

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Фурсов Владимир Иванович

Должность: И.о. директора Пятигорского института (филиал) Северо-Кавказского

федерального университета

Дата подписания: 08.06.2026 13:50:44

Уникальный программный ключ:

1c378726a41fd0143ae5bccc8ba81860b06daa02

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ ОП.08 УПРАВЛЕНИЕ ИТ-ПРОЕКТАМИ

Специальности СПО

09.02.11 Разработка и управление программным обеспечением

Квалификация программист

Пятигорск 2026

Методические указания для практических занятий по дисциплине «Управление ИТ проектами» составлены в соответствии с ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.11 Разработка и управление программным обеспечением

Практическая работа № 1

Тема : Определение проекта, его ключевые характеристики: уникальность, временные ограничения, ресурсы, цели.

Проект – это временное предприятие, предназначенное для создания уникальных продуктов, услуг или результатов.

ИТ-проект – это проект в сфере создания, внедрения или применения информационных технологий.

Все проекты обладают определенным набором **свойств**.

1. Временная ограниченность проекта.

У любого проекта есть четкое начало и четкое завершение.

Завершение наступает, когда выполняется одно из условий:

- достигнуты цели проекта;
- осознано, что цели проекта не будут или не могут быть достигнуты;
- исчезла необходимость в проекте.

Временная ограниченность проекта не относится к создаваемому в ходе проекта результату. Большинство проектов предпринимается для достижения устойчивого, длительного результата.

2. Уникальность результатов.

В результате проекта получаются уникальные продукты, услуги или результаты. Ими могут быть.

— *Продукт или изделие*, которое можно измерить и которое может быть как конечным звеном производственной цепи, так и её элементом (например, информационная система в целом, или библиотека подпрограмм, которая будет использована при разработке информационных систем).

— *Способность* предоставить некоторые услуги (оптоволоконная сеть, которая будет использоваться для предоставления доступа клиентов в интернет).

— *Последствия или документы*. Например, данные, которые можно использовать для обнаружения тенденций или перспектив новой технологии.

3. Последовательная разработка.

Последовательная разработка означает развитие проекта по этапам и протекание по шагам.

Например, для проекта создания информационной системы (ИС) некоторого предприятия можно выделить следующие этапы.

- Обследование предприятия и построение его функциональной модели.
- Составление технического задания.
- Проектирование ИС.
- Реализация ИС средствами некоторой системы программирования.
- Тестирование работоспособности ИС и устранение обнаруженных ошибок.
- Ввод ИС в эксплуатацию и сдача заказчику.

4. Выполняются людьми.

5. Ограничены доступностью ресурсов.

В качестве ресурсов могут выступать люди, материальные и технические средства.

6. Планируются, исполняются и управляются.

Примерами IT-проектов являются:

- Разработка нового программного продукта;
- Разработка новой IT-услуги;
- Разработка, внедрение, приобретение или новая или усовершенствованной информационной системы;
- Техническое переоснащение корпоративной сети.

Проекты обычно **инициируются** в результате одного или нескольких стратегических соображений:

- *требования рынка* (компания инициирует проект создания нового завода по производству ноутбуков в ответ на постоянно повышающийся спрос на них);
- *нужды организации* (продавец ПО инициирует проект разработки нового курса обучения для продвижения нового ПО и увеличения своих доходов);
- *требования заказчика* (софтверная компания инициирует проект разработки новой информационной системы для автоматизации предприятия заказчика);
- *технологический прогресс* (разработчик программного обеспечения инициирует проект разработки нового поколения видеоигр после появления новых игровых приставок от производителей электроники);
- *требования законодательства* (производитель мониторов инициирует проект разработки мониторов, соответствующих утверждённому эргономическим стандартам).

Понятие управления проектами

Управление проектами — это приложение знаний, навыков, инструментов и методов к операциям проекта для удовлетворения предъявляемых к проекту требований.

Управление проектами выполняется при помощи совокупности процессов управления проектами:

- инициации,
- планирования,
- исполнения,
- мониторинга и управления,
- завершения.

Менеджер проекта — это лицо, ответственное за достижение целей проекта.

В управление проектом входят:

- определение требований;
- установка четких и достижимых целей;
- уравнивание противоречащих требований по качеству, содержанию, времени и стоимости;

— коррекция характеристик, планов и подхода в соответствии с мнением и ожиданиями различных участников проекта. Менеджеры проектов часто говорят о "тройном ограничении"

— *содержанию проекта, времени и стоимости*, — которое приходится учитывать при согласовании разнообразных требований проекта. Качество исполнения проекта зависит от уравнивания этих трех факторов.

Участники проекта – это лица или организации, активно участвующие в проекте, либо на чьи интересы могут повлиять результаты исполнения или завершения проекта. Участники могут влиять на цели и результаты проекта. Команда управления проектом должна выявить участников проекта, определить их требования и ожидания и управлять их влиянием, чтобы обеспечить успешное завершение проекта.

Иногда выявить участника проекта довольно сложно. Например, программист, чей профессиональный рост на

предприятии зависит от результата проекта разработки нового программного продукта, тоже является участником проекта. Незнание ключевых участников проекта может привести к большим сложностям при исполнении проекта..

Участники могут оказывать положительное или отрицательное влияние на проект. Положительно влияющие участники – это обычно те, кому выгодно успешное завершение проекта, тогда как отрицательно влияющим участникам успешное завершение проекта представляется нежелательным.

К **ключевым участникам** любого проекта относятся:

- *Менеджер проекта.* Лицо, ответственное за управление проектом.
- *Заказчик/пользователь.* Лицо или организация, которые будут использовать продукт проекта.
- *Исполняющая организация.* Предприятие, чьи сотрудники непосредственно участвуют в исполнении проекта.
- *Члены команды проекта.* Группа, которая выполняет работы по проекту.
- *Команда управления проектом.* Члены команды проекта, непосредственно занятые в управлении его операциями.
- *Спонсор.* Лицо или группа лиц, предоставляющая финансовые или материальные ресурсы для проекта.
- *Источники влияния.* Лица или группы, которые напрямую не связаны с получением или использованием продукта проекта, но которые, в связи с их положением в организации-заказчике или исполняющей организации, могут положительно или отрицательно повлиять на ход выполнения проекта.

Менеджеры проекта должны управлять ожиданиями участников проекта, что может быть достаточно сложно, так как у участников проекта могут быть разные или противоположные цели. Например: руководитель отдела, который потребовал установить новую информационную систему управления, может быть заинтересован в ее низкой стоимости, специалист по созданию системы может сделать акцент на техническом совершенстве системы, а подрядчик, получивший заказ на программирование, может быть заинтересован в получении максимальной прибыли.

Области знаний в управлении проектами

Многие знания, инструменты и методы, используемые в управлении проектами, применяются исключительно в этой области. К их числу относятся:

- иерархические структуры работ,
- анализ критического пути,
- управление освоенным объемом.

Однако, для эффективного управления IT-проектами необходимо, чтобы команда управления проектами понимала и использовала знания и навыки как минимум **пяти областей**.

1. Свод знаний по управлению проектами.
2. Знания, стандарты и нормативные

акты, относящиеся к информационным технологиям.

3. Понимание окружения проекта.
4. Знания и навыки в области общего менеджмента.
5. Навыки межличностных отношений.

В **своде знаний по управлению проектами** описаны знания, уникальные для управления проектами, а также общие с другими дисциплинами управления. Американский институт управления проектами PMI разработал стандарт управления проектами, описанный в Руководстве PMBOK. В него входят определение жизненного цикла проекта, описание пяти групп процессов управления проектами и описание девяти областей знаний, необходимых при управлении проектами.

Каждая область применения технологии управления проектами имеет свои самостоятельные стандарты, регламентирующие требования качества в том или ином виде деятельности.

Для сферы информационных технологий существует ряд **стандартов**, регламентирующих:

- этапы разработки программного продукта;
- требования к качеству программного продукта;
- другие требования.

Эти стандарты должны соблюдаться наряду со стандартами в области управления проектами.

Практически все проекты планируются и выполняются в том или ином социальном, экономическом и природном **окружении** и сопровождаются запланированными и незапланированными, благоприятными и неблагоприятными воздействиями. Выделяются виды окружения, которое должно учитываться при управлении проектом.

— *Социально-культурное окружение.* Команда должна понимать, как проект воздействует на людей и как люди воздействуют на проект.

— *Международно-политическое окружение.* Международные, национальные, региональные и местные законы и обычаями, политическая ситуация, временные пояса, национальные и региональные праздники и т.п., которые могут оказывать влияние на проект.

— *Окружающая среда.* Местная экология и физическая география, которая может оказать влияние на проект или быть затронута проектом.

Общий менеджмент охватывает планирование, организацию, обеспечение персоналом, исполнение и управление операционной деятельностью работающего предприятия. В него входят вспомогательные дисциплины, такие как:

- управление финансами и бухгалтерский учет;
- закупки и снабжение;
- продажи и маркетинг;
- контракты и торговое право;
- производство и дистрибуция;
- логистика и логистическая цепочка;
- стратегическое, тактическое и оперативное планирование;
- организационные структуры, организационное поведение, управление персоналом, вознаграждением, признанием и карьерным ростом;

- здравоохранение и техника безопасности;
- информационные технологии.

В управление межличностными отношениями входит:

- *Эффективные коммуникации.* Обмен информацией.
- *Влияние на организацию.* Способность “делать дело”.
- *Лидерство.* Разработка определенных видения и стратегии, и мотивация людей для воплощения этого видения и стратегии.
- *Мотивация.* Стимуляция людей с целью достижениями высокого уровня исполнения и преодоления препятствий.
- *Переговоры и улаживание конфликтов.* Совещания с другими людьми для того, чтобы договориться с ними или прийти к соглашению.
- *Решение проблем.* Определение проблем, обнаружение и анализ альтернатив и принятие решений.

Жизненный цикл проекта

Жизненный цикл проекта – это совокупность фаз проекта, которые связывают его начало и окончание. Переход из одной фазы в другую в пределах жизненного цикла проекта обычно подразумевает некую форму технической передачи или сдачи результатов, и часто именно это указывает на переход от одной фазы к другой. Результаты одной фазы обычно проверяются на предмет завершенности и точности и проходят процедуру одобрения, прежде чем начнутся работы следующей фазы.

Жизненный цикл проекта обычно определяет:

1. Какие технические работы должны быть проведены в каждой фазе (например, в какой фазе должно быть проведено проектирование?).
2. В какой момент каждой фазы должны быть получены конкретные результаты и как проходит проверка и подтверждение каждого результата.
3. Кто участвует в каждой фазе (например, работы по созданию корпоративной сети требуют, чтобы те, кто их выполняет, участвовали в определении требований и проектировании).
4. Как контролировать и подтверждать каждую фазу.

Жизненные циклы различных проектов имеют ряд **общих свойств**:

- Фазы обычно идут последовательно и ограничиваются передачей либо технической информации, либо некоторого технического элемента.
- Уровень затрат и численность задействованного персонала невелики в начале, увеличиваются по ходу выполнения проекта и быстро падают на завершающем этапе проекта. График их зависимости от времени показан на рис. 1.1.
- Уровень неуверенности и риск недостижения целей наиболее велики в начале проекта. Уверенность в завершении проекта, как правило, увеличивается по ходу выполнения проекта.

—Способность участников проекта повлиять на конечные характеристики продукта проекта и окончательную стоимость проекта максимальны в начале проекта и уменьшаются по ходу выполнения проекта (рис.1.2). Главная причина этого состоит в том, что стоимость внесения изменений в проект и исправления ошибок, в общем случае, возрастает по ходу выполнения проекта.

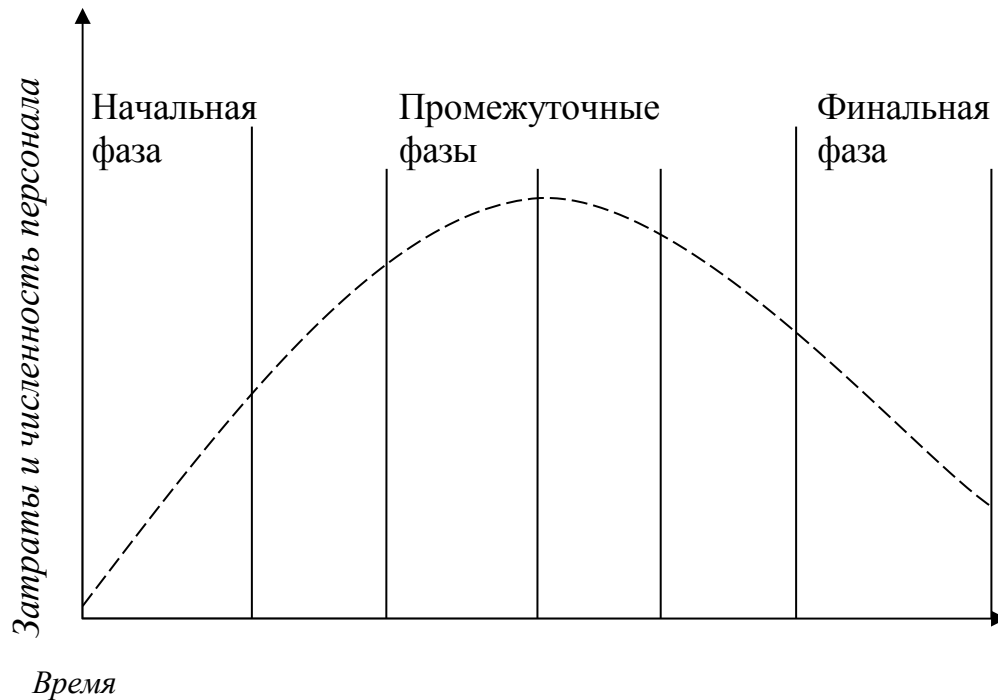


Рис.1.1. Изменение затрат и численности персонала проекта во времени



Рис.1.2. Влияние участников в течение проекта

Фаза проекта характеризуется завершением и одобрением одного или нескольких её результатов. Результат фазы – это измеримый, проверяемый продукт работы, например спецификация, отчет, детальный план или опытный образец. Фазы могут разбиваться на подфазы из соображений размера, сложности, уровня риска и ограничений на финансирование.

