным стадиям выполнения, которые представлены на физической или виртуальной доске Kanban.

Kanban — это японское слово, означающее «визуальный сигнал» или «сигнальная доска». Kanban-карточки были изобретены в конце 1940-х годов Тайити Оно, инженером-технологом компании Toyota, который искал способы улучшить производственный процесс.

Исследование Оно привело его в ряды американского супермаркета, где он обнаружил кое-что интересное. Вместо того чтобы постоянно набивать полки товарами до отказа и тратить много времени и усилий, работники супермаркета повторно наполняли их, только когда видели, что одна из полок почти пуста. После этого они помещали на полку достаточное количество товаров, чтобы удовлетворить текущий покупательский спрос.

Интуиция подсказывала Оно, что поставки точно в срок (JIT) произведут революцию в производственной цепочке поставок. Его первой задачей было выяснить, как визуализировать множество производственных процессов в цехах. Он сделал это с помощью Kanban-карточек.

К каждой корзине с деталями была прикреплена Kanban-карточка, которая служила для рабочих сигналом тревоги, когда количество деталей становилось меньше указанного на карточке. Когда это происходило, запускался процесс пополнения запасов, и фабрика отправляла поставщику пустую корзину вместе со своей Kanban-карточкой.

На карточке указывались сведения о спецификациях деталей, требуемом количестве и ожидаемом времени доставки. С учетом этой информации поставщик мог быстро исполнить заказ и отправить заполненную корзину с Kanban-карточкой обратно на фабрику, как раз к тому времени, чтобы производственный цикл продолжил работу без остановки.

В течение почти полувека эта система производства была «секретным ингредиентом» успеха компании Toyota. Когда в 1990-х годах компания начала публично делиться сведениями о своей системе, люди обратили на нее внимание.

Идеологи из мира бизнеса превратили производственную систему Toyota в набор процессов, известных как «бережливое производство».

Канбан-карточки быстро превратились в визуальный артефакт, отслеживающий задачи по мере их прохождения по рабочему процессу. По сути, Канбан-карточка способствует командной работе, помогая участникам общаться и наглядно делиться информацией.

Канбан-карточки позволяют командам делать следующее.

- 1) Быстро просматривать важные сведения о рабочих задачах. Каждая Канбан-карточка обычно содержит краткое описание рабочей задачи, а также информацию о ее владельце, дате выполнения и статусе. Она может содержать и другие сведения, например ссылки на исходную документацию или список задач, блокирующих продвижение этой задачи.
- 2) Плавно и эффективно передавать результаты работы. Канбан-карточки побуждают команду устанавливать четкие и предсказуемые ожидания для каждой рабочей зоны. Когда приходит время перевести рабочую задачу из одного состояния в другое (например, переместить ее с этапа планирования на этап реализации), эти четкие политики показывают, кому передается ответственность и какими будут следующие шаги.
- 3) Повышать эффективность. Канбан-карточки позволяют легко отслеживать время выполнения, то есть время, необходимое для выполнения рабочей задачи от начала и до конца. Канбан-карточки вместе с доской Канбан помогают командам выявлять узкие места в рабочем процессе и упрощать сам процесс. Большинство команд стараются сократить время выполнения, т. е. выполнять работу быстрее.

Канбан-карточки не только позволяют без труда поддерживать порядок, но и удобны в использовании. Будем честны: есть определенное удовольствие в том, чтобы отслеживать прогресс таким понятным и осязаемым способом.

Канбан-карточки имеют ряд следующих преимуществ.

- 1) Канбан-карточки позволяют легко обнаруживать неэффективную работу. Канбан-карточки невероятно наглядны. Даже одного взгляда на доску Канбан хватает, чтобы определить, в какой момент возникла проблема на одном из этапов рабочего процесса. Если у команды разработчиков программного обеспечения в столбце «Тестирование» всегда находится в два раза больше карточек, чем в столбцах «Дизайн» и «Разработка», другие участники команды быстро обратят внимание на эту визуальную очередь. Возможно, команде нужно больше тестировщиков либо требуется подправить сами критерии этапа тестирования.
- 2) Канбан-карточки упрощают поставку точно в срок. Канбан-карточки были разработаны, чтобы отслеживать уровень запасов и предотвращать складирование материалов. Производственный сектор по-прежнему использует их для этой цели, но методика «точно в срок» полезна и в других отраслях, например в разработке программного обеспечения. Накопление нереализованных

программных возможностей на одной стадии разработки ничуть не лучше, чем накопление автомобильных деталей в цехе.

