1. **Определение целей ИТ-проекта.**

**Цель** – разработать мультимедийное пособие для изучающих фреймворк Flutter с нуля

**Задачи:**

* Анализ предметной области, поиск аналогов и анализ их достоинств и недостатков.
* Определение структуры мультимедийного пособия, включающей теоретическую часть с лекциями и практическую часть с заданиями, упражнениями и лабораторными работами. Форматы пособия могут включать исполняемый файл (exe + dll), электронный курс на платформе LMS Moodle и другие форматы. Практическая часть должна включать установленное ПО для разработки Flutter, задания в электронном виде в формате pdf, систему контроля в LMS Moodle и возможность выполнения онлайн-лабораторных работ с дистанционным доступом.
* Определение оптимального ПО для разработки мультимедийного пособия, такого как инструменты для создания видеолекций, интерактивных заданий и лабораторных работ.
* Формирование команды разработчиков, авторов контента и тестировщиков для создания мультимедийного пособия.
* Установка сроков выполнения каждой задачи, определение временных рамок и графика разработки мультимедийного пособия.
* Определение источников финансирования проекта, таких как бюджет организации, спонсоры, гранты или другие источники финансирования.

1. **Анализ предметной области для поставленной задачи.**

**Flutter** — это открытый и бесплатный фреймворк разработки мобильных приложений, разработанный компанией Google. Он позволяет создавать кроссплатформенные мобильные приложения с использованием одного исходного кода, который может быть скомпилирован в нативный код для iOS, Android и других платформ.

**Ключевые особенности:**

* **Кроссплатформенность**: Flutter позволяет разработчикам создавать мобильные приложения, работающие одинаково на разных платформах, таких как iOS, Android, macOS, Windows и других.
* **Быстрая разработка интерфейсов**: Flutter предлагает горячую перезагрузку (hot reload) функцию, которая позволяет разработчикам мгновенно видеть изменения в интерфейсе приложения без перезапуска приложения.
* **Однородный пользовательский интерфейс**: Flutter использует собственный набор виджетов (widgets), что позволяет создавать красивые и интерактивные пользовательские интерфейсы, выглядящие одинаково на разных платформах.
* **Богатый набор инструментов**: Flutter предоставляет разнообразные инструменты и библиотеки для работы с анимациями, сетью, базами данных, состояниями и другими аспектами разработки мобильных приложений.
* **Язык программирования Dart**: Flutter использует язык программирования Dart, который предлагает простой и выразительный синтаксис, статическую типизацию, асинхронное программирование и другие современные возможности.

**Применение**: Flutter широко используется для разработки мобильных приложений разных масштабов и сложности, включая приложения для социальных медиа, финансовых сервисов, электронной коммерции, игр и других областей.

**Основные преимущества:**

* **Быстрая разработка**: Благодаря горячей перезагрузке и однородному пользовательскому интерфейсу, разработка на Flutter может быть быстрой и эффективной.
* **Кроссплатформенность**: Возможность создания кроссплатформенных приложений с использованием одного исходного кода позволяет сократить время и затраты на разработку и поддержку приложений на разных платформах.
* **Богатый набор инструментов**: Flutter пред Красивый пользовательский интерфейс: Использование собственных виджетов Flutter позволяет создавать привлекательные и интерактивные пользовательские интерфейсы, выглядящие одинаково на разных платформах.
* **Поддержка сообщества**: Flutter имеет активное сообщество разработчиков, что обеспечивает поддержку, обновления и регулярные релизы, а также обмен опытом и решениями проблем.

**Ограничения:**

* Ограниченные возможности нативных API: В некоторых случаях, особенно для более сложных и специфичных функциональностей, могут потребоваться обращения к нативным API, что может потребовать дополнительных усилий и знаний.
* Размер приложения: Размер приложений, разработанных на Flutter, может быть немного больше по сравнению с нативными приложениями, из-за включения движка Flutter и Dart Runtime.
* Опыт разработки: Хотя Flutter предлагает простой и выразительный синтаксис языка Dart, разработка на Flutter может потребовать некоторого времени и опыта для освоения всех его возможностей и особенностей.