Фаза проекта обычно завершается изучением проделанной работы и её результатов, чтобы определить, насколько они приемлемы, и решить, необходимы ли еще дополнительные работы или фазу можно считать закрытой. Для обеспечения эффективного контроля в каждой фазе имеется своя группа процессов инициации, на выходе которой получается специфичный для данной фазы выход. Этот выход определяет, что для данной фазы полагается и что от нее ожидается, как это показано на рис. 1.3.

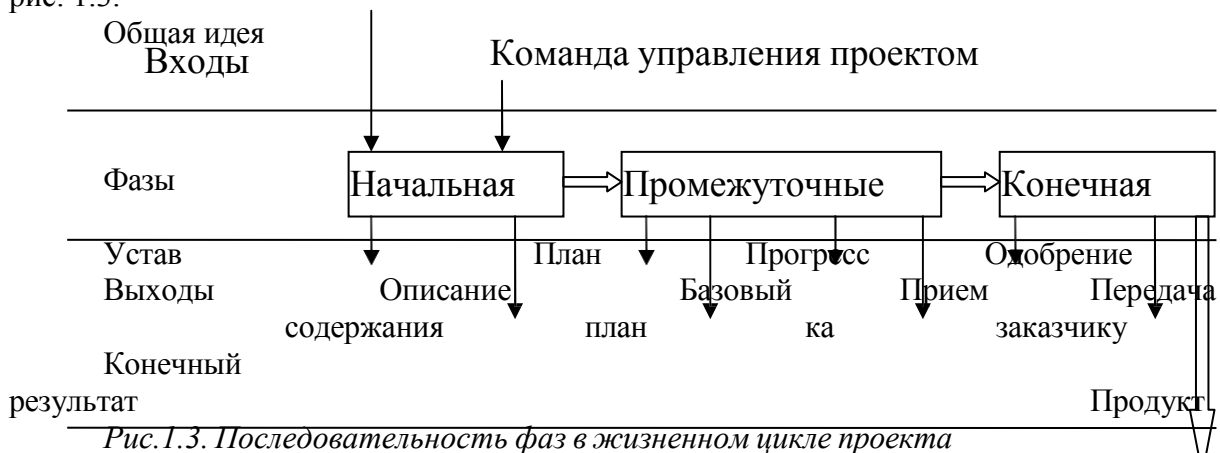


Рис.1.3. Последовательность фаз в жизненном цикле проекта

Следует различать жизненный цикл проекта и жизненный цикл продукта. Жизненный цикл продукта намного шире и включает в себя жизненный цикл проекта. Например, проект, предпринимаемый с целью выпуска на рынок нового персонального компьютера, является лишь одним из аспектов жизненного цикла продукта. На рис. 1.4 показан жизненный цикл продукта, начиная с бизнес-плана, идеи, до продукта, текущих операций и реализации продукта.

Влияние организации на проект

Проекты обычно являются частью организации, которая сама по себе больше, чем проект. Организация всегда оказывает влияние на проект.

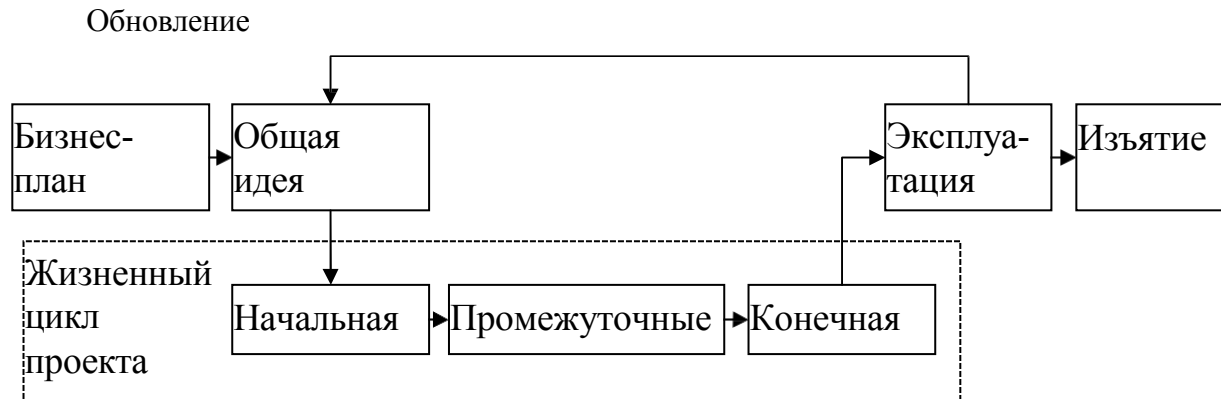


Рис.1.4. Жизненный цикл продукта

Проектные организации можно разделить на два вида:

— Организации, получающие прибыль за счет выполнения проектов для других по контракту: консалтинговые фирмы, системные интеграторы, разработчики ПО.

— Организации, в которых внедрено управление через проекты. Эти организации, как правило, имеют и используют системы, облегчающие управление проектами во всех сферах деятельности (финансы, производство и т.д.).

Классическая **функциональная организация**, показанная на рис.1.5 (участники проекта выделены серым цветом), является иерархической структурой, в которой каждый служащий имеет одного четко выделяемого руководителя. Персонал группируется по специальностям, например, производство, маркетинг, финансы, отчетность, конструирование. Когда новый продукт разрабатывается в чисто функциональной организации, фаза проектирования включает только персонал инженерного отдела. Когда возникает вопрос, касающийся производства, служащие подают запрос вверх по иерархической лестнице организации руководителю их отдела, который консультируется с руководителем производственного отдела. Руководитель инженерного отдела затем передает ответ по иерархической лестнице вниз функциональному руководителю.

На противоположном конце спектра находится проектная организация, показанная на рис. 1.6 (участники проекта выделены серым цветом). В проектной организации члены команд часто собраны в одном месте. Большая часть ресурсов организации задействована в работах проектов, а менеджеры проектов в значительной степени независимы и обладают большими полномочиями. Проектные организации часто имеют подразделения, называемые отделами, но эти подразделения подотчетны непосредственно менеджеру проекта.

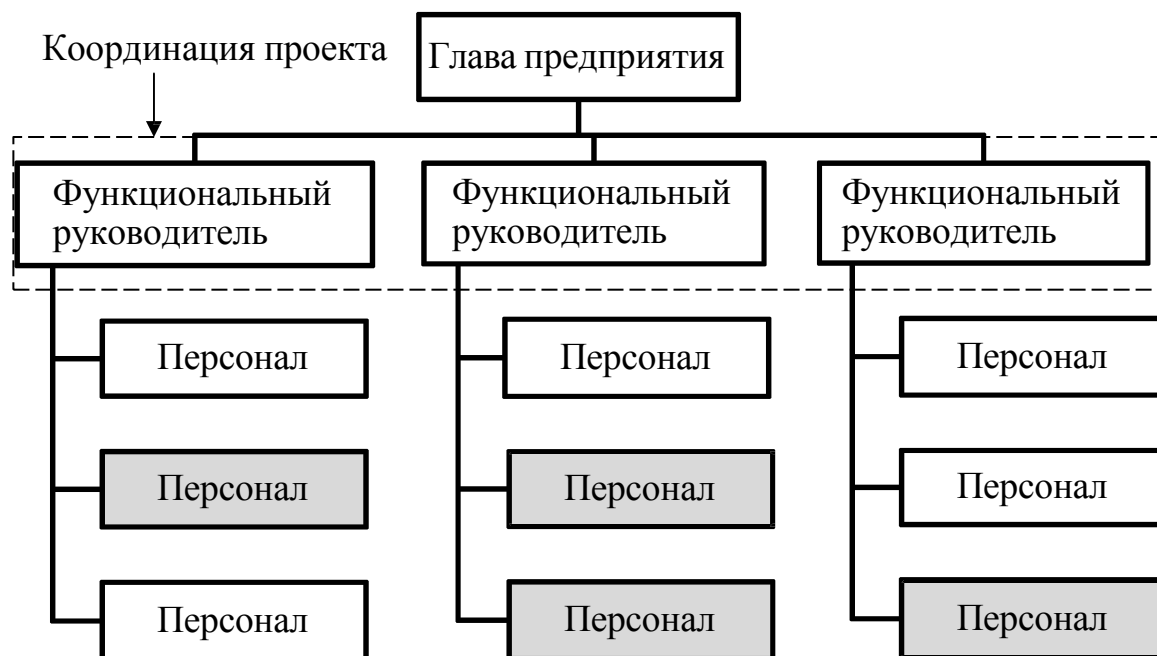


Рис.1.5. Функциональная организация

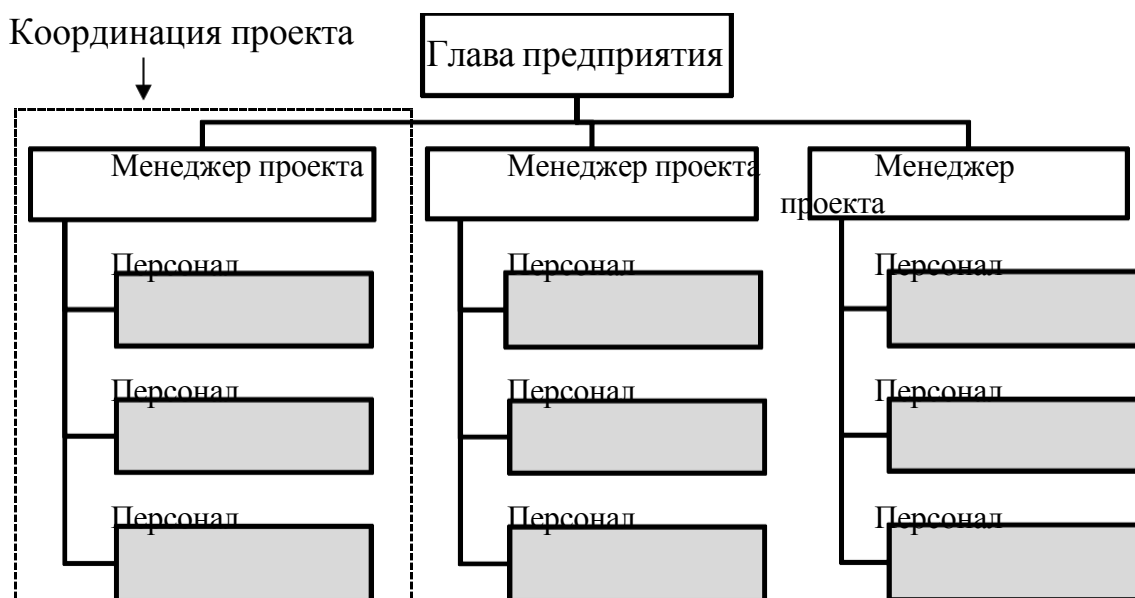


Рис.2.6. Проектная организация

Существуют смешенные типы организационных структур, сочетающие элементы функциональной и проектной организации: слабая, сбалансированная и жёсткая матричные организации, изображённые на рис.1.7 – 1.9. **Слабая матрица** сохраняет многие характеристики функциональной организации, и функции менеджера проекта в ней скорее соответствуют функциям координатора или диспетчера проектов, а не менеджера. **Сильная матрица** обладает многими характеристиками проектных организаций, в ней могут быть штатные менеджеры проектов с широкими полномочиями и входящий в штат управленческий персонал проектов. В **сбалансированной матричной** организации осознают необходимость в менеджере проекта, однако в ней он не обладает всеми полномочиями по управлению и финансированию проектом.

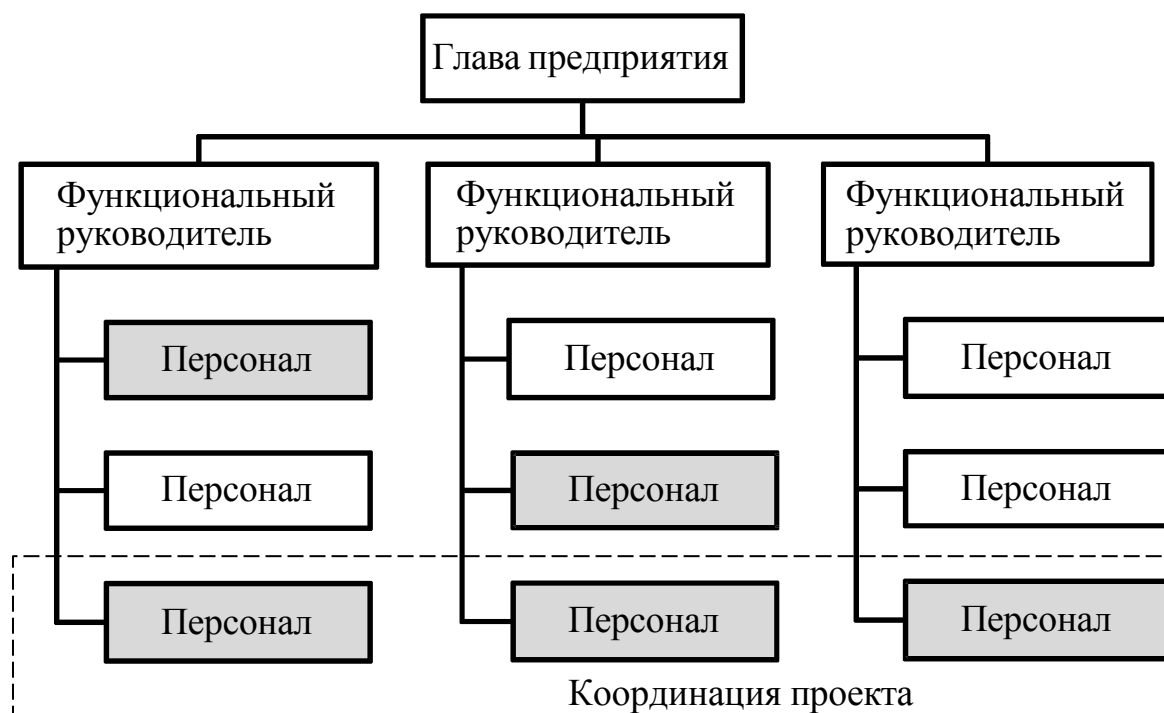


Рис.1.7. Слабая матричная организация

Характеристики различных видов организационных структур приведено в табл.1.1.