- 3) Kanban-карточки способствуют совместной работе между командами. Одной из замечательных особенностей досок Kanban является то, что ими делятся. По мере продвижения карточек по столбцам они часто переходят от сотрудника к сотруднику или от команды к команде. Несмотря на то, что каждый участник несет ответственность за разные участки процесса, все равно все наблюдают одну и ту же картину; а наглядность метода Kanban позволяет легко различить рабочие процессы другой команды, пробежав глазами по столбцам и карточкам. Именно поэтому метод Kanban является хорошим противоядием от местечкового мышления.

Физические Kanban-карточки интересно брать в руки и перемещать, но что, если не все участники команды работают в одной комнате? К счастью, существует множество программных инструментов, таких как Jira Software и Trello, которые позволяют легко настраивать цифровые доски Kanban и заполнять их цифровыми Kanban-карточками.

Помимо возможности сотрудничать для удаленных команд, цифровые Kanban-карточки предлагают ряд других преимуществ по сравнению с физическими карточками. Kanban-карточки Jira и Trello включают в себя широкие возможности настройки. Например, можно отображать или скрывать на карточках выбранные информационные поля, а также добавлять динамический контент вроде комментариев, вложений и ссылок на другие карточки. В большей части программного обеспечения для Kanban можно настроить доставку уведомлений по электронной почте, когда карточку переназначают, перемещают на другой этап рабочего процесса или вносят в нее другие изменения.

Кроме того, программные инструменты помогают командам лучше понять, насколько эффективно они работают. Такие инструменты, как Jira Software, отслеживают самые важные показатели, которые Kanban-команда может использовать для улучшения своего процесса.

Каждая карточка, практически в любой системе учета, содержит следующие элементы:

- 1) Заголовок.
- 2) Уникальный номер и краткое название из заголовка (slug).
- 3) Описание задачи.
- 4) Комментарии.
- 5) Подпункты (подзадачи).
- 6) Время и дата создания.
- 7) Планируемое время и дата завершения.
- 8) Прикрепленные файлы и картинки.
- 9) Статус (в соответствии с колонкой).
- 10) Создатель задачи.
- 11) Исполнители.
- 12) Связь с другими задачами.

Дополнительные элементы карточки зависят от системы управления этими карточками, но основной состав фактически соответствует указанному выше (уже является некоторым «стандартом»).

16.5. Современные методы учета работ сотрудников

Учитывать отработанное время – обязательная часть рабочего процесса, влияющая на многие факторы, в том числе на начисление сотрудникам заработной платы. Грамотный учет рабочего времени способствует поддержанию трудовой дисциплины, рациональному перераспределению рабочих заданий, повышению производительности труда, что приводит к увеличению доходов всего предприятия и отдельных сотрудников.

Основные цели учета рабочего времени сотрудников.

Распределение должностных обязанностей осуществляется работодателем или руководителем, исходя из объемов заданий, которые возможно выполнить в установленный период времени с учетом специфики функционирования предприятия.

Продолжительность рабочего времени регламентируется Трудовым кодексом (статьей 91) и составляет 40 часов в неделю. А ориентировочный перечень рабочих обязанностей прописан в квалификационном справочнике.

Учет рабочего времени сотрудников ведется с определенной целью:

- 1) Определение эффективного времени работы сотрудника по проекту и вне проекта.
- 2) Определение трудозатрат сотрудников в общем.
- 3) Определение рисков и форс-мажоров на предприятии и на проекте.
- 4) Учет времени, затраченного на каждый проект и на каждый вид работ.

Особенно внимательно относятся к учету рабочего времени сотрудников компании, где принято платить **почасовую** оплату. Норма выработки в этом случае остается неизменной – 40 часов в неделю.

Но счетчики измеряют «чистое» время, причем с уровнем «активности» разработчика. Чем активнее человек работает с клавиатурой и мышкой, использует приложения, тем выше его уровень активности.

Здесь есть только одна проблема: «перекуры» счетчиком не учитываются, так как в это время программист не пользуется компьютером. Налить себе кофе, побеседовать с коллегами, посетить туалетную комнату – все это «неоплачиваемая работа» согласно счетчику. При этом такие счетчики очень любят зарубежные заказчики, так как намерены платить исключительно за «чистое» время, проведенное за решением задач. Обеденный перерыв, кстати говоря, тоже «не оплачивается».