**Актуальность:** Flutter является популярным фреймворком разработки мобильных приложений и продолжает активно развиваться. Он имеет широкое сообщество разработчиков, активную поддержку со стороны Google, и используется множеством компаний и команд разработчиков по всему миру.

**Вывод**: Flutter — это мощный и гибкий фреймворк разработки мобильных приложений, предлагающий кроссплатформенность, быструю разработку интерфейсов, богатый набор инструментов и красивые пользовательские интерфейсы. Однако, как и любой другой инструмент, у него также есть ограничения, такие как ограниченные возможности нативных API и размер приложения. В целом, Flutter остается популярным выбором для разработки мобильных приложений, особененно для создания кроссплатформенных приложений с одним кодом для разных платформ. Он предлагает удобную разработку интерфейсов, поддержку различных устройств и операционных систем, а также активное сообщество разработчиков. Однако, перед выбором Flutter в качестве фреймворка для разработки мобильных приложений, разработчикам стоит учитывать его ограничения и ознакомиться с его особенностями и требованиями для достижения наилучших результатов. В целом, Flutter предлагает множество преимуществ и остается одним из ведущих инструментов для создания мобильных приложений на разных платформах.

1. **Выявление и формулировка требований к ИТ-проекту.**
2. **Функциональные требования:**

* Разработка мультимедийного пособия, включающего теоретическую часть (лекции, теория) и практическую часть (задания, упражнения, лабораторные работы).
* Возможность изучения фреймворка Flutter с нуля, включая основы разработки мобильных приложений, создание интерфейсов, управление состоянием, работу с API и другими основными функциональностями.
* Интерактивные элементы, такие как тесты, вопросы для самопроверки, интерактивные примеры кода и демонстрации.

1. **Управленческие требования:**

* Определение конкретных сроков завершения ИТ-проекта, включая этапы разработки, тестирования и выпуска готового мультимедийного пособия.
* Определение ролей и обязанностей команды разработки, включая разработчиков, дизайнеров, тестировщиков и других участников проекта.
* Определение процессов управления проектом, таких как планирование, контроль, отчетность и коммуникация между участниками проекта.

1. **Эргономические требования:**

* Удобный и интуитивно понятный пользовательский интерфейс мультимедийного пособия, который позволит изучающим эффективно осваивать материалы.
* Адаптивный дизайн, который будет корректно отображаться на различных устройствах и экранах, включая мобильные устройства, планшеты и компьютеры.

1. **Архитектурные требования:**

* Модульная архитектура мультимедийного пособия, позволяющая легко добавлять, изменять или удалять контент.
* Использование современных технологий и паттернов разработки, таких как MVP, MVVM или Clean Architecture, для обеспечения масштабируемости и поддерживаемости кода.

1. **Взаимодействия:**

* Возможность взаимодействия с другими участниками обучения, такими как преподаватели и студенты, через форумы, чаты или другие инструменты коммуникации.
* Возможность отслеживания прогресса изучения материалов, таких как выполнение заданий, прохождение тестов и получение обратной связи.

**6. Требования сервисного уровня:**

* Обеспечение высокой доступности и надежности мультимедийного пособия, чтобы пользователи могли получить качественный опыт изучения без проблем с доступом или сбоями в работе.
* Защита пользовательских данных и конфиденциальности, включая соблюдение требований по защите персональных данных и других законодательных норм в области конфиденциальности.
* Поддержка различных языков и локализация контента, чтобы обеспечить доступность для пользователей разных стран и культур.
* Возможность обновления контента и функциональностей мультимедийного пособия, чтобы поддерживать актуальность и соответствие современным требованиям изучения фреймворка Flutter.