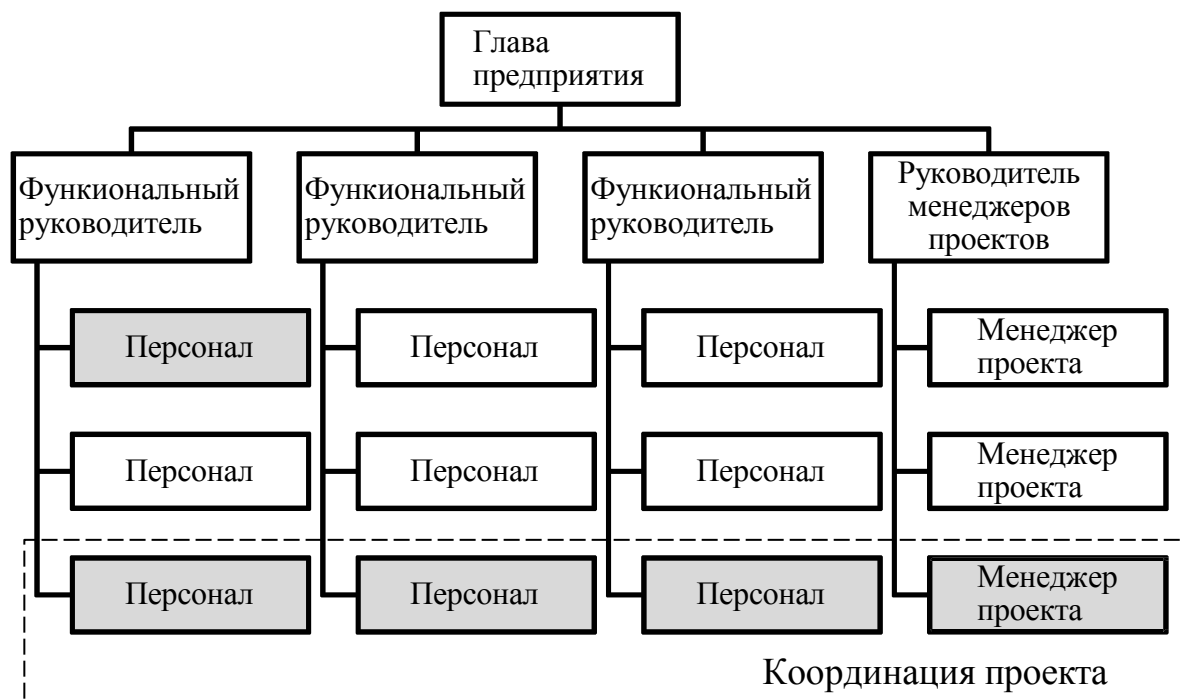


Рис.1.8. Сильная матричная организация

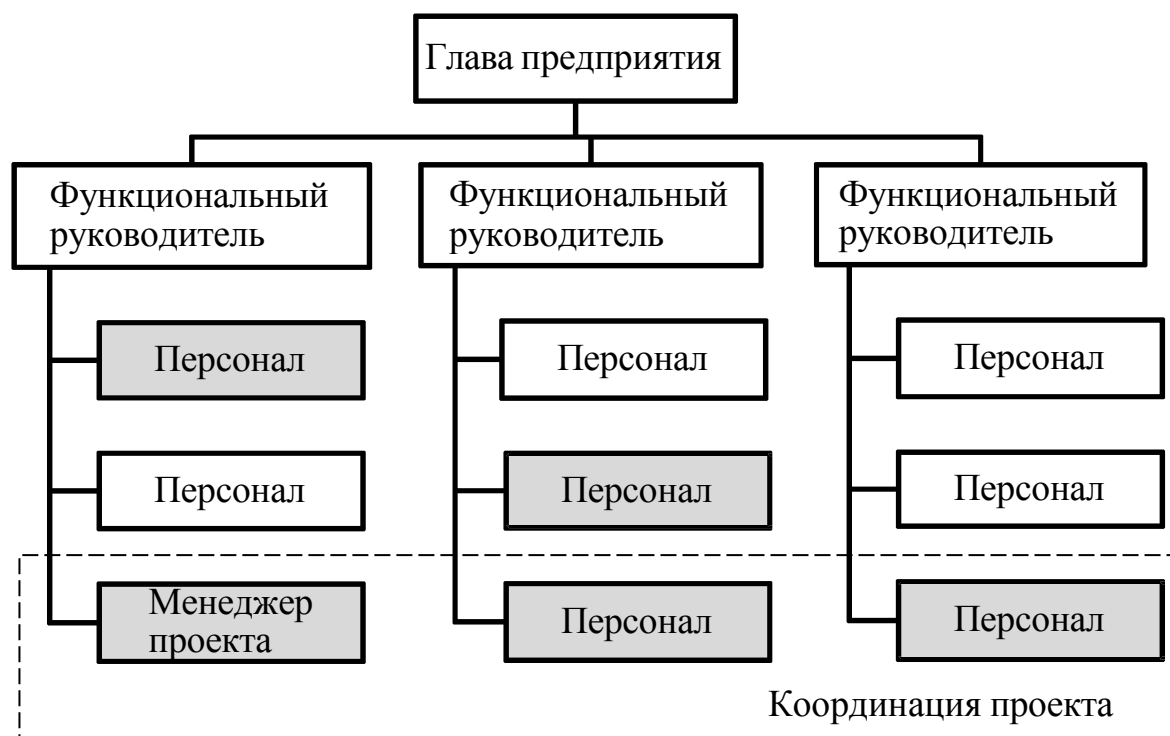


Рис.1.9. Сбалансированная матричная организация

Таблица 1.1.

Характеристики проекта	Виды организационных структур				
	Функциональная	Матричные			Проектная
		Слабая	Сбалансированная	Жесткая	
Полномочия менеджера проекта	Незначительные или нет	Ограничено	Средний уровень	Высокий уровень	Полный контроль
Наличие ресурсов	Незначительно или нет	Ограничено	Средний уровень	Высокий уровень	Полный контроль
Контроль бюджета проекта	Функциональный.	Функциональный.	Смешанный.	Менеджер проекта	Менеджер проекта
Роль менеджера проекта	Частичная занятость в проекте	Частичная занятость в проекте	Полная занятость в проекте	Полная занятость в проекте	Полная занятость в проекте
Административный персонал проекта	Частичная занятость в проекте	Частичная занятость в проекте	Частичная занятость в проекте	Полная занятость в проекте	Полная занятость в проекте

Контрольные вопросы

1. Что такое проект и IT-проект?
2. Какие свойства присущи проектам?
3. В чем заключается временный характер проекта?
4. Какие виды результатов могут создаваться проектами?
5. Назовите примеры IT-проектов?
6. В чем заключается последовательная разработка проекта?
7. По каким причинам инициируются проекты?
8. Что такое управление проектами?
9. Кто входит в состав участников проекта?
10. Каковы ключевые участники проекта?
11. Какие области знаний используются для эффективного управления IT-проектами?
12. Что регламентирует свод знаний по управлению проектами?
13. Какие виды окружения могут воздействовать на проект?
14. Какие разделы общего менеджмента используются при управлении проектами?
15. Что входит в управление межличностными отношениями?

16. Что характеризует жизненный цикл проекта?
17. Какие общие свойства имеют жизненные циклы разных проектов?
18. Что такое фаза проекта?
19. Каково соотношение между жизненным циклом проекта и жизненным циклом продукта?
20. Какова структура функциональной организации?
21. Какова структура проектной организации?
22. Что такое слабая матричная организация?
23. Что такое сильная матричная организация?
24. Что такое сбалансированная матричная организация?
25. Каковы сравнительные характеристики различных матричных структур?

Практическое занятие № 2

Тема: Этапы жизненного цикла проекта: инициация, планирование, выполнение, мониторинг, завершение.

Процесс – это ряд взаимосвязанных действий и операций, выполняемых для достижения заранее определенных продуктов, результатов или услуг. Процессы управления проектом выполняются командой проекта и обычно бывают двух типов.

— *Процессы управления проектом*, общие для большинства проектов, связанные между собой тем, что они нацелены на выполнение общей задачи. Например, инициация, планирование, управление и т.д.

— *Процессы, ориентированные на продукт*. Определяют и создают продукт проекта. Например, постановка задачи, программирование, тестирование и т.п.

Эти типы процессов накладываются друг на друга.

Стандартом РМВОК выделяется 5 групп процессов:

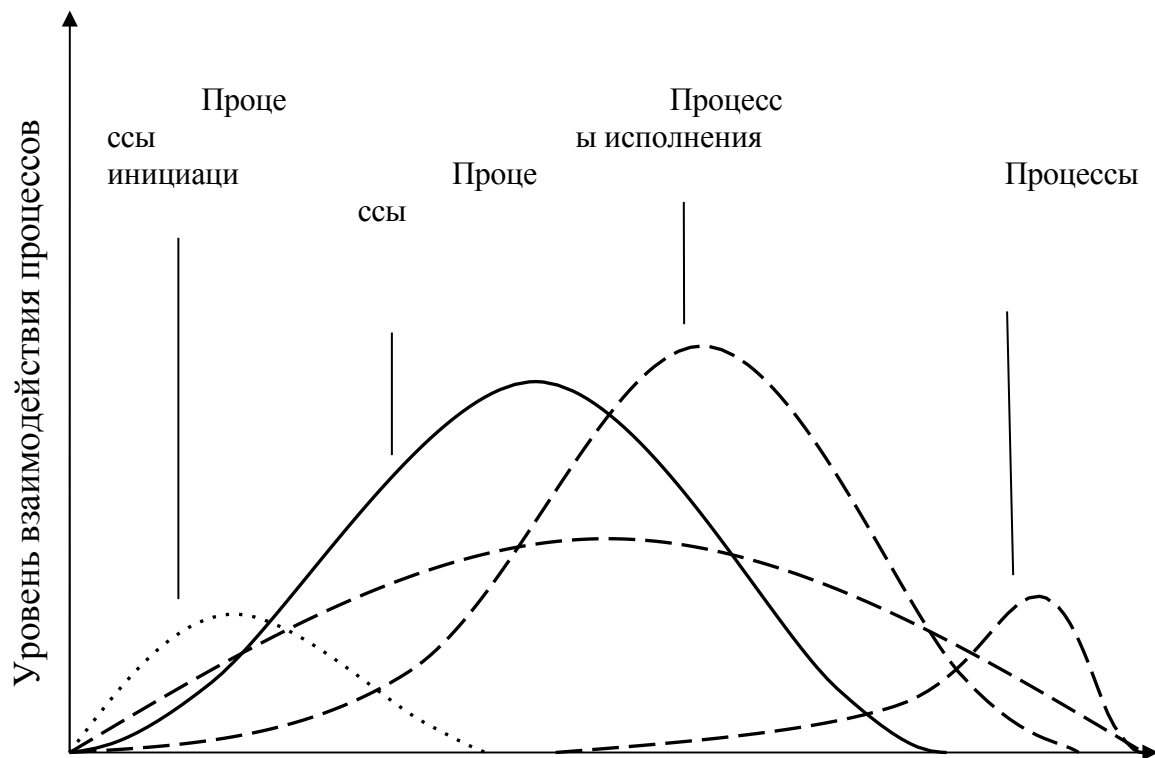
1. **Группа процессов инициации.** Определяет и авторизует проект или фазу проекта.
2. **Группа процессов планирования.** Определяет и уточняет цели и планирует действия, необходимые для достижения целей, ради которых был предпринят проект.
3. **Группа процессов исполнения.** Объединяет человеческие и другие ресурсы для выполнения плана управления проектом
4. **Группа процессов мониторинга и управления.** Регулярно оценивает прогресс проекта и осуществляет мониторинг, чтобы обнаружить отклонения от плана управления проектом, и, в случае необходимости, провести корректирующие действия для достижения целей проекта.
5. **Группа завершающих процессов.** Формализует приемку продукта проекта и подводит проект к правильному завершению.

Взаимосвязь между этими группами процессов изображена на рис.2.1.

Группы процессов управления проектом связаны целями, которые перед ними поставлены. Выход одного процесса обычно является входом для другого или является конечным результатом проекта. Группы процессов редко являются дискретными или однократными событиями, они накладываются друг на друга в течение жизненного цикла проекта. На рис. 2.2 изображено, как взаимодействуют группы процессов, а также показан уровень наложения в различные периоды реализации проекта.



Рис.2.1. Взаимосвязь групп процессов проекта



Группа процессов инициации

В ходе процесса инициации уточняются первоначальное описание содержания проекта и ресурсы, которые организация планирует вложить. На этом этапе выбирается менеджер проекта, и документируются исходные допущения и ограничения. Эта информация заносится в Устав проекта и, если он одобряется, проект официально авторизуется.

К процессам инициации относятся.

1. Разработка Устава проекта. Это процесс, необходимый для формулирования практических нужд и документального оформления нового продукта, услуги или иного результата, создание которого будет являться конечной целью проекта и который должен удовлетворять всем сформулированным требованиям. С помощью Устава проект привязывается к текущей работе организации, а также осуществляется авторизация проекта.

2. Разработка предварительного описания содержания проекта. Этот процесс направляет и документирует требования к проекту и конечному результату, требования к продукту, границы проекта, методы приемки и общее управление содержанием.

Группа процессов планирования

Цель группы процессов планирования – собрать информацию из нескольких источников, различных по уровню полноты и доверия. Процессы планирования разрабатывают план управления проектом. Эти процессы также обнаруживают, определяют и дорабатывают содержание и стоимость проекта и составляют расписание для операций проекта. В ходе проекта неоднократно возникает необходимость в дополнительном анализе, а значит и в возврате к уже утвержденным процессам. Значительные изменения, происходящие во время жизненного цикла проекта, приводят к необходимости пересмотреть один или несколько процессов планирования и, возможно, некоторые из процессов инициации.