В результате такого подхода время нахождения на рабочем месте у сотрудника может составлять более 10 часов, что не слишком положительно сказывается на эмоциональном и физическом состоянии сотрудника.

Однако, при учете рабочего времени сотрудника любым способом следует учитывать и возможные объективные обстоятельства, срывающие рабочий график. К ним относят:

- 1) аварии на предприятии;
- 2) недопоставку деталей или комплектующих;
- 3) стихийные бедствия.

Перечисленные объективные факторы также сказываются на количестве реализованных задач, что приводит к дисбалансу между материальными затратами (оплата труда) и объемом выполненных работ.

С этой целью проводится учет отработанных часов, позволяющий достичь нескольких целей:

- 1) Усиление рабочей дисциплины: минимизация опозданий, досрочного ухода с работы, игр на рабочем месте, переписки в соцсетях, перекуров и кофе-брейков в течение дня.
- 2) Увеличение качества и результативности труда.
- 3) Уменьшение ресурсов на заработную плату сотрудникам. Из оплачиваемого периода исключается время, на протяжении которого сотрудники не исполняют свои обязанности, даже если простой вызван технологическим сбоем или связан с нарушением соглашения третьими лицами.

Учтенная работа сотрудника должна иметь документальное оформление, причем по трудовому законодательству.

Порядок оформления учета закрепляется внутренними документами компании:

- 1) Правила трудового распорядка.
- 2) Коллективное соглашение.
- 3) Контракт о выполнении работ (оказании услуг).

Указанные документы регламентируют распорядок рабочего времени: продолжительность смены, количество перерывов, их длительность, время начала и окончания смены, количество выходных, длительность отпуска.

Это предполагаемый рабочий график. А ежедневный учет рабочего времени оформляется в виде табеля выхода сотрудников. Его задача – зафиксировать отработанное время с учетом:

- 1) сверхурочного труда;
- 2) нахождения в командировке;
- 3) болезни, подтвержденной листком нетрудоспособности;
- 4) выхода в ночь;
- 5) выходных, праздничных, нерабочих дней.

На отдельных предприятиях учетная документация ведется по упрощенной схеме или разрабатывается внутренняя кодировка рабочего времени сотрудников. Это связано с особенностью функционирования организации или спецификой труда отдельных сотрудников.

Учет рабочего времени, согласно трудовому законодательству, проводится:

- 1) ежедневно;
- 2) еженедельно;
- 3) обобщенно.

Выбор разновидности учета осуществляется исходя из специфики работы организации.

Ежедневный учет характерен для организации с одинаковой длительностью рабочих смен и проводится с учетом законодательства РФ, - по ТК, статья 94 (для сотрудников младше 16 лет, - не более 5 часов в день, от 16 до 18 лет – до 7 часов в день).

Еженедельный учет проводится согласно табелю учета организации. Если сотрудники работают по гибкому графику с нерегламентированным рабочим временем, без ведения недельного учета не обойтись. Он помогает скорректировать длительность рабочего времени для соблюдения норм, предусмотренных ТК РФ, и избежать переработок. В отдельных организациях практикуются раздробленные смены. Это связано со спецификой функционирования компании. В недельном учете рабочего времени отражают отработанные по факту часы, что позволяет вложиться в предусмотренную законодателем норму рабочего времени, избежать недоработок или сверхурочного труда.

Обобщенный учет проводится согласно статье 104 ТК РФ. Такая форма контроля рабочего времени предусмотрена для компаний, где другие формы учета вестись не могут вследствие особенностей производственного процесса предприятия. Таким компаниям законодатель предоставляет возможность суммировать отработанное время за длительный период с целью соблюдения предусмотренной ТК РФ нормы рабочих часов.

Это особенно актуально для предприятий, у которых приняты ночные или «праздничные» смены.

Кроме установленных на рабочих компьютерах индивидуальных средств учета рабочего времени, также может контролироваться рабочее время сотрудника «на предприятии в целом» при помощи следующих способов:

- 1) организация пропускного пункта на входе предприятия и входного контроля сотрудников (подразумевается наличие электронного ключа или пропуска-карты с фиксацией времени входа и выхода);
- 2) контроль руководителя над соблюдением рабочего графика его подчиненными;
- 3) самоконтроль персонала компании;
- 4) автоматизация контроля и учета времени прибытия (ухода) сотрудника при помощи специальных сканеров;
- 5) установка камер видеонаблюдения за использованием сотрудниками рабочего времени на местах и на предприятии в целом.