**7. Требования к аппаратной платформе, операционной системе и дополнительному ПО:**

* Поддержка работы мультимедийного пособия на различных аппаратных платформах, таких как мобильные устройства (смартфоны и планшеты) на операционных системах Android и iOS, а также на компьютерах с различными операционными системами (Windows, macOS, Linux).
* Зависимости от дополнительного ПО, такого как интегрированные среды разработки (IDE) для Flutter (например, Android Studio, Visual Studio Code), эмуляторы и симуляторы устройств для тестирования мобильных приложений, а также другие инструменты и библиотеки, необходимые для разработки и запуска приложений на платформе Flutter.

**8. Дополнительные требования:**

* Совместимость с существующими учебными материалами и курсами, связанными с изучением фреймворка Flutter, чтобы мультимедийное пособие могло быть эффективно интегрировано в существующие образовательные программы.
* Удовлетворение требований целевой аудитории, включая начинающих разработчиков, имеющих базовые знания программирования, но не знакомых с фреймворком Flutter.
* Удовлетворение требований заказчика и его бизнес-целей, включая бюджетные ограничения, сроки и ожидания по функциональности и качеству мультимедийного пособия.

1. **Разработка структуры декомпозиции работ ИТ-проекта и календарное планирование**
2. **Подготовительный этап**
   * Определение требований к мультимедийному пособию (2 дня)
   * Подготовка проектного плана, включая определение задач и ресурсов (2 дня)
   * Веха: Утверждение и одобрение проектного плана
3. **Анализ и проектирование**
   * Исследование и изучение фреймворка Flutter (5 дней)
   * Разработка структуры и дизайна мультимедийного пособия (5 дней)
   * Веха: Утверждение структуры и дизайна мультимедийного пособия
4. **Разработка контента**
   * Создание мультимедийных элементов (графика, видео, аудио и т. д.) (7 дней)
   * Написание текстового контента (5 дней)
   * Веха: Завершениеразработки контента
5. **Разработка программного кода**
   * Разработка интерактивных модулей и функций при помощи Flutter (10 дней)
   * Тестирование и отладка разработанных модулей (5 дней)
   * Веха**:** Завершение разработки программного кода
6. **Тестирование и отладка**
   * Проведение тестирования мультимедийного пособия (7 дней)
   * Исправление ошибок и отладка (2 дня)
   * Веха: Завершение тестирования и отладки
7. **Завершение проекта**
   * Подготовка окончательной документации (3 дня)
   * Верификация выполненных задач и закрытие проекта (1 день)
   * Веха: Завершение проекта и передача результата заказчику
8. **Сетевая модель.**

Выбор сетевой модели для создания проектного плана в Microsoft Project зависит от характеристик проекта, его сложности, продолжительности, уровня детализации задач, доступных ресурсов и требований к управлению проектом. В Microsoft Project доступны несколько сетевых моделей, таких как:

1. Модель "Диаграмма Гантта" (Gantt Chart): Это классическая модель, которая представляет проект в виде горизонтального графика с осью времени и задачами, расположенными на оси. Диаграмма Гантта позволяет визуально отслеживать прогресс выполнения задач, определять зависимости между задачами и ресурсами, и управлять ресурсами.
2. Модель "Сетевая диаграмма" (Network Diagram): Это модель, основанная на технике "Диаграмма стрелок" или "Диаграмма прецедентов", которая позволяет моделировать зависимости между задачами, определять критический путь и оценивать длительность проекта. Сетевая диаграмма в Microsoft Project позволяет более детально определить логику выполнения задач и распределение ресурсов.
3. Модель "PERT-анализ" (Program Evaluation Review Technique): Это модель, основанная на методе PERT, который используется для оценки вероятных продолжительностей задач с учетом степени неопределенности. Модель PERT позволяет учесть возможные риски и неопределенности при определении длительности задач.