Взаимодействие процессов планирования между собой изображено на рис.2.3.

К процессам планирования относятся.

1. Разработка плана управления проектом. Это процесс, необходимый для определения, подготовки, координации и интеграции всех вспомогательных планов в план управления проектом. План управления проектом становится первичным источником информации по планированию, исполнению, мониторингу и управлению, а также закрытию проекта.

2. Планирование содержания. Создание плана управления содержанием проекта, который описывает, как будет определяться, проверяться и управляться содержанием проекта

и как будет создана и определена иерархическая структура работ.



Рис.2.3. Взаимодействие процессов планирования

3. **Определение содержания.** Разработка подробного описания содержания проекта, на основании которого будут впоследствии приниматься решения по проекту.

4. **Создание иерархической структуры работ (ИСР).** Разделение конечных результатов проекта и работ проекта на меньшие элементы, которыми легче управлять. Например, Постановка задачи

Технические требования	Интерфейс
Функциональные	требования
Программирование	

Разработка модели данных Программирование
интерфейса Программирование функциональных модулей
Отладка модулей
Отладки взаимодействия модулей
Тестирование
и т.д.

5. **Определение состава операций.** Идентификация конкретных операций, которые следует выполнить для получения тех или иных результатов проекта.

6. **Определение взаимосвязей операций.** Определение и документирование взаимосвязей между операциями.

7. **Оценка ресурсов операций.** Оценка типа и количества ресурсов, необходимых для выполнения каждой плановой операции.

8. **Оценка длительности операций.** Оценка количества рабочих периодов, которые потребуются для завершения каждой плановой операции.

9. **Разработка расписания.** Анализ последовательности, длительности операций, требований к ресурсам и ограничений на сроки с целью создания календарного расписания проекта.

10. **Стоимостная оценка.** Разработка приблизительных значений стоимости ресурсов, необходимых для выполнения операций проекта.

11. **Разработка бюджета расходов.** Суммирование оценок стоимости отдельных операций или пакетов работ для оценки базового плана по стоимости.

12. **Планирование качества.** Определение стандартов качества, которые соответствуют проекту, и средств достижения этих стандартов.

13. **Планирование человеческих ресурсов.** Определение и документирования ролей в проекте, ответственности и отчетности, а также создания плана управления обеспечением проекта персоналом.

14. **Планирование коммуникаций.** Определение потребностей участников проекта в информации и коммуникациях.

15. **Планирование управления рисками.** Определения подходов к планированию и выполнению операций по управлению рисками проекта.

16. **Идентификация рисков.** Определение, какие именно риски могут повлиять на проект и документирование их характеристик.

17. **Качественный анализ рисков.** Определение приоритетов рисков с целью дальнейшего анализа действий, оценка вероятности и воздействия рисков.

18. **Количественный анализ рисков.** Количественный анализ воздействия определенного риска на общие цели проекта.

19. **Планирование реагирования на риски.** Разработка вариантов и операций для снижения угроз рисков целям проекта.

20. **Планирование покупок.** Определение, что, как и когда следует приобрести.

21. **Планирование контрактов.** Документирование требований к продуктам, услугам и результатам, а также поиск потенциальных продавцов.

Процессы исполнения

Данная группа процессов включает в себя координацию людей и ресурсов, и исполнение операций проекта в соответствии с планом управления проектом. В ходе исполнения идет работа с содержанием проекта, определенным в описании содержания

проекта, и в него вносятся одобренные изменения.

Обычно при исполнении имеют место отклонения, приводящие к корректировке планов. Эти отклонения могут затрагивать длительность операций, наличие и эффективность ресурсов, непредусмотренные риски и т.п. Результаты анализа отклонений могут повлечь за собой запрос на изменение. Если этот запрос будет одобрен, то это может привести к изменению плана управления проектом и утверждению нового базового

плана. Подавляющая часть бюджета проекта идёт на выполнение группы процессов исполнения.

К процессам исполнения относятся.

1. **Руководство и управление исполнением проекта.** Управление трудовыми и материальными ресурсами для выполнения работ, предусмотренных в плане управления проектом. Конечные результаты представляются как выходы выполненных процессов, указанных в плане управления проектом. По мере выполнения проекта собирается информация о завершённости конечных результатов и о том, какие именно работы завершены. Эта информация становится входом для процесса отчетности по исполнению.

2. **Процесс обеспечения качества.** Применение плановых систематических операций по проверке качества, (например аудит или независимая экспертиза), чтобы удостовериться, что в проекте используются все необходимые процессы для выполнения заданных требований.

3. **Набор команды проекта.** Получение человеческих ресурсов, нужных для выполнения проекта.

4. **Развитие команды проекта.** Повышение компетенции и взаимодействия членов команды для улучшения исполнения проекта.

5. **Распространение информации.** Обеспечение своевременного доступа участников проекта к нужной им информации.

6. **Запрос информации у продавцов.** Получение информации, расценок или предложений.

7. **Выбор продавцов.** Изучение предложений, выбор из потенциальных продавцов и заключение письменного контракта с продавцом.

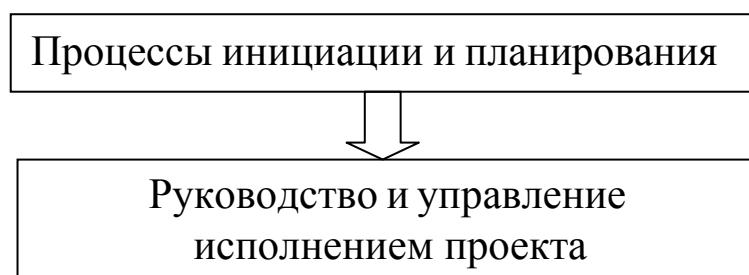
На рис.2.4 изображено взаимодействие процессов исполнения.

Процессы мониторинга и управления

Группа процессов мониторинга и управления состоит из процессов, предназначенных для правильного исполнения проекта, так чтобы возможные проблемы были обнаружены вовремя и по ним могли быть предприняты корректирующие действия. Постоянный мониторинг дает команде проекта представление о состоянии проекта и выделяет участки, которым нужно дополнительное внимание.

В группу процессов мониторинга и управления входят следующие процессы.

1. **Мониторинг и управление работами проекта.** Этот процесс включает в себя:



Процесс
обеспечения
качества

Рис.2.4. Взаимодействие процессов исполнения

- a. Сбор, измерение и распространение информации об исполнении проекта.
 - b. Анализ влияния тенденций на ход процессов проекта.
 - c. Мониторинг рисков, обеспечивающий выявление рисков на ранних стадиях, составление отчета об их состоянии, и приведение в исполнение соответствующие планов реагирования на риски.
 - d. Составление отчетов о текущем состоянии, оценку прогресса и прогнозирование. Отчеты об исполнении предоставляют информацию об исполнении проекта по таким показателям, как содержание, расписание, стоимость, ресурсы, качество и риски.
2. **Общее управление изменениями.** Управление факторами, создающими изменения, чтобы эти изменения были благотворными, отслеживание внесения изменений и управление одобренными изменениями. Этот процесс выполняется в течение всего проекта, от инициации до закрытия проекта.
3. **Подтверждение содержания.** Приемка завершенных результатов проекта.
4. **Управление содержанием.** Управление изменениями в содержании проекта.
5. **Управление расписанием.** Управление изменениями в расписании проекта.
6. **Управление стоимостью.** Управление изменениями бюджета проекта.

7. **Процесс контроля качества.** Мониторинг определенных результатов проекта с целью определения их соответствия принятым стандартам качества и выработки путей устранения причин неудовлетворительного исполнения.

8. **Управление командой проекта.** Отслеживание деятельности членов команды, обеспечение обратной связи, решение проблем и координация изменений с целью улучшения исполнения проекта.

9. **Отчетность по исполнению.** Сбор и распространение информации об исполнении. Эта информация включает в себя отчеты о текущем состоянии, оценку прогресса, а также прогнозирование.

10. **Управление участниками проекта.** Управление коммуникациями с целью удовлетворения требований участников проекта и решения вместе с ними возникающих проблем.

11. **Наблюдение и управление рисками.** Отслеживание выявленных рисков, мониторинг остаточных рисков, выявление новых рисков, выполнение планов реагирования на риски и оценка их эффективности в течение жизненного цикла проекта.

12. **Администрирование контрактов.** Управление контрактом и взаимоотношениями между продавцом и покупателем, изучение и документирование действий продавца и, в соответствующих случаях, управление контрактными отношениями с внешним покупателем проекта.

Процессы завершения

Это процессы, используемые для формального завершения всех операций проекта, передачи завершеного продукта другим лицам или закрытия остановленного проекта.

К процессам завершения относятся:

1. **Закрытие проекта.** Завершение всех операций всех групп процессов, чтобы формально закрыть проект.

2. **Закрытие контрактов.** Завершение и урегулирование каждого контракта, в том числе завершение действующих контрактов и закрытие каждого контракта, затрагивающего проект.

Управление интеграцией проекта

Интеграция – это принятие решений о том, где концентрировать ресурсы на каждую конкретную дату, предугадывание потенциальных проблем, и их решение до того, как эти проблемы станут критическими, а также хорошая координация работы проекта в целом. Интеграция подразумевает нахождение компромиссов между пересекающимися целями и альтернативами. **Цель интеграции** – это эффективное интегрирование (объединение) процессов в группах процессов управления проектами, необходимых для достижения конечных целей проекта.

Интегративные процессы управления проектами включают в себя следующие элементы:

1. **Разработка Устава проекта,** формально авторизующего проект.

2. **Разработка предварительного описания содержания проекта,** включающего в себя самое общее изложение содержания.

3. **Разработка плана управления проектом** – документирование операций, необходимых для определения, подготовки, интеграции всех вспомогательных планов в план управления проектами и их координации.

4. **Руководство и управление исполнением проекта** – выполнение

работы, предусмотренной Планом управления проектом, для выполнения требований, определенных в описании содержания проекта.

5. Мониторинг и управление работами проекта – мониторинг и управление процессами инициации, планирования, выполнения и завершения проекта для достижения целевых показателей эффективности, намеченных в Плане управления проектом.

6. Общее управление изменениями – обработка всех запросов на изменения, утверждение этих изменений и управление ими для оптимизации конечных результатов.

7. Заккрытие проекта – завершение всех операций во всех группах процессов управления проектами для формального закрытия проекта или проектной фазы.

На рис. 3.1 показана взаимосвязь интегративных процессов между собой.

Устав проекта является документом, формально авторизующим проект. Он наделяет менеджера проекта полномочиями задействовать ресурсы организации для операций проекта. Устав составляется инициатором проекта или спонсором, не входящим в организацию проекта и имеющим достаточные полномочия для финансирования проекта.

Устав должен содержать следующую информацию:

1. Требования, удовлетворяющие потребности, пожелания и ожидания заказчика, спонсора и других участников проекта.
2. Требования к продукту, который является предметом проекта.
3. Цель или обоснование проекта.
4. Информацию о назначенном менеджере проекта и уровне его полномочий.
5. Расписание контрольных событий.
6. Отношения между участниками проекта.
7. Функциональные организации и их участие.
8. Допущения, принимаемые в проекте (например, уровень инфляции).
9. Ограничения, принимаемые в проекте (например, операционная система и т.п.).
10. Реальная бизнес-ситуация, служащая обоснованием проекта с данными о прибыли на инвестиции.
11. Бюджет проекта.

Практическое занятие № 3

Тема : Преимущества и недостатки классической водопадной модели для IT-проектов.

****Водопадная модель**** (Waterfall) — это классический подход к управлению разработкой, предложенный Уинстоном Ройсом в 1970 году. Она предполагает последовательное прохождение этапов: анализ требований → проектирование → реализация → тестирование → эксплуатация.

Ниже приведен анализ преимуществ и недостатков этой модели для IT-проектов.

Преимущества (Когда это работает)

****1. Четкость структуры и дисциплина****

Проект разбит на строгие последовательные фазы. Переход на следующий этап возможен только после полного завершения предыдущего. Это дисциплинирует команду и заказчика, исключая хаос.

****2. Простота управления и контроля****

Благодаря линейности, менеджеру легко оценивать прогресс. Существуют четкие точки контроля (milestones) и артефакты (техническое задание, архитектурная документация), по которым можно измерить выполнение.

****3. Стабильность требований (при наличии)****

Если требования к продукту абсолютно понятны с самого начала, не меняются в процессе разработки и жестко зафиксированы в контракте (например, в госсекторе или аутсорсинге с фиксированной ценой), водопад позволяет точно рассчитать сроки и бюджет.

****4. Дисциплина документирования****

Модель требует исчерпывающей документации на ранних этапах. Это снижает риски «ухода знаний» вместе с сотрудником и упрощает передачу проекта на сопровождение другой команде.

****5. Предсказуемость бюджета****

При фиксированном объеме требований легко рассчитать итоговую стоимость проекта на старте.

Недостатки (Риски и ограничения)

****1. Главный риск: требования****

В современном IT требования редко бывают идеально сформулированы на старте. Водопад не предполагает возврата назад. Если ошибка в требованиях обнаруживается на этапе тестирования, её исправление стоит огромных ресурсов (иногда дешевле переписать всё с нуля).

****2. Позднее тестирование и обратная связь****

Заказчик впервые видит работающий продукт (или обнаруживает критическую ошибку) только в самом конце проекта. Если продукт не соответствует реальным ожиданиям бизнеса, у команды уже нет времени и бюджета на изменения.

****3. Низкая гибкость (Rigidity)****

Любое изменение требований («а давайте добавим кнопку», «рынок изменился, нужен другой функционал») требует пересмотра всего плана, внесения изменений в контракт и часто приводит к катастрофическому срыву сроков.

****4. Накопление технического долга****

Пока идет этап разработки, архитектурные ошибки, заложенные на этапе проектирования, не видны. К моменту тестирования может выясниться, что архитектура устарела или не выдерживает нагрузок, но переделывать её уже поздно.