Порядок ведения учета предусмотрен законодательством и законодательно фиксируется табелем. Унифицированная форма табеля учета для компаний всех форм собственности закреплена Постановлением Госкомстата №1.

Табель предусматривает ежедневное заполнение вне зависимости от выбранной разновидности учета рабочего времени, - так обосновано

возможными форс-мажорными обстоятельствами, предусмотреть которые невозможно.

Существует две методики фиксации отработанных часов:

- 1) Методика полной фиксации выхода и невыхода работника.
- 2) Методика фиксации отсутствия сотрудника на рабочем месте: вследствие невыхода, опоздания, отгула, отпуска, больничного, командировки.

Сейчас эти данные фиксируются в 1С предприятия и отображаются в личном кабинете сотрудника. Там же сотрудник может контролировать свой отпуск, праздничные и нерабочие дни, размер оклада, налоговые вычеты и отчисления, а также получать различные справки и выписки.

Если штат компании невелик, табель заполняется вручную ответственным за его ведение сотрудником. В крупных корпорациях используется электронная программа контроля рабочего времени персонала. Подсчет результатов осуществляется администратором программы в автоматическом режиме и сдается в сроки, установленные внутренними документами.

Заполненный табель направляется в бухгалтерию. Заработная плата начисляется, исходя из данных про отработанное время. Для усиления контроля табель заполняется и автоматически, и вручную. Это позволяет исключить «человеческий фактор», а также технический сбой и является наиболее объективной методикой учета отработанных часов.

16.6. Методы ведения учета карточек работ

Если мы говорим о современных системах учета рабочего времени с учетом гибких методологий разработки программного обеспечения, то в этом процессе задействовано несколько процессов.

- 1) Предварительная оценка проекта с расчетом предполагаемой занятости сотрудников с почасовой и ставочной моделью оплаты.
- 2) Расчет стоимости проекта в часах и в «деньгах».
- 3) Составление плана и портфеля проекта.
- 4) Передача данных о занятости сотрудников в бухгалтерию и отдел кадров.
- 5) Текущий контроль выполнения задач и анализ активности сотрудника.
- 6) Сопоставление планируемых и фактических данных.
- 7) Разрешение конфликтных ситуаций и коррекция полученных данных.

При этом рабочее время сотрудника остается неизменным и согласуется нормативами (например, 40 часов в неделю и 160 часов в месяц).

Потом выполняется регистрация карточек заданий таким образом, чтобы на них было видно планируемое время выполнения каждой задачи, и объем выделенных таким образом задач не превышал тех самых 40 часов в неделю.

После выполнения задач работник выставляет фактически потраченное время на выполнение задач и считаются показатели эффективности.

Затраченное время соотносится с планируемым и считается объем работ в отношении 40-часовой рабочей недели, - определяется переработка или недоработка сотрудника.

В соответствии с этим, на предприятии должна использоваться либо автоматизированная система учета времени выполнения задач, принимающая параметры карточек задач, назначенных сотрудникам, либо эти операции выполняются вручную ответственным лицом, в качестве которого может быть менеджер проекта, лидер команды или специальный сотрудник, отвечающий сразу за несколько проектов.

В любом случае, система контроля времени и система карточек задач в своей совокупности должна быть источником достоверных данных для бухгалтерии и отдела кадров предприятия и работать согласно всем законодательным актам, нормам и правилам ТК Российской Федерации.

16.7. Способы синхронизации работ команды при помощи технических средств

Мы уже говорили о важности работы в команде для каждого проекта в отдельности и для всей организации в целом. И мы уже говорили про важность коммуникации между сотрудниками.

Коммуникация и синхронизация работ команды очень важны для того, чтобы не было:

- 1) дублирования функций исполнителей (два сотрудника получили одну задачу);
- 2) задержек в выполнении актуальных задач (сотрудник не имеет активных задач);
- 3) блокировок задач (когда нельзя выполнить задачу А без завершения задачи В);
- 4) задержек в выполнении технически сложных задач (сотрудник не знает, как решить проблему) и так далее.

К тому же процесс коммуникации охватывает не только разработчиков, но и других заинтересованных лиц: заказчика и представителей заказчика, инвесторов проекта, менеджеров проекта, технических лидеров, технических писателей, переводчиков, дизайнеров, тестировщиков и так далее, - у них у всех должен быть постоянный непрерывный доступ к актуальной информации.