**В нашем проекте была выбрана диаграмма Гантта. Вот некоторые достоинства модели "Диаграмма Гантта":**

1. **Простота визуализации:** Диаграмма Гантта представляет проект в виде горизонтального графика, где задачи отображаются на оси времени. Это позволяет легко визуализировать и понять последовательность и длительность задач, а также их зависимости и отношения с другими задачами.
2. **Легкость создания и редактирования:** В Microsoft Project создание и редактирование диаграммы Гантта осуществляется с помощью интуитивно понятного интерфейса, что делает процесс разработки проектного плана достаточно простым и удобным даже для новичков. Задачи могут быть легко добавлены, удалены, изменены или перемещены на графике с использованием простых действий.
3. **Возможность отслеживания прогресса**: Диаграмма Гантта позволяет отслеживать прогресс выполнения задач, помечать выполненные задачи, отображать процент выполнения, а также определять завершенность проекта. Это позволяет контролировать выполнение проекта и вносить необходимые корректировки в расписание и ресурсы при необходимости.
4. **Управление ресурсами**: Диаграмма Гантта также предоставляет возможности управления ресурсами, такими как назначение ресурсов на задачи, определение их загрузки и распределения, а также отслеживание затрат ресурсов на выполнение задач. Это помогает управлять доступными ресурсами проекта и оптимизировать их использование.
5. **Представление критического пути**: Диаграмма Гантта в Microsoft Project также позволяет определить критический путь - последовательность задач, определяющую наиболее длительный путь выполнения проекта. Это помогает идентифицировать наиболее критические задачи, определить резерв времени и управлять временными ограничениями проекта.
6. Метод критического пути.

**Метод критического пути** (Critical Path Method, CPM) является инструментом проектного управления, который позволяет определить наиболее критические задачи и последовательность выполнения задач, определяющую наиболее длительный путь выполнения проекта. Для данного проекта, связанного с разработкой мультимедийного пособия для изучающих фреймворк Flutter, Метод критического пути может быть применен следующим образом:

1. **Определение задач и их зависимостей**: Вначале необходимо определить все задачи, необходимые для выполнения проекта, и их зависимости друг от друга. Задачи могут включать, например, разработку мультимедийного контента, создание учебных материалов, тестирование и документирование проекта.
2. **Определение продолжительности задач**: Для каждой задачи необходимо определить ее продолжительность, то есть оценить время, необходимое для ее выполнения. Это может включать оценку времени, затраченного на выполнение различных задач, таких как создание графики, запись видео, подготовка текстов и т.д.
3. **Создание сетевой модели**: Далее необходимо создать сетевую модель, которая отображает все задачи и их зависимости в виде графа. В этой модели задачи представлены в виде узлов, а зависимости между задачами - в виде стрелок, указывающих на направление выполнения задач.
4. **Определение ранних и поздних сроков**: После создания сетевой модели можно определить ранние и поздние сроки выполнения каждой задачи. Ранние сроки (ES - Early Start) определяются исходя из начала проекта и продолжительности задач, а также зависимостей между задачами. Поздние сроки (LS - Late Start) определяются исходя из завершения проекта и продолжительности задач, а также связей между задачами и критического пути.
5. **Определение полного резерва времени**: Полный резерв времени (Total Float, TF) — это разница между поздними и ранними сроками выполнения задачи. Задачи с полным резервом времени равным нулю находятся на критическом пути и являются наиболее критическими задачами, определяющими последовательность выполнения проекта и его длительность.
6. **Определение критического пути**: Критический путь — это последовательность задач с нулевым полным резервом времени, то есть задачи, которые не могут быть задержаны без задержки всего проекта. Определение критического пути позволяет идентифицировать наиболее критические задачи, определяющие общую длительность проекта.
7. **Мониторинг и управление проектом**: После определения критического пути и оценки продолжительности проекта, метод критического пути может быть использован для мониторинга и управления проектом. Если одна из задач на критическом пути задерживается, это может привести к задержке всего проекта, и соответствующие меры могут быть предприняты для минимизации этого влияния, например, ресурсное перераспределение или сокращение длительности критических задач.

Применение метода критического пути в проекте по разработке мультимедийного пособия для изучения фреймворка Flutter позволит идентифицировать наиболее критические задачи, определить последовательность выполнения задач, определить общую длительность проекта и производить мониторинг и управление проектом с целью соблюдения сроков и достижения успеха проекта.

1. Оценка времени и затрат.