****5. Долгий Time-to-Market****

Продукт выходит на рынок только после прохождения всех этапов. В условиях современной конкуренции (Agile, DevOps) это часто означает, что к моменту релиза конкурент уже занял нишу, а требования пользователей изменились.

Резюме: когда стоит выбирать Waterfall?

Вопреки популярному мнению, водопад не является «устаревшим» и «плохим». Он остается оптимальным выбором при соблюдении следующих условий:

1. ****Требования строго фиксированы**** и не будут меняться (например, модернизация бухгалтерской системы с четкими законодательными нормами).
2. ****Проект небольшой**** или средний, с простой логикой.
3. ****Высокая цена ошибки**** в архитектуре (например, разработка ПО для космической отрасли или медицинских устройств), где необходимо скрупулезное планирование до написания кода.
4. ****Контрактная модель**** с фиксированной ценой (Fixed Price), где заказчик не хочет платить за «итеративность» и требует четкой сметы до старта работ.

В остальных случаях (стартапы, продукты с неясными требованиями, сложные интеграции) современные адаптивные методологии (Agile, Scrum, Kanban) обычно демонстрируют более высокую эффективность за счет гибкости.

5 вопросов по теме преимуществ и недостатков водопадной модели, которые подойдут для обсуждения, самопроверки или собеседования:

1. Почему в водопадной модели критически важно полное документирование требований на старте, и какие последствия возникают, если требования оказываются ошибочными?

(Проверяет понимание принципа последовательности и цены поздних изменений.)

2. В каких типах IT-проектов водопадная модель остаётся предпочтительнее гибких методологий (Agile), несмотря на её жёсткость? Приведите два-три конкретных примера.

(Проверяет знание областей применения, где важна предсказуемость и фиксированность.)

3. Сравните управление рисками в водопадной модели и в Scrum. Какие риски в Waterfall закладываются «по умолчанию», а какие — минимизируются?

(Проверяет способность анализировать риски: позднее обнаружение несоответствий vs. чёткое планирование бюджета.)

4. Представьте, что вы руководите проектом разработки медицинской системы с жесткими требованиями регулятора. Заказчик настаивает на водопаде, но команда хочет использовать итеративный подход. Как бы вы аргументировали выбор модели, учитывая плюсы и минусы каждой?

(Проверяет практическое применение знаний, умение балансировать.)

5. Что такое «технический долг» в контексте водопадной модели и почему он часто накапливается незаметно, проявляясь лишь на этапе тестирования или внедрения?

(Проверяет понимание скрытых недостатков последовательной разработки и важности раннего тестирования.)

Практическое занятие № 4

Тема : Как выбрать подход к управлению проектом в зависимости от типа задачи и специфики проекта.

Выбор подхода к управлению проектом — это компромисс между ****определенностью**** (знаем, что делать) и ****гибкостью**** (готовы к изменениям). Не существует универсальной методологии; правильный выбор зависит от характеристик проекта, команды, заказчика и внешних условий.

Ниже приведена система критериев, помогающая определить, какой подход (Waterfall, Agile, гибрид или специализированный) будет наиболее эффективным.

1. Ключевые факторы выбора

А. Требования

	Если требования...		**Рекомендуемый подход**	
	----- -----			
	Полностью известны, стабильны, зафиксированы в контракте		**Waterfall**	
	(каскад)			
	Неизвестны, меняются, уточняются в процессе		**Agile** (Scrum, Kanban)	
	Частично известны, есть критический костяк		**Гибрид** (waterfall для архитектуры, agile для разработки)	

Б. Сложность и риски

	Фактор		**Waterfall подходит, если...**		**Agile подходит, если...**	
	----- ----- -----					
	Техническая сложность		Низкая или средняя, архитектура предсказуема			
	Высокая, требуется прототипирование и быстрая обратная связь					
	Интеграция с legacy-системами		Жесткие интерфейсы определены заранее			
	Требуется итеративная стыковка					
	Риск изменения требований		Минимальный		Высокий, бизнес-среда нестабильна	

В. Команда и организация

- ****Waterfall**** требует четкого разделения ролей, подробной документации, возможности долгосрочного планирования. Хорошо работает в крупных компаниях с матричной структурой.

- ****Agile**** требует автономной, кросс-функциональной команды, высокой дисциплины в коммуникациях, готовности менять планы. Плохо сочетается с жесткой бюрократией.

Г. Время и бюджет

- ****Waterfall**** позволяет жестко зафиксировать сроки и бюджет при неизменном объеме. Но если объем меняется — сроки срываются.

- ****Agile**** фиксирует бюджет и время, но позволяет менять объем (scope) в рамках итераций. Это предпочтительно, когда важнее выйти в рынок быстро, а не сделать «всё и сразу».

Д. Заказчик и бизнес-контекст

- ****Waterfall**** удобен, когда заказчик хочет получить точную смету до старта и не может участвовать в процессе ежедневно.

- ****Agile**** требует активного вовлечения заказчика (product owner), быстрых решений. Если заказчик недоступен — agile превращается в хаос.

2. Практические сценарии выбора

Сценарий 1: Госсектор, финансы, аутсорсинг с фиксированной ценой

****Характеристики****: жесткие требования регулятора, контракт с фиксированной ценой и сроками, изменения дороги и формализованы.

****Выбор****: ****Waterfall**** или ****V-модель**** (разновидность водопада с акцентом на верификацию).

Почему: предсказуемость бюджета, формальная отчетность, минимизация рисков судебных споров.

Сценарий 2: Стартап, новый продукт, внутренний R&D

****Характеристики****: требования гипотетические, нужно быстро проверить гипотезы, рынок меняется.

****Выбор****: ****Agile**** (Scrum для организации работы, Kanban для потока задач).

Почему: возможность менять приоритеты каждые 1–2 недели, быстрый вывод MVP, низкие потери при смене курса.

Сценарий 3: Крупная ERP-система с четким ядром и вариативной настройкой

****Характеристики****: базовая функциональность строго регламентирована (бухгалтерия), но пользовательские интерфейсы и отчеты могут уточняться.

****Выбор****: ****Гибрид**** (Waterfall для ядра, Agile для кастомизации и внедрения).

Почему: критическая часть делается по жесткому плану, а гибкая часть адаптируется под реальные потребности бизнеса в процессе.

Сценарий 4: Проект с жестким дедлайном (например, к определенной дате)

****Характеристики****: дата несдвигаема (закон, мероприятие), но функциональность можно сократить.

****Выбор****: ****Agile с фиксированным временем**** (time-boxed).

Почему: команда делает максимум возможного к сроку, жертвуя менее важными функциями, а не срывает дату.

3. Дополнительные специализированные подходы

Если классические Waterfall и Agile не подходят, можно рассмотреть:

- ****Lean Development**** — для оптимизации потока ценности, устранения потерь. Хорош для проектов, где важна экономическая эффективность.

- ****DevOps**** — не метод управления проектом, но обязателен для проектов с высокой частотой релизов.

- ****PRINCE2**** — процессный подход, хорошо ложится на водопадные проекты в крупных корпорациях, где нужна строгая governance.

- ****SAFe**** (Scaled Agile Framework) — для масштабирования agile на крупные организации со сложной структурой.

4. Алгоритм выбора (чек-лист)

Чтобы принять решение, пройдите по пунктам:

1. ****Определите степень изменчивости требований****

- Стабильны → Waterfall

- Будут меняться → Agile

2. ****Оцените возможность участия заказчика****

- Заказчик доступен и готов вовлекаться → Agile
- Заказчик дает ТЗ и исчезает → Waterfall
- 3. ****Уточните критичность соблюдения сроков и бюджета****
 - Жесткие и неизменные → Waterfall
 - Гибкие, можно менять объем → Agile
- 4. ****Проанализируйте риски****
 - Риск неправильной архитектуры высок → лучше итеративный подход с ранним прототипированием
 - Риск несоблюдения регуляторных требований высок → формальный подход с этапами контроля
- 5. ****Учтите корпоративную культуру****
 - Бюрократия, строгая иерархия → Waterfall проще внедрить
 - Гибкие команды, плоская структура → Agile

5. Важное замечание: чистота методологии — это миф

На практике большинство проектов используют ****гибридные**** подходы. Например:

- Водопадное планирование верхнего уровня (milestones) + итеративная разработка внутри фаз.
- Scrum с элементами Kanban для стабилизации.
- Waterfall для аппаратной части и Agile для софта.

Главное — не слепое следование методологии, а ****адаптация**** под конкретные условия. Успешный проект — тот, где выбранный подход позволяет эффективно управлять неопределенностью, поддерживать коммуникацию и вовремя доставлять ценность.

Практическое занятие № 5

Тема : Составление плана деловой беседы с заказчиком

Разработка технического задания

Вы переходите от выбора методологии к практическому этапу запуска проекта: ****переговорам с заказчиком и составлению технического задания (ТЗ)****. Это ключевой момент, от которого зависит, будет ли проект предсказуемым, уложится ли в бюджет и не превратится ли в бесконечное согласование.

Ниже — структура деловой беседы и пошаговый подход к формированию ТЗ с учётом того, какой подход (водопадный или гибкий) выбран.

1. План деловой беседы с заказчиком

Цель беседы — не просто записать требования, а ****выявить реальные потребности бизнеса, согласовать формат взаимодействия и зафиксировать границы проекта****.

Этап 1. Подготовка

- Изучите бизнес-домен заказчика (если незнакомы).
- Подготовьте список открытых вопросов (не предполагающих ответ «да/нет»).
- Определите, кто будет участвовать: с вашей стороны — РМ, аналитик, tech lead; со стороны заказчика — инициатор, руководитель, ключевые пользователи.

Этап 2. Открытие и установление контекста

- ****Представьтесь и обозначьте цель встречи****: «Мы здесь, чтобы понять ваши задачи и предложить оптимальный подход к реализации».

- ****Узнайте о бизнес-целях****:

«Какую бизнес-проблему мы решаем? Какие метрики должны измениться после внедрения?»

Это поможет отличать действительно важные требования от второстепенных.

Этап 3. Сбор требований (функциональных и нефункциональных)

Задавайте вопросы, структурируя по слоям:

| Категория | Примеры вопросов |

|-----|-----|

| ****Функциональность**** | «Какие действия пользователь должен выполнять? Опишите основной сценарий успеха».

| ****Пользователи**** | «Кто будет работать с системой? Есть ли разные роли и права?»

| ****Данные и интеграции**** | «Какие данные уже есть? Нужна ли интеграция с учётной системой, CRM, 1С?»

| ****Производительность и надёжность**** | «Сколько пользователей одновременно? Требования к отказоустойчивости?»

| ****Безопасность**** | «Нужно ли соблюдать 152-ФЗ, ФСТЭК? Есть ли требования к аудиту действий?»

| ****Ограничения**** | «Есть ли жёсткие сроки, фиксированный бюджет, обязательные технологии?»

Этап 4. Определение формата работы и методологии

- Если видно, что требования могут меняться → предлагайте гибкий подход (Agile, поэтапное подписание).

- Если требования жёсткие, нужна фиксированная смета → обсуждайте водопадный формат с подробным ТЗ.

- Обязательно проговаривайте ****степень участия заказчика****: «В agile-проекте вы должны уделять 2–3 часа в неделю на приоритизацию бэклога. Готовы?»

Этап 5. Следующие шаги и фиксация

- Зафиксируйте в протоколе: что обсудили, какие решения приняты, какие вопросы остались открытыми.

- Согласуйте дату следующей встречи и формат предоставления документов (ТЗ, бэклог, прототипы).

2. Разработка технического задания (ТЗ)

Техническое задание — это документ (или набор артефактов), который фиксирует договорённости. Его структура зависит от выбранного подхода.

А. Водопадный (классический) подход

ТЗ максимально детальное, утверждается один раз до начала разработки. Состоит из разделов:

1. ****Общие сведения****

- Наименование проекта, стороны, основания для разработки.

2. ****Цели и задачи****

- Бизнес-цели, целевые показатели успеха.

3. ****Требования к функциональности****

- Подробные сценарии (use cases) с описанием входных/выходных данных, алгоритмов, экранных форм.

4. ****Требования к архитектуре и технологиям****

- стек, требования к масштабируемости, способ развёртывания.

5. ****Нефункциональные требования****

- Производительность, безопасность, надёжность, удобство использования.

6. ****Состав и содержание работ****

- Этапы, контрольные точки, артефакты.

7. ****Порядок приёмки****

- Критерии приемки, кто подписывает, сроки устранения замечаний.

8. ****Приложения****

- Прототипы интерфейсов, диаграммы, схемы данных.

> ****Плюс****: единая точка отсчёта, легко контролировать исполнение.

> ****Минус****: сложно менять, высокие риски, если требования изначально определены неверно.

Б. Agile (гибкий) подход

Техническое задание заменяется ****бэклогом продукта**** и ****определением минимально жизнеспособного продукта (MVP)****. Однако для фиксации границ часто создают лёгкий ****Vision Document**** или ****Project Charter****, где описывают:

- ****Цель и целевую аудиторию**** (Product Vision)

- ****Границы проекта**** (что входит в MVP, что — в следующие версии)

- ****Ключевые пользовательские истории**** (User Stories) с критериями приемки

- ****Архитектурные принципы**** (например, использование микросервисов)

- ****Метрики успеха**** (KPI)

Бэклог ведётся в системе управления задачами (Jira, Trello, YouTrack). Каждая история детализируется перед планированием итерации.

В. Гибридный подход

Часто встречается в практике:

- ****Водопадное ТЗ**** на архитектурную часть, интеграции и нефункциональные требования.

- ****Agile-бэклог**** на развитие функциональности внутри утверждённой архитектуры.

Такой подход даёт предсказуемость на уровне инфраструктуры и гибкость на уровне бизнес-логики.

3. Практические советы

1. ****Не начинайте разработку без согласованного документа****

Даже в agile должен быть утверждённый бэклог на ближайшие спринты и понимание границ MVP. Устные договорённости ведут к бесконечным правкам.

2. ****Используйте визуализацию****

Прототипы экранов, схемы процессов, диаграммы интеграций помогают быстрее согласовать, чем страницы текста.