Для решения таких проблем на проектах используются следующие инструменты:

- 1) системы управления задачами (task management);
- 2) системы учета ошибок (bugtracking);
- 3) системы контроля версий исходного кода (GIT);
- 4) системы непрерывного развертывания и непрерывной доставки (CI/CD);
- 5) облачные системы документооборота (Cloud documenting);

- 6) чаты и мессенджеры (Messengers);
- 7) интерактивные доски (Mind maps);
- 8) видеочаты и видеоконференции и так далее.

Системы управления задачами мы уже рассматривали, напомним, что это могут быть:

- 1) Trello.
- 2) JIRA Software.
- 3) ClickUp.
- 4) YouTrack.
- 5) Yandex.Трекер.
- 6) Gitlab Issue Tracker.
- 7) Asana.
- 8) Wrike.

Системы учета ошибок или bugtracking применяются для фиксации ошибок в системе и агрегации ошибок, а также для публикации решений этих ошибок. Примером может служить Bugzilla.

Системы контроля версия мы тоже рассматривали, - сейчас самыми популярными являются:

- 1) Github.
- 2) Gitlab.
- 3) Bitnucket.

Говорят, что в России скоро создадут свою собственную систему контроля версий, но подробностей пока нет.

Системы непрерывного развертывания и непрерывной доставки (CI/CD) помогают развернуть систему «в облаке», собрать контейнеры виртуальных машин, автоматически выполнить тесты, сформировать и опубликовать документацию, загрузить изменения на сервер и так далее. Сейчас наиболее популярны решения CI/CD на базе:

- 1) Gitlab CI/CD.
- 2) Jenkins.
- 3) Bitbucket pipelines.
- 4) Github Actions.

Облачные системы документооборота сейчас используются для объединения документации по проекту. Обычно это JIRA Confluence либо просто Google Documents. За рубежом также часто используются OneDrive/Dropbox, но почти никогда Яндекс и Mail документы.

В качестве чатов и мессенджеров чаще всего используются Slack, Skype, реже Telegram, Rocketchat и даже Discord. На некоторых проектах видел даже WhatsApp, но это скорее исключение из правил.

Стандартом отрасли является Slack. Он имеет не только приватные и групповые чаты и каналы, но также имеет возможность интеграции с системами управления заданиями, репозиториями, облачными системами документооборота, системами управления версиями, в нем можно создавать ботов, выполнять видео- и аудио звонки, и он работает на всех платформах.

Интерактивные доски больше относятся к системе документооборота, но мы выделим их отдельно, так как практика **ментальных карт** и **потоков** чрезвычайно эффективна при попытке понять, как работает система. В этой сфере лидируют системы типа Miro и Draw.io. В Miro мы работаем с ментальными картами и потоками, принимаем решения и собираем материалы, работая сразу всей командой. В Draw.io мы строим диаграммы связей, UML-диаграммы, структурные схемы системы, визуализируем архитектуру, описываем протоколы обмена, структурируем данные и представляем любую информацию в графическом виде.

Для общения, кроме простой переписки текстовыми сообщениями, мы должны выполнять групповые видеозвонки, работать напрямую с клиентами или заказчиком, работать с коллегами. Это лучше делать в интерактивном режиме с возможностью одновременного разговора, наблюдения за экраном коллеги, записи звонка и так далее.

В российских компаниях можно использовать Яндекс.Телемост, но в мировой практике чаще всего используется Zoom, Skype, Google Meet.

Тема № 17. Инструментальные средства сопровождения проектов. Методы деловой коммуникации.

17.1. Обзор возможностей классических систем управления проектами на основе принципов РМВОК

РМВОК является классическим трудом в современной теории управления проектами. Давайте рассмотрим его содержание, смысл и значение для управления IT-проектами.

По своей сути, РМВОК является такой своеобразной энциклопедией управления проектами, в которую включены описания методов и рекомендации к работе. Это разработка американского Института управления проектами (PMI).

Эксперты называют РМВОК классификатором управленческих процессов. Всего их 47, и эти процессы объединены в пять групп:

- 1) Инициирование.
- 2) Планирование.
- 3) Реализация.
- 4) Контроль и мониторинг.
- 5) Завершение.

Кроме того, РМВОК предлагает управленцам десять областей знания — по сути, фокусов внимания руководителя. Среди них:

- 1) управление содержанием (что должно получиться в итоге и как вносить изменения в ходе проекта);
- 2) управление временем (разработка и контроль календарного плана);
- 3) управление рисками (идентификация и методы реагирования);
- 4) управление персоналом (создание и работа в команде);
- 5) другие важные моменты теории и практики.