3. ****Фиксируйте критерии приемки****

Каждое требование должно проверяться: «Как мы поймём, что это сделано?» Без критериев вы рискуете получить бесконечную итерацию доработок.

4. ****Учитывайте методологию в плане беседы****

Если вы планируете работать по водопаду, на встрече нужно собрать максимум деталей. Если по agile — достаточно сформулировать эпики и приоритеты, а детали будут уточняться перед спринтом.

5. ****Оставьте место для изменений****

Даже в водопадном проекте предусмотрите процедуру внесения изменений (Change Request). В agile изменения — естественная часть процесса.

4. Пример структуры беседы (шаблон для РМ)

1. ****Введение**** (5 мин)

— Представление, цели встречи, регламент.

2. ****Бизнес-контекст**** (10 мин)

— Почему проект нужен? Какие показатели должны улучшиться?

3. ****Пользователи и сценарии**** (20 мин)

— Кто будет использовать, какие задачи решать? Приоритетные сценарии.

4. ****Технические ограничения**** (10 мин)

— Интеграции, платформы, требования к безопасности, нагрузке.

5. ****Организационные моменты**** (10 мин)

— Доступность заказчика, согласование ТЗ, формат приёмки.

6. ****Следующие шаги**** (5 мин)

— Документ, сроки следующей встречи.

Практическое занятие № 6

Тема 4: Выполнение SWOT-анализа

****SWOT-анализ**** (Strengths, Weaknesses, Opportunities, Threats) — это инструмент стратегического планирования, который помогает оценить проект или методологию с точки зрения внутренних и внешних факторов. В контексте управления IT-проектами SWOT применяют на этапе инициации, выбора подхода или перед составлением ТЗ, чтобы объективно взвесить сильные и слабые стороны, возможности и угрозы.

1. Как провести SWOT-анализ для IT-проекта

Этап 1. Определите объект анализа

Это может быть:

- Весь проект в целом.
- Выбранная методология (Waterfall, Agile, гибрид).
- Техническое решение или архитектура.
- Выход на новый рынок с продуктом.

Этап 2. Соберите команду и данные

Пригласите ключевых стейкхолдеров: РМ, аналитика, tech lead, представителя заказчика. Используйте мозговой штурм, опросы, анализ рисков.

Этап 3. Заполните матрицу 2×2

Внутренние факторы (управляемые)	**Внешние факторы** (неуправляемые)
S (Strengths) — сильные стороны: что у нас хорошо получается, какие ресурсы есть, конкурентные преимущества.	**O (Opportunities)** — возможности: рыночные тренды, новые технологии, потребности клиентов, которые мы можем использовать.
W (Weaknesses) — слабые стороны: чего не хватает, узкие места, недостаток компетенций.	**T (Threats)** — угрозы: риски со стороны конкурентов, регуляторов, нестабильность требований, технические ограничения.

Этап 4. Проанализируйте пересечения

- ****S+O****: как использовать сильные стороны, чтобы реализовать возможности?
- ****S+T****: как сильные стороны помогут снизить угрозы?
- ****W+O****: как устранить слабости, используя внешние возможности?
- ****W+T****: какие комбинации слабостей и угроз критичны? Как избежать?

Этап 5. Сформулируйте выводы и решения

На основе анализа примите решение о подходе, приоритетах, необходимости снижения рисков.

2. Пример: SWOT-анализ для выбора методологии (Waterfall vs Agile)

Strengths (сильные стороны)	**Weaknesses (слабые стороны)**
---	---

| ****Waterfall**** | – Чёткое планирование и контроль
– Предсказуемый бюджет при фикс. ТЗ
– Полная документация | – Сложность внесения изменений
– Позднее тестирование
– Высокий риск несоответствия ожиданиям |

| ****Agile**** | – Гибкость, адаптация к изменениям
– Ранняя поставка ценности (MVP)
– Активное вовлечение заказчика | – Непредсказуемость итоговой стоимости
– Требуется высокой дисциплины команды
– Недостаточная документация |

| ****Opportunities (возможности)**** | ****Threats (угрозы)**** |

|---|---|---|

| ****Waterfall**** | – Чёткая формализация для гос/фин сектора
– Лёгкая передача на аутсорсинг
– Интеграция с жёсткими внешними системами | – Изменение требований со стороны регулятора
– Потеря бюджета при ошибках в ТЗ
– Конкуренты выйдут быстрее |

| ****Agile**** | – Быстрая проверка гипотез на рынке
– Возможность корректировать продукт по обратной связи
– Повышение удовлетворённости заказчика | – Отсутствие активного заказчика («bus factor»)
– Сложность масштабирования в крупной организации
– Риск бесконечного добавления функций |

****Результат анализа**:**

- Если проект требует жёсткой фиксации бюджета и имеет стабильные требования → Waterfall.
- Если требования неясны, важна скорость вывода на рынок → Agile.
- Если критичны интеграции и архитектура, но бизнес-логика гибкая → гибрид.

3. Когда выполнять SWOT-анализ в IT-проекте

- ****На стадии инициации**** — для обоснования выбора подхода или оценки целесообразности проекта.
- ****Перед разработкой ТЗ**** — чтобы понять, какие моменты нужно детализировать, а где оставить пространство для гибкости.
- ****При изменении внешних условий**** (появление нового конкурента, изменение законодательства) — для пересмотра стратегии.
- ****В рамках пост-проектного анализа**** — чтобы накопить опыт для будущих проектов.

4. Практический шаблон (можно использовать в Miro, Excel, документе)

****Проект:**** _____

****Дата:**** _____

****Участники:**** _____

| ****Strengths**** | ****Weaknesses**** |

|-----|-----|

1.	1.	
2.	2.	
3.	3.	

Opportunities		**Threats**	
-----		-----	
1.		1.	
2.		2.	
3.		3.	

****Ключевые выводы и действия:****

- S+O: ...
- W+T: ...
- Решение: ...

5. Рекомендации

- Будьте ****честными**** в оценке слабостей — они не менее важны, чем сильные стороны.
- Не путайте ****возможности**** с сильными сторонами: возможности — это внешние факторы, которые можно использовать; сильные стороны — внутренние.
- После SWOT обязательно ****переходите к плану действий****: что именно вы меняете, какой подход выбираете, какие риски митигируете.

Если вы хотите, я могу помочь провести SWOT-анализ для конкретного проекта — опишите его контекст.

Практическое занятие № 7

Тема : Распределение рисков по вероятности их возникновения и степени влияния Программные измерительные мониторы

Распределение рисков по вероятности и степени влияния — это основа ****количественной и качественной оценки рисков****, которая позволяет определить приоритеты для реагирования. В IT-проектах (особенно при выборе водопада, гибрида или agile) этот подход помогает не утонуть в длинном списке угроз, а сфокусироваться на самых опасных.

1. Матрица вероятности и влияния (Probability-Impact Matrix)

Обычно используют сетку ****5×5**** или ****3×3****, где по горизонтали — вероятность (от очень низкой до очень высокой), по вертикали — степень влияния на проект (затраты, сроки, качество, репутацию).

****Пример уровней:****

Вероятность	Оценка	Влияние	Оценка
< 10%	1 (очень низкая)	< 5% от бюджета / 1–2 дня задержки	1 (незначительное)
10–30%	2 (низкая)	5–10% бюджета / до 1 недели	2 (умеренное)
30–60%	3 (средняя)	10–20% бюджета / до 1 месяца	3 (существенное)
60–90%	4 (высокая)	20–40% бюджета / срыв вехи	4 (критическое)
> 90%	5 (очень высокая)	>40% бюджета / провал проекта	5 (катастрофическое)

2. Как построить матрицу для IT-проекта

Шаг 1. Идентификация рисков

Составьте полный список рисков, разделяя по категориям:

- ****Технические****: сложность интеграции, отказ оборудования, устаревший стек.
- ****Организационные****: текучка ключевых сотрудников, слабая вовлечённость заказчика.
- ****Управленческие****: изменение требований (особенно в водопаде), нереалистичные сроки.

- ****Внешние****: изменение законодательства, действия конкурентов, форс-мажор.

Шаг 2. Оценка вероятности и влияния

По каждому риску экспертным путём (или на основе исторических данных) назначаются оценки. Лучше, если оценку дают несколько участников (PM, tech lead, представитель заказчика) для объективности.

Шаг 3. Размещение на матрице

Постройте сетку и распределите риски по ячейкам. Например:

Влияние / Вероятность	1 (низкая)	2	3	4	5 (высокая)
5 (катастрофическое)			X	X	
4 (критическое)		X	X		
3 (существенное)	X		X		
2 (умеренное)		X			
1 (незначительное)	X				

Цветовая индикация:

- ****Красная зона**** (высокая вероятность + высокое влияние) — требуют немедленного плана реагирования.
- ****Жёлтая зона**** (средние оценки) — мониторинг и резервные планы.
- ****Зелёная зона**** (низкие оценки) — принять или наблюдать.

3. Примеры рисков в разных методологиях

Для водопадного проекта

| Риск | Вероятность | Влияние | Зона |

|-----|-----|-----|-----|

| Ошибка в требованиях, выявленная на тестировании | Средняя (3) | Критическое (4) | Красная |

| Задержка поставки оборудования | Низкая (2) | Существенное (3) | Жёлтая |

| Уход архитектора | Низкая (2) | Катастрофическое (5) | Красная (из-за влияния) |

Для agile-проекта

| Риск | Вероятность | Влияние | Зона |

|-----|-----|-----|-----|

| Заказчик не участвует в приоритизации бэклога | Высокая (4) | Существенное (3) | Красная |

| Невозможность завершить спринт из-за нестабильных требований | Средняя (3) | Умеренное (2) | Жёлтая |

| Технический долг накапливается из-за спешки | Высокая (4) | Умеренное (2) | Жёлтая → требует внимания |

4. Стратегии реагирования в зависимости от зоны

| Зона | Подход | Пример действий |

|-----|-----|-----|

| ****Красная**** (высокий приоритет) | ****Избегание**** или ****передача**** | Изменить архитектуру, чтобы исключить риск; застраховать; нанять внешнего эксперта; зафиксировать ТЗ жёстко. |

| ****Жёлтая**** (средний приоритет) | ****Снижение**** или ****резервирование**** | Добавить буфер в бюджет/сроки; провести дополнительные проверки; усилить контроль на этапах. |

| ****Зелёная**** (низкий приоритет) | ****Принятие**** | Не тратить ресурсы, но отслеживать триггеры; иметь план отката. |

5. Практический шаблон для IT-проекта

| ID | Риск | Вероятность (1–5) | Влияние (1–5) | Интегральный рейтинг (P×I) | Зона |
Стратегия | Ответственный | План действий |

|-----|-----|-----|-----|-----|-----|
|-----|

| R1 | Изменение требований после утверждения ТЗ | 4 | 4 | 16 | Красная | Снижение |
PM, заказчик | Внедрить процедуру Change Request, заложить бюджет на изменения |

| R2 | Сбой сервера в процессе приёмки | 2 | 5 | 10 | Красная | Передача | DevOps |
Использовать облачное резервирование, авто-бекапы |

| R3 | Нехватка фронтенд-ресурсов | 3 | 3 | 9 | Жёлтая | Снижение | HR, PM |
Заблаговременный найм, привлечение фрилансера |

| R4 | Новый релиз конкурента | 1 | 3 | 3 | Зелёная | Принятие | Маркетинг |
Мониторинг, но без остановки разработки |

6. Связь с предыдущими темами

- При выборе методологии (waterfall vs agile) матрица рисков помогает объективно оценить, какие риски доминируют. Если ****изменение требований**** — высокий риск, то agile будет предпочтительнее (вероятность сохраняется, но влияние снижается за счёт итеративности).

- В SWOT-анализе угрозы (Threats) — это входные данные для матрицы рисков. А сильные стороны (Strengths) могут быть использованы для снижения рисков.

- При составлении ТЗ важно отразить результаты оценки рисков: заложить резервы времени, определить процедуру управления изменениями, прописать критерии приемки, чтобы минимизировать споры.

7. Рекомендации по использованию

- ****Обновляйте матрицу регулярно**** (на старте проекта, после каждой крупной вехи, при изменении внешних условий).

- ****Используйте количественные данные****, если есть история проектов — это повышает точность.

- ****Привлекайте команду**** к оценке: разработчики часто видят технические риски, которых нет у РМ.

- ****Не перегружайте матрицу**** десятками рисков; 10–15 наиболее значимых достаточно для управления.

Если хотите, я могу помочь построить такую матрицу для вашего конкретного проекта — опишите контекст.

Практическое занятие № 8

Тема : Как поддерживать регулярную коммуникацию в команде, с заказчиком, с пользователями.

Регулярная коммуникация — это «кровеносная система» IT-проекта. Даже идеально составленное ТЗ и правильно выбранная методология не спасут проект, если информация искажается, теряется или доходит слишком поздно.

Ниже — практическая схема организации коммуникаций для трёх ключевых групп: **команда, заказчик, пользователи.**

1. Внутри команды

Цель: синхронизация по задачам, выявление blockers, обмен знаниями, поддержание морального духа.

Ритуалы (зависят от методологии)

Ритуал	Agile (Scrum/Kanban)	Waterfall / Гибрид
Daily stand-up	Ежедневно, 15 мин: что сделано, что планируется, какие blockers.	Можно реже (2–3 раза в неделю) или в формате короткого синка.
Планирование итерации / этапа	В начале спринта (2–4 недели) — детальное планирование задач.	На старте этапа — распределение работ, уточнение сроков.
Ретроспектива	После каждого спринта — обсуждение улучшений.	После ключевых вех (этапов) — анализ, что пошло не так.
Обзор / демо	В конце спринта — показ результатов команде (и заказчику).	На контрольных точках — демо заказчику; внутри команды реже.
Grooming (уточнение бэклога)	Регулярно (раз в спринт или чаще) — проработка требований.	Аналог — уточнение деталей в рамках утверждённого ТЗ по запросу.

Каналы коммуникации

- **Синхронные:** стендапы (видеозвонок или офлайн), встречи для решения сложных вопросов (ad-hoc).
- **Асинхронные:** чат (Slack, Telegram, Mattermost), доска задач (Jira, Trello, YouTrack), wiki/Confluence для документации.

Принципы

- **Прозрачность:** статусы задач всегда актуальны в системе, чтобы любой член команды мог видеть прогресс.
- **Психологическая безопасность:** люди должны без страха сообщать о проблемах и ошибках.
- **Минимум встреч:** каждое собрание должно иметь чёткую цель и тайминг.

2. С заказчиком

Цель: выравнивание ожиданий, согласование изменений, демонстрация прогресса, сбор обратной связи.

Форматы и частота

Формат	Agile	Waterfall	Рекомендуемая частота
--------	-------	-----------	-----------------------

Формат	Agile	Waterfall	Рекомендуемая частота
Демонстрация результатов	В конце каждого спринта (обзор спринта)	На ключевых вехах (завершение этапа, приёмочные испытания)	Agile: каждые 1–4 недели; Waterfall: по вехам (раз в 1–3 месяца)
Планирование / приоритизация	Перед спринтом — вместе с Product Owner	На старте проекта — утверждение ТЗ; далее — по процедуре Change Request	Agile: регулярно (раз в спринт); Waterfall: при изменении границ
Статус-митинги (Steering Committee)	Кратко (раз в 2–4 недели) по ключевым метрикам, рискам, бюджету	Раз в месяц или по вехам — формальный отчёт	В зависимости от масштаба: раз в 2–4 недели
Оперативное взаимодействие	Product Owner / бизнес-аналитик доступны ежедневно для вопросов	Назначенный представитель заказчика для уточнений	По мере необходимости, но с зафиксированным каналом

Каналы

- **Синхронные:** Zoom/Teams для демо и встреч, телефон для срочных вопросов.
- **Асинхронные:** email для протоколов и официальных решений, общий чат с заказчиком (WhatsApp, Slack) для оперативных уточнений, система управления требованиями (Jira с доступом заказчика).

Ключевые принципы

- **Единое окно коммуникации:** все официальные запросы и решения идут через одного человека (PM или Product Owner), чтобы избежать противоречий.
- **Протоколирование:** после каждой встречи — краткий протокол с решениями, ответственными и сроками.
- **Управление ожиданиями:** если сроки или объём меняются, сообщать заранее, а не постфактум.
- **Адаптация под стиль заказчика:** кто-то любит детальные отчёты, кто-то — короткие дашборды.

3. С пользователями (конечными)

Цель: сбор требований, обратная связь по продукту, вовлечение, обучение, подтверждение ценности.

Когда и как взаимодействовать

Этап	Формат	Цель
До разработки	Интервью, наблюдение, опросы, анализ данных	Понять реальные боли и сценарии
Во время разработки	Прототипы, usability-тестирование, бета-тестирование (по подгруппам)	Проверить гипотезы, скорректировать интерфейсы
После	Сбор обратной связи (NPS, опросы),	Улучшение продукта,

Этап	Формат	Цель
релиза	поддержка, группы roadmap-сессии	пользователей, повышение лояльности

Каналы

- **Синхронные:** фокус-группы, демо-сессии, вебинары.
- **Асинхронные:** формы обратной связи, баг-трекер с доступом для пользователей (Zendesk, UserVoice), рассылки, сообщество в Telegram/Slack.

Особенности для разных подходов

- **Agile:** пользователи могут участвовать в демо спринтов, давать обратную связь, которая сразу попадает в бэклог.
- **Waterfall:** пользователи чаще вовлекаются на этапе сбора требований и на приёмочном тестировании; обратная связь после релиза идёт в следующий проект или версию.

4. Инструменты для поддержки коммуникации

Тип	Примеры	Для чего
Чат	Slack, Telegram, Microsoft Teams, Mattermost	Оперативная команда, каналы с заказчиком
Задачи	Jira, Trello, YouTrack, Asana	Прозрачность статусов, комментарии по задачам
Wiki / Docs	Confluence, Notion, SharePoint	Документация, протоколы, база знаний
Видеозвонки	Zoom, MS Teams, Google Meet	Демо, встречи, стендапы (если удалёнка)
Дашборды	Jira Dashboard, Power BI, Google Data Studio	Визуализация статуса для заказчика

5. Пример расписания коммуникаций (гибридный подход)

Ежедневно (команда): стендап 15 мин.

Раз в неделю:

- Командный grooming / уточнение задач (1 час).
- Синк с Product Owner (30 мин).

Раз в 2 недели:

- Демо спринта с заказчиком (1 час).
- Ретроспектива команды (1 час).
- Отчёт заказчику: дашборд + краткий текст (15 мин на подготовку).

Раз в месяц:

- Стейкхолдер-митинг (Steering Committee) с топ-менеджерами заказчика (1 час).
- Сбор обратной связи от фокус-группы пользователей (по мере необходимости).

6. Общие принципы эффективной коммуникации

1. **Договоритесь о правилах на старте.** Зафиксируйте: каналы, частоту встреч, формат отчётов, время реакции.
2. **Избыточность не вредна, если она структурирована.** Одно и то же решение можно продублировать в протоколе, в задаче Jira и в чате — главное, чтобы информация была непротиворечивой.

3. **Адаптируйте ритм под фазу проекта.** На старте нужно больше встреч для синхронизации, в стабильной разработке — меньше, перед релизом — снова плотнее.
4. **Используйте визуализацию.** Графики, дашборды, прототипы воспринимаются быстрее и однозначнее текста.
5. **Следите за обратной связью.** Если заказчик перестаёт отвечать или команда молчит о проблемах — это первый признак скрытых рисков.

Практическое занятие № 9

Тема : Эффективное использование отчетности и онлайн-инструментов.

1. Принципы эффективной отчетности

1.1. Отчетность должна отвечать на конкретные вопросы

Перед созданием любого отчета спросите: «**Какое решение будет принято на основе этого отчета?**» Если ответа нет — отчет не нужен.

1.2. Разные аудитории — разные форматы

Аудитория	Что важно	Формат
Команда	Статус задач, блокеры, технические риски	Доска задач (Jira, Trello), чат, короткий стендап
Заказчик / бизнес	Прогресс по срокам, бюджет, ключевые вехи, демо функциональности	Дашборд с графиками, слайд-дайджест, демонстрация
Руководство / стейкхолдеры	Общее состояние проекта, риски, отклонения, инвестиции	Steering Committee отчет (1–2 страницы + приложения)

1.3. Регулярность без спама

- **Оперативная отчетность** (daily/weekly) — в инструменты управления задачами, чаты.
- **Тактическая** (раз в 2–4 недели) — демо, обзор спринта, статус-митинг.
- **Стратегическая** (раз в квартал/по вехам) — формальные отчеты для руководства.

1.4. Визуализация вместо текста

- График сгорания задач (burndown chart) показывает тренд лучше, чем список «сделано/не сделано».
- Диаграмма Ганта (для водопада) понятнее таблицы со сроками.
- Цветовые индикаторы (красный/желтый/зеленый) по рискам и статусам.

1.5. Автоматизация сбора данных

Если отчет требует ручного копирования из Jira в Excel — он устарел. Используйте встроенные дашборды или инструменты BI, подключенные к API.

2. Виды отчетов в IT-проектах (с примерами)

2.1. Для Agile (Scrum/Kanban)

Отчет	Что показывает	Инструмент
Burndown / Burnup	Тренд выполнения спринта / всего проекта	Jira, Azure DevOps
Velocity	Скорость команды (story points) для прогнозирования	Jira, таблица
Cumulative Flow	Накопление задач по статусам (узкие места)	Jira, ActionableAgile
Sprint Report	Что завершено, что перенесено, отклонения	Jira, отчет спринта
Lead / Cycle Time	Время от создания задачи до завершения	Jira, Trello (плагины)

2.2. Для Waterfall / Гибрида

Отчет	Что показывает	Инструмент
-------	----------------	------------

Отчет	Что показывает	Инструмент
Гантт с вехами	Факт vs план по этапам	MS Project, Jira (плагины), GanttPRO
Отчет по рискам	Список рисков, статус митигации	Excel, Confluence, Jira (риски)
Приемочные испытания	Процент пройденных тестов, дефекты	TestRail, Jira + Xray, Zephyr
Отчет по изменениям (Change Log)	Все изменения ТЗ, их влияние на бюджет/сроки	Excel, Confluence

2.3. Универсальные для всех подходов

Отчет	Описание
Статус-репорт (1 страница)	Цель, достижения за период, планы, проблемы, риски, запросы к заказчику
Бюджетный трекинг	План vs факт, прогноз на завершение
Дашборд здоровья проекта	5–6 метрик: расписание, бюджет, качество (дефекты), удовлетворенность, риски

3. Онлайн-инструменты: как выбирать и эффективно использовать

3.1. Классификация инструментов

Категория	Примеры	Назначение
Управление задачами	Jira, Trello, Asana, YouTrack, Azure Boards	Единый источник истины по задачам, статусам, назначениям
Документация и wiki	Confluence, Notion, SharePoint	ТЗ, архитектура, протоколы, база знаний
Дашборды и BI	Jira Dashboard, Power BI, Google Looker, Tableau	Визуализация метрик, автоматические отчеты
Коммуникация	Slack, Teams, Mattermost, Telegram	Оперативная переписка, каналы по проектам
Репозитории и CI/CD	GitLab, GitHub, Bitbucket, Jenkins	Код, CI/CD пайплайны, связь с задачами
Тестирование	TestRail, Zephyr, Allure	Управление тест-кейсами, отчеты о дефектах

3.2. Критерии выбора инструментов

- **Совместимость с методологией:** Jira хороша для Scrum, Trello — для Kanban, MS Project — для водопада.
- **Интеграция:** инструменты должны обмениваться данными (например, Jira ↔ Confluence ↔ Slack ↔ Git).
- **Принятие командой:** сложный инструмент, который никто не использует, бесполезен.
- **Доступ для заказчика:** давайте заказчику ограниченный доступ к дашбордам или выделяйте упрощенный интерфейс.

3.3. Как не утонуть в инструментах

- **Единый источник правды (SSOT):** вся информация о статусе задач — только в Jira (или другом основном инструменте). Не допускайте, чтобы параллельно велись Excel-таблицы.
- **Минимум инструментов:** 2–3 основных (задачи + документы + чат) + специализированные по необходимости.
- **Автоматические уведомления:** настройте триггеры (например, при изменении статуса задачи на «блокер» — уведомление в чат).

4. Интеграция отчетности и инструментов в процессы

4.1. Пример: единая экосистема (Jira + Confluence + Slack + Power BI)

1. **Jira** — все задачи, спринты, баги.
2. **Confluence** — ТЗ, протоколы, архитектурные решения. Ссылки на задачи Jira вставлены в документы.
3. **Slack** — каналы: #проект-общий, #деплой, #риски. Интеграция Jira: при создании критического бага приходит уведомление.
4. **Power BI** — автоматический дашборд, обновляемый раз в час, с графиками burndown, velocity, бюджетом. Доступен заказчику через веб-ссылку.

4.2. Пример отчета для заказчика (автоматизированный)

Вместо ручного составления PowerPoint:

- Создаете в Jira дашборд с гаджетами: фильтр выполненных задач за период, burndown, список рисков.
- Делаете PDF-экспорт дашборда раз в 2 недели или даете заказчику доступ к живому дашборду.
- Сопровождаете кратким текстом в Confluence (2–3 абзаца с выводами).

5. Ловушки и как их избежать

Ловушка	Решение
Отчеты ради отчетов	На старте согласуйте с заказчиком и руководством список обязательных отчетов и их периодичность. Отмените те, которые не влияют на решения.
Сложность инструментов	Проведите 1–2 тренинга по Jira/Confluence. Настройте упрощенные дашборды для заказчика.
Разрозненность данных	Интегрируйте инструменты через API. Не разрешайте вести параллельные таблицы.
Низкая дисциплина команды	Сделайте обновление статусов задач частью Definition of Done. Показывайте последствия неактуальных данных (сорванные сроки, неверные прогнозы).
Переизбыток уведомлений	Настройте правила: уведомления только по задачам, где пользователь является исполнителем, наблюдателем или указан в поле.

6. Чек-лист: эффективное использование отчетности и инструментов

- **Определены цели отчетности** для каждой аудитории.
- **Выбран основной инструмент управления задачами**, и вся команда его использует.
- **Настроены дашборды** (burndown, velocity, статус vex) с автоматическим обновлением.
- **Согласован перечень регулярных отчетов** и их формат (живой дашборд / PDF / встреча).
- **Инструменты интегрированы** (задачи ↔ документация ↔ чат).

- **Заказчик имеет доступ** к дашборду или получает регулярный дайджест без ручного труда РМ.
 - **Проводится ретроспектива инструментов** раз в 1–2 месяца: что мешает, что можно упростить.
-

7. Связь с предыдущими темами

- **Выбор методологии** определяет набор отчетов (burndown для agile, Гантт для водопада) и инструментов (Jira с Scrum-доской vs MS Project).
- **SWOT-анализ** помогает выявить слабые стороны в коммуникациях и отчетности, которые можно закрыть правильными инструментами.
- **Управление рисками** требует отчетности по рискам — удобно вести их в Jira или Confluence с регулярным пересмотром.
- **Коммуникация** напрямую зависит от того, насколько удобно и прозрачно информация представлена в инструментах.

Если вы используете конкретный стек (например, Jira + Confluence + Slack) или работаете в крупной организации с жесткими требованиями к отчетности, я могу предложить более детальную настройку дашбордов и регламентов.

Практическое занятие № 10

Тема : Средства создания презентационных материалов: обзор, основные возможности

Средства создания презентационных материалов прошли значительную эволюцию: от классических офисных пакетов до AI-генераторов, создающих слайды по текстовому запросу. Для IT-проекта выбор инструмента зависит от того, кому адресована презентация (команда, заказчик, руководство) и какова её цель (согласование ТЗ, демонстрация прогресса, защита архитектуры).

Ниже представлен обзор основных категорий инструментов с акцентом на возможности, актуальные для проектного управления.

1. Классический офисный софт (Базовый стандарт)

Эти инструменты остаются индустриальным стандартом, особенно в крупных корпорациях с требованиями к информационной безопасности и офлайн-доступом.