Этот свод знаний популярен из-за своей универсальности: он подходит для большинства проектов, вне зависимости от сферы бизнеса. Однако все же РМВОК — это не готовая методология, а скорее фреймворк для методологии, который можно (и нужно) адаптировать, подгонять по себя и только тогда уже использовать.

Кстати говоря, часто РМВОК называют стандартом, но это не так. Даже выдержка из РМВОК, утвержденная Американским институтом стандартов ANSI, которая называется «Стандарт управления проектами», может считаться «стандартом», но не сама РМВОК.

Давайте далее рассмотрим чуть подробнее содержимое идеи РМВОК через его пять основных групп процессов.

Группа процессов инициации проекта.

Состоит из процессов и подпроцессов, которые способствуют фазе начала нового проекта через проработку идеи, разработку устава проекта, идентификацию заинтересованных сторон и так далее.

Группа процессов планирования проекта.

Эта группа процессов определяет и уточняет первоначальное содержимое проекта, уточняет цели и утверждает предварительный план действий для достижения этих целей. В группу процессов планирования обычно входят следующие подпроцессы:

- 1) разработка плана управления проектом;
- 2) разработка плана управления содержанием проекта;
- 3) сбор требований и ограничений проекта;
- 4) определение содержания проекта;
- 5) определение структуры работ и их иерархии зависимостей;
- 6) формирование расписания проекта;
- 7) формирование списка необходимых операций;
- 8) определение стоимости проекта;
- 9) планирование управлением стоимостью проекта;
- 10) базовое бюджетирование проекта;
- 11) идентификация и анализ рисков проекта;
- 12) планирование закупок;
- 13) вовлечение заинтересованных сторон и так далее.

Группа процессов исполнения проекта.

Эти процессы объединяют человеческие и другие ресурсы для выполнения утвержденного плана управления проектом для создания продукта. Включает следующие подпроцессы:

- 1) непосредственное руководство проектом;
- 2) управление знаниями проекта;
- 3) управление качеством продукта;
- 4) управление ресурсами проекта;
- 5) управление развитием команды проекта;
- 6) управление коммуникациями;
- 7) оперативное управление рисками;
- 8) управление закупками;

9) коммуникация с заинтересованными сторонами.

Группа процессов контроля и мониторинга проекта.

Эти процессы необходимы для регулярной оценки прогресса проекта для того, чтобы обнаружить отклонения от утвержденного плана и сгенерировать корректирующие действия. Включает следующие подпроцессы:

- 1) мониторинг работ по проекту;
- 2) контроль изменений;
- 3) управление содержанием;
- 4) управление расписанием проекта;
- 5) управление стоимостью проекта;
- 6) управление ресурсами проекта;
- 7) управление коммуникациями проекта;
- 8) управления рисками проекта;
- 9) управление закупками проекта;
- 10) коммуникация с заинтересованными сторонами.

Группа процессов закрытия проекта.

Эти процессы формализуют приемку продукта заказчиком и заинтересованными лицами, вводят проект в последнюю фазу и готовят к правильному завершению.

РМВОК описывает следующие области знаний:

- 1) знания в области интеграции проекта;
- 2) знания в области содержания проекта;
- 3) знания в области расписания проекта;
- 4) знания в области стоимости проекта;
- 5) знания в области качества проекта;
- 6) знания в области ресурсов проекта;
- 7) знания в области коммуникаций проекта.
- 8) знания в области рисков проекта.
- 9) знания в области закупок.

Таким образом, РМВОК описывает рекомендации для множества процессов в самых разных ситуациях, но не дает четких и абсолютных «приказов», как действовать и в каких ситуациях.

Также следует отметить, что РМВОК развивается и дает новые «импульсы» к изменению современных систем управления проектами.

17.2. Современные системы управления проектами

Сейчас первичными являются электронные системы управления проектами в виде программных комплексов, запускаемых локально на машинах разработчиков и менеджеров или расположенных в сети Интернет.

Для них характерны следующие возможности:

- 1) работа из любого места и с любого устройства, где есть доступ в Интернет;
- 2) управление множеством проектов по одним и тем же принципам из одного приложения;

- 3) привлечение сотрудников для разных ролей;
- 4) управление доступом и защита данных;
- 5) управление задачами проекта (создание карточек задач, заполнение деталей, назначение ответственных, добавление дополнительных материалов);
- 6) управлением временем выполнения задач;
- 7) формирование отчетных материалов;
- 8) базовая аналитика проекта;
- 9) интеграция с другими системами и так далее.

Давайте рассмотрим далее несколько современных систем управления проектами, реализующими принципы PMBOK частично или полностью.

17.3. Простая система Trello

Trello – это простая система управления задачами проекта, основанная на принципах Kanban. Эта система объединяет функции для эффективной работы команды. Им пользуются миллионы людей на постоянной основе и не только при разработке программного обеспечения.

Основными функциями Trello являются:

- 1) Подробное описание задач в виде карт.
- 2) Простое редактирование задач и их состояний с помощью перетаскивания карточек задач.
- 3) Встроенное редактирование содержимого карточек задач.
- 4) Простая организация задач с маркировкой, тегами и комментариями.
- 5) Контрольный список показателей прогресса на каждой карточке задачи.
- 6) Архив карточек задач.
- 7) Простая загрузка документов (локальные устройства, Dropbox, Google Диск).
- 8) Вложение файлов и изображений.
- 9) Фильтры карточек задач и исполнителей.
- 10) Предупреждения и уведомления о крайних сроках по каждой карточке задачи.
- 11) Автоматические уведомления по электронной почте участникам проекта.
- 12) Журналы активности по проекту.
- 13) Индивидуальные/групповые задачи (подключение исполнителей к карточке задачи).
- 14) Резервное копирование информации.
- 15) Поиск информации по доскам проектов.
- 16) Шифрование данных SSL.
- 17) Просмотр, удобный для мобильных устройств и мобильные приложения.
- 18) API разработчика.
- 19) Служба поддержки клиентов.
- 20) Варианты голосования и оценки задач.

21) Обсуждения.

Конечно, немаловажным фактором популярности Trello является его простота. В нем нет некоторых функций, которыми обладает программное обеспечение, которое мы рассмотрим далее, но Trello практически не требует подготовки специалиста или команды чтобы начать работать сразу, - только общие технические знания и навыки работы с веб-приложениями.

17.4. Система управления досками проектов Wrike, Asana

Wrike – это корпоративный сервис в виде веб-приложения и мобильных приложений, предназначенный для работы команд и управления командными проектами. Wrike позволяет пользователям планировать проекты, управлять приоритетами задач, отслеживать график выполнения и взаимодействовать с командой.

Сервис используется десятками тысяч организаций из всех стран.

Wrike интегрирован с множеством офисных решений типа Adobe Creative Cloud, Gmail, Google Docs, Google Apps, Apple Mail, Dropbox, Office 365 и Salesforce. Также Wrike имеет открытый API для разработчиков.

Функции Wrike делятся на две основные категории: управление проектами и совместная работа. К проектным относятся возможности, которые позволяют команде визуализировать проекты (например, в форме диаграммы Ганта или канбан-доски), отслеживать сроки и потраченное время, устанавливать зависимости задач в проектах, назначать исполнителей и распределять рабочие ресурсы.

Wrike позволяет создавать шаблоны типовых проектов, описывать процесс с помощью индивидуального набора статусов и полей в задачах и автоматического назначения исполнителей на задачи с характерными признаками.

Функции совместной работы включают интеграцию с электронной почты для работы с задачами из писем, управление группами пользователей, формы входящих запросов, комментирование задач, сообщения, обсуждение, согласование и редактирование текстов, графикой и видеоматериалов, а также систему уведомлений об изменениях в проектах команды.

Asana является похожим решением в области управления проектами.

Этот проект также имеет веб-приложение и мобильное приложение для работы в команде. Проект Asana организован сооснователем Facebook в 2008 г. для повышения продуктивности сотрудников. Основная версия выпущена в 2011 г.

Это SaaS приложение (программное приложение как услуга, сервис) для небольших компаний и индивидуальных разработчиков, в состав функций которой включены все необходимые элементы для эффективного управления проектами.

Концепция работы с приложением Asana проста и напоминает большинство систем подобного назначения. Каждая команда может создать для себя удобное рабочее пространство (workspace). Каждое рабочее пространство может включать в себя множество проектов, а каждый проект, в

свою очередь, множество задач. Пользователи, имеющие на то полномочия, могут дополнять задачу, добавлять комментарии и теги, прикреплять к задаче файлы. Также Asana дает пользователям возможность подписаться на интересующие его задачи. В случае изменения или закрытия такой задачи все подписчики моментально получают соответствующее уведомление.