Microsoft PowerPoint

Основные возможности:

- **Глубокий контроль дизайна:** неограниченные возможности по настройке анимации, переходов, триггеров и мастер-слайдов.
- **Работа с данными:** бесшовная интеграция с Excel для построения сложных диаграмм и связанных таблиц
- **AI-ассистент (Copilot/Designer):** автоматическая генерация макетов, идей оформления и даже текстов по запросу.
- **Офлайн-доступ:** незаменим для выступлений в местах с нестабильным интернетом.

Когда использовать: подготовка детальных ТЗ для заказчика, архитектурные схемы, финальные отчёты по водопадным этапам, презентации для госсектора.

Google Slides

Основные возможности:

- **Реальное время:** несколько участников (включая удалённых) могут одновременно редактировать слайды, видеть курсоры друг друга и комментировать
- **Интеграция с экосистемой:** автоматическое сохранение в Google Drive, связь с Google Sheets и Docs, встроенные видеозвонки Google Meet.
- **История версий:** позволяет откатить любые изменения, что критично при коллективной работе над ТЗ.

Когда использовать: совместная разработка требований с заказчиком, брейнштормы, ретроспективы, быстрые статус-отчёты для распределённой команды

Apple Keynote

Основные возможности:

- Кинематографические анимации и бесшовная работа внутри экосистемы Apple.
- Интуитивный интерфейс с акцентом на типографику и визуальную эстетику.

Когда использовать: продуктовые презентации, демонстрации для дизайн-команд, выступления на конференциях.

2. AI-генераторы презентаций (Создание за минуты)

Новое поколение инструментов, которое автоматически структурирует информацию, подбирает дизайн и генерирует контент на основе текстового запроса или загруженного документа.

Gamma

Основные возможности:

- Генерация презентации, документа или лендинга по текстовому запросу без ограничений по длине.
- Редактирование структуры *до* генерации: можно скорректировать план слайдов, выбрав уровень детализации (краткий, средний, полный)
- Визуал: использует красивые AI-изображения, текст на русском языке получается грамотным.
- **Формат:** онлайн-презентации с возможностью публикации по ссылке.

Когда использовать: быстрая подготовка черновиков КП (коммерческих предложений), создание основы для питч-деков, структурирование сырых идей

Практическое занятие № 11

Тема: Архивирование документации, закрытие контрактов с поставщиками, финальный отчет.

Завершение проекта — этап, который часто недооценивают, но именно он позволяет зафиксировать результаты, освободить ресурсы, накопить опыт и корректно закрыть финансовые и юридические обязательства. Без системного закрытия проект может «висеть» годами, создавая иллюзию незавершённости, а накопленные знания теряются.

Ниже — пошаговый подход к архивированию документации, закрытию контрактов с поставщиками и подготовке финального отчёта.

1. Архивирование документации

Цель: сохранить всю значимую информацию для дальнейшего использования (сопровождение, аудит, аналогичные проекты) и обеспечить её доступность по регламенту.

Что архивировать

| Категория | Примеры |

|-----|-----|

| ****Управленческая**** | Устав проекта, план управления, протоколы встреч, отчёты по рискам, матрица ответственности |

| ****Техническая**** | ТЗ, архитектурные решения, схемы инфраструктуры, исходные коды (с тегами релизов), инструкции по развёртыванию |

| ****Договорная**** | Контракты, допсоглашения, акты выполненных работ, счета, лицензии |

| ****Приёмочная**** | Акты приёмки, протоколы испытаний, закрывающие документы от заказчика |

| ****Коммуникационная**** | Переписка с ключевыми решениями (если не дублирована в протоколах), презентации |

Принципы архивирования

- ****Единое хранилище****: используйте wiki (Confluence, Notion), систему управления документами (SharePoint, 1С:Документооборот) или сетевую папку с чёткой структурой.

- ****Версионность****: финальные версии документов должны быть помечены, черновики удалены или перемещены в архив черновиков.

- ****Метаданные****: укажите проект, дату закрытия, ответственного за архив, срок хранения (согласно внутренним регламентам или требованиям заказчика).

- ****Доступ****: определите, кто и в каких случаях может получить доступ к архиву (обычно ограниченный круг: РМ, архитектор, юридическая служба).

Для IT-проектов особенно важно

- ****Репозиторий кода****: создайте тег финальной версии, убедитесь, что все ветки слиты, документация в репозитории (README, схемы) актуальна.

- ****Инфраструктурный код (IaC)****: сохраните конфигурации, Docker-образы, скрипты развёртывания.

- ****Базы данных****: если требуется — дампы структуры и обезличенных тестовых данных для будущих доработок.

2. Закрытие контрактов с поставщиками

В IT-проектах поставщиками могут быть: подрядчики по разработке, облачные провайдеры, лицензиары ПО, аппаратное обеспечение, консультанты.

Поставщики услуг/подрядчики

1. **Проверка исполнения обязательств**

- Все ли работы приняты по актам?
- Нет ли открытых гарантийных обязательств?
- Переданы ли исходные коды, документация, доступы?

2. **Финансовое закрытие**

- Сверка взаиморасчётов (акт сверки).
- Оплата финального транша.
- Возврат обеспечительного платежа (если был).

3. **Закрытие доступов**

- Отзыв учётных записей подрядчиков из корпоративных систем, VPN, репозиторий, облачных консолей.
- Смена паролей, если доступы могли быть скомпрометированы.

4. **Оценка поставщика**

- Заполнение внутренней карточки оценки (сроки, качество, коммуникация). Это поможет при выборе партнёров в будущих проектах.

Поставщики лицензий и облачных сервисов

- ****Аудит использования****: отключите неиспользуемые ресурсы (виртуальные машины, БД, серверы), чтобы не платить пост-проект.
- ****Перенос данных****: если проект завершён и сервис больше не нужен, экспортируйте данные и удалите аккаунты.
- ****Пролонгация****: если лицензии будут использоваться на поддержке, переоформите их на операционную команду.

Юридическое закрытие

- Подготовьте ****соглашение о прекращении договора**** или подпишите ****финальный акт сверки**** (если это предусмотрено контрактом).
- Убедитесь, что все конфиденциальные материалы возвращены или уничтожены согласно NDA.

3. Финальный отчёт

Финальный отчёт — это не просто формальность, а документ, который подводит итог для заказчика, руководства и команды. Он служит основой для постпроектного анализа и улучшения процессов.

Структура финального отчёта

Раздел	Содержание
1. Общая информация	Название проекта, период, заказчик, руководитель проекта, цели

| ****2. Достигнутые результаты**** | Выполненные цели, реализованный функционал, достигнутые KPI (сравнение план/факт) |

| ****3. Выполнение бюджета**** | План vs факт, причины отклонений, эффективность использования ресурсов |

| ****4. Соблюдение сроков**** | График (вехи) план/факт, критические задержки и их причины |

| ****5. Качество**** | Количество дефектов, результаты тестирования, удовлетворённость заказчика (если измерялась) |

| ****6. Управление рисками**** | Реализовавшиеся риски, эффективность мер реагирования |

| ****7. Извлечённые уроки (Lessons Learned)**** | Что пошло хорошо, что можно улучшить, рекомендации для будущих проектов |

| ****8. Закрывающие документы**** | Перечень подписанных актов, архивных артефактов |

| ****9. Передача на поддержку**** (если применимо) | Кому передан проект, статус документации, регламенты |

Как подготовить

- ****Используйте данные**** из ранее созданных отчётов (статус-репорты, дашборды), чтобы не собирать информацию заново.
- ****Привлеките команду**** к сбору уроков: проведите ретроспективу (или постмортем) с участием разработчиков, аналитиков, тестировщиков.
- ****Формат****: для руководства — краткая версия (1–2 страницы) с акцентом на ключевые цифры и выводы; полная версия — приложение.

Пример извлечённых уроков (Lessons Learned)

Что сработало хорошо	Что можно улучшить
-----	-----
Ежедневные стендапы помогали быстро выявлять блокеры	Поздно начали интеграционное тестирование, пришлось сдвигать релиз
Использование Jira-дашборда дало заказчику прозрачность	Недостаточно проработали требования к производительности на старте

4. Взаимосвязь с ранее обсуждаемыми темами

- ****Методология****: при водопаде финальный отчёт обычно объёмнее, с детальным сравнением с ТЗ; при agile — акцент на достигнутый бизнес-результат и эволюцию продукта.
- ****Коммуникация****: все ключевые решения, повлиявшие на закрытие, должны быть зафиксированы в протоколах и отражены в финальном отчёте.
- ****Отчётность и инструменты****: именно на этом этапе пригодится автоматизированный сбор данных из Jira, Git, бюджетных систем для минимизации ручной работы.
- ****Риски****: реализовавшиеся риски и их последствия становятся частью Lessons Learned.

5. Чек-лист завершения проекта

- [] ****Все работы приняты заказчиком**** (акты приёмки подписаны).
- [] ****Документация архивирована**** (код, ТЗ, схемы, протоколы) в согласованном месте.
- [] ****Контракты поставщиков закрыты****: сверки, оплаты, возврат обеспечительных, отзыв доступов.
- [] ****Финальный отчёт**** подготовлен и утверждён (заказчиком или внутренним руководством).
- [] ****Бюджет закрыт****: остатки средств возвращены или перенесены, учтены все фактические затраты.
- [] ****Команда расформирована**** или переведена на другие проекты; проведена оценка сотрудников (при необходимости).
- [] ****Передача на поддержку**** (если применимо): оформлен регламент, обучена служба эксплуатации.
- [] ****Проведена ретроспектива / постмортем****, результаты включены в базу знаний организации.

6. Распространённые ошибки на этапе закрытия

- | Ошибка | Последствие |
|---|--|
| ----- | ----- |
| Пропуск формального закрытия контрактов | Поставщики продолжают выставлять счета, доступы остаются открытыми → угроза безопасности |
| Неполный архив | При будущих доработках или аудите тратится много времени на поиск информации |
| Финальный отчёт «для галочки» | Потеря ценного опыта, повторение тех же ошибок в новых проектах |
| Отсутствие передачи на поддержку | После закрытия проекта команда поддержки не готова принимать инциденты |

Основная литература:

.Основные печатные издания

1. 1. Баланов, А. Н. Управление IT-проектами : учебное пособие для СПО / А. Н. Баланов. — 1-е изд. — Санкт-Петербург : Лань, 2024. — 136 с
2. Ермакова, А. Н. Управление ИТ-проектами : учебник / А. Н. Ермакова. — Ставрополь : СтГАУ, 2024 — Часть 1 — 2024. — 196 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/510226> (дата обращения: 30.03.2026). — Режим доступа: для авториз. пользователей.
3. Блюм, М. А. Управление проектами : учебное пособие : в 3 частях / М. А. Блюм, О. В. Коробова, А. В. Сузюмов. — Тамбов : ТГТУ, 2024 — Часть 1 — 2024. — 109 с. — ISBN 978-5-8265-2832-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/472283> (дата обращения: 30.03.2026). — Режим доступа: для авториз. пользователей.

Основные электронные издания

1. Ермакова, А. Н. Управление ИТ-проектами : учебник / А. Н. Ермакова, С. В. Богданова. — Ставрополь : СтГАУ, 2024 — Часть 2 — 2024. — 220 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/510227> (дата обращения: 30.03.2026). — Режим доступа: для авториз. пользователей.
2. Информационные технологии [Электронный ресурс]: учебное пособие / Л. Г. Гагарина, Я. О. Теплова, Е. Л. Румянцева, А. М. Баин; под ред. Л. Г. Гагариной. — М.: ИД «ФОРУМ: ИНФРА-М», 2019. — 320 с. — Режим доступа: <https://znanium.com/catalog/product/1018534>.
3. Плотникова, Н. Г. Информатика и информационно-коммуникационные технологии (ИКТ) [Электронный ресурс]: учеб. пособие / Н. Г. Плотникова. — М.: РИОР: ИНФРА-М, 2019. — 124 с. - Режим доступа: <https://znanium.com/catalog/product/994603>.
4. Суворова, Г. М. Информационные технологии в управлении средой обитания: учебное пособие для вузов / Г. М. Суворова. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2022. — 210 с. — (Высшее образование). — ISBN 978-5-534- 205 14062-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496743>.
5. Федотова, Е. Л. Информационные технологии в профессиональной деятельности [Электронный ресурс]: учебное пособие / Е. Л. Федотова. — М.: ИД «ФОРУМ: ИНФРА-М», 2021. — 367 с. —Режим доступа: <https://znanium.com/catalog/product/1189329>.
6. Коломейченко, А. С. Информационные технологии : учебное пособие для спо / А. С. Коломейченко, Н. В. Польшакова, О. В. Чеха. — 2-е изд., перераб. — СанктПетербург : Лань, 2021. — 212 с. — ISBN 978-5-8114-7565-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/177031>
7. Информационные технологии и основы вычислительной техники : учебник. — Санкт-Петербург : Лань, 2020. — 264 с. — ISBN 978-5-8114-4287-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/148223>.
8. Журавлев, А. Е. Информатика. Практикум в среде Microsoft Office 2016/2019 / А. Е. Журавлев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 124 с. — ISBN 978- 5-507-45070-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/257537>.
9. Журавлев, А. Е. Компьютерный анализ. Практикум в среде Microsoft Excel : учебное пособие для спо / А. Е. Журавлев, А. В. Макшанов, Л. Н. Тындыкарь. — СанктПетербург : Лань, 2020. — 280 с. — ISBN 978-5-8114-5678-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152625>.

10. Калмыкова, С. В. Работа с таблицами на примере Microsoft Excel / С. В. Калмыкова, Е. Ю. Ярошевская, И. А. Иванова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 136 с. — ISBN 978-5-507-44924-8. — Текст : электронный // Лань : электроннобиблиотечная система. — URL: <https://e.lanbook.com/book/249632>.
11. Васильев, А. Н. Числовые расчеты в Excel : учебное пособие для спо / А. Н. Васильев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 600 с. — ISBN 978-5-8114-9367-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/193370>.

1.