Он других систем управления, Asana немного отличается понятием post-email, то есть в ней есть секция сообщений между пользователями, работающих аналогично по отношению к внешним email системам. При этом, через Входящие сообщения, пользователи Asana могут менять параметры задач, создавать комментарии, менять даты выполнения и так далее.

У Asana также есть публичное API для разработчиков, которое позволяет интегрироваться с такими распространенными инструментами, как Dropbox, Evernote, Google Диск, Harvest, Instagantt, Jira, Zendesk, DigiSpoke и многими другими.

Подобных проектов существует множество, однако самым «крупным» игроком на рынке управления проектами является продукция Atlasian JIRA.

17.5. Комплексная система управления проектами Jira Software

Jira — коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами.

Это продукт компании Atlassian, которая сейчас является скорее интегратором, приобретя Trello, создав Bitbucket, JIRA Software, JIRA Confluence. Компания Atlassian уже 20 лет на рынке и ее влияние только увеличивается.

Для управления проектами JIRA предоставляет сразу три продукта:

- 1) JIRA Software – система управления проектами.
- 2) JIRA Confluence – система управления документацией.
- 3) Bitbucket – система управления и контроля версиями.

Система позволяет работать с несколькими проектами. Для каждого из проектов создаёт и ведёт схемы безопасности и схемы оповещения.

Процесс универсализации ускорился после запуска Atlassian Marketplace в 2012 году, который позволил сторонним разработчикам предлагать плагины для Jira типа BigPicture, Portfolio for Jira, Structure и Tempo Planner. На сегодняшний день это крупнейшая система управления проектами, имеющая бесплатную версию и гибкую систему тарифных планов.

В основе JIRA лежит понятие расширяемой карточки задачи.

Задача содержит название проекта, тему, тип, приоритет, компоненты и содержание. Задача может быть расширена дополнительными полями и даже приложениями.

Задача может редактироваться и менять статус. Переходы между состояниями задачи регулируется специально настраиваемым потоком операций. Все изменения пишутся в журнал событий.

Jira имеет большое количество возможностей конфигурации: для каждого приложения может быть определён отдельный тип задачи с

собственным workflow, набором статусов, одним или несколькими видами представления.

Система поддерживает два представления проекта:

- 1) Доска Kanban.
- 2) Списки задач и расписание Agile.

Благодаря универсальному подходу можно приспособить Jira для многих непрофильных задач, например, управления требованиями, управления рисками, вплоть до реализации небольшой системы бронирования, автоматизации процессов и так далее.

Для интеграции с внешними системами поддерживает интерфейсы SOAP, XML-RPC и REST. Поставляется со средствами интеграции с такими системами управления версиями, как Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial и Perforce. Существуют дополнения, позволяющие встроить Jira в интегрированные среды разработки, в том числе Eclipse и IntelliJ IDEA.

JIRA, Trello, Asana, Wrike уже плотно внедрились в современный мир управления проектами, как подсистемы контроля задач проекта. Подобные системы применяются повсеместно и работают в синхронизации с системами направленными на управление проектами, помогая командам создавать прекрасные вещи.

17.6. Планирование, управление проектами с применением инструментальных средств Microsoft Project, OpenProj, Redmine

Системы типа Microsoft Project, OpenProj, Redmine и их аналогов предназначены для управления именно проектами. Обычно они работают под любыми операционными системами, имеют веб-интерфейсы и переведены на все распространенные языки.

В общем случае основой процесса управления проектами здесь являются следующие возможности:

- 1) Диаграммы Ганта.
- 2) PERT-диаграммы.
- 3) Сетевой график проекта.
- 4) Система управления ресурсами проекта.
- 5) Система управления задачами.
- 6) Система отчетов по проекту.
- 7) Экспорт и импорт с различными форматами.

В таких системах работа начинается с создания проекта и установки его параметров. Также определяется вид задач, вид ресурсов проекта, определяются задачи и их тип (сводные задачи, подзадачи и вехи). Далее определяется последовательность выполнения и устанавливаются взаимозависимость и ограничения свойств задач, оценивается продолжительность реализации проекта и рассчитывается стоимость реализации.

На следующем этапе устанавливаются связи, которые можно увидеть и отредактировать. Например, в случае задержки или перекрытия (накладок)

можно указать время задержки в днях или процентах. А в случае необходимости – и разделить назначенную задачу на части. Кроме этого, можно установить ограничение (например, дату контрольного срока), на которое, в случае нарушения ограничения, укажет специальный индикатор.

В общем случае работа в системе практически все время проводится при помощи диаграммы Ганта и сопутствующих инструментов.