Примерная оценка времени и затрат для проекта разработки мультимедийного пособия для изучения фреймворка Flutter с нуля может варьироваться в зависимости от множества факторов, таких как сложность проекта, опыт команды, стоимость ресурсов и других.

Однако, в качестве ориентировочных значений, можно предположить **следующую оценку времени**:

1. Исследование и планирование проекта: 2 дня (16 часов)
2. Создание дизайна и интерфейса: 5 дней (40 часов)
3. Разработка функциональности: 20 дней (160 часов)
4. Создание мультимедийных элементов (графика, видео, аудио и т.д.): 10 дней (80 часов)
5. Тестирование и отладка: 5 дней (40 часов)
6. Документация: 3 дня (24 часа)

Общая оценка времени: 45 дней (360 часов)

Оценка затрат на проект может варьироваться в зависимости от стоимости ресурсов, используемых в проекте, и может быть уточнена на основе конкретных данных о расценках на российском рынке. В качестве примера, предположим, что средняя стоимость одного часа работы разработчика составляет 1000 рублей, а стоимость мультимедийных элементов составляет 5000 рублей.

**Оценка затрат на проект:**

1. Исследование и планирование проекта: 16000 рублей (16 часов x 1000 рублей/час)
2. Создание дизайна и интерфейса: 40000 рублей (40 часов x 1000 рублей/час)
3. Разработка функциональности: 160000 рублей (160 часов x 1000 рублей/час)
4. Создание мультимедийных элементов: 40000 рублей (80 часов x 500 рублей/час)
5. Тестирование и отладка: 40000 рублей (40 часов x 1000 рублей/час)
6. Документация: 24000 рублей (24 часа x 1000 рублей/час)

Общая оценка затрат на проект: 460000 рублей

Однако, стоит отметить, что это всего лишь примерная оценка, и реальные затраты на проект могут варьироваться в зависимости от ряда факторов, таких как реальные расценки на российском рынке, дополнительные требования проекта, изменения в ходе работы и другие факторы. Поэтому рекомендуется провести более детальную оценку затрат, учитывая конкретные условия и требования проекта, а также согласовать ее с соответствующими заинтересованными сторонами и командой проекта перед его началом.

1. Смета расходов.

Примерная смета расходов проекта "Разработка мультимедийного пособия для изучающих фреймворк Flutter с нуля":

1. **Затраты на персонал:**

* Проектный менеджер: 45 дней \* 8 часов/день \* ставка проектного менеджера
* Разработчики: 45 дней \* 8 часов/день \* ставка разработчика
* Дизайнер: 45 дней \* 8 часов/день \* ставка дизайнера
* Тестировщик: 45 дней \* 8 часов/день \* ставка тестировщика

1. **Затраты на аппаратную платформу и программное обеспечение**:

* Компьютеры и периферийное оборудование для команды разработки
* Лицензии на программное обеспечение, такие как операционная система, разработческие инструменты и другое

1. **Затраты на организацию и администрирование**:

* Аренда офиса или рабочих мест для команды проекта
* Интернет-соединение, телефония и другие коммуникационные услуги
* Расходы на организацию внутренней коммуникации и взаимодействия команды проекта

1. **Затраты на материалы и ресурсы**:

* Расходы на покупку или создание мультимедийных контентов, таких как графика, аудио и видео
* Затраты на покупку или аренду литературы, учебных материалов или других источников для создания пособия

1. **Затраты на маркетинг и продвижение**:

* Расходы на рекламу и продвижение мультимедийного пособия, такие как создание сайта, контекстная реклама, социальные сети и другие маркетинговые активности

1. **Затраты на другие необходимые ресурсы и расходы:**

* Расходы на обучение и сертификацию команды проекта, если требуется дополнительное профессиональное образование
* Расходы на организацию и проведение совещаний, презентаций и других мероприятий, связанных с проектом

Итоговая смета расходов будет зависеть от конкретных ставок оплаты персонала, стоимости аппаратной платформы, программного обеспечения и т.п.

Участники ИТ-проектов и их роли.

* Проектный менеджер: Соловых Мария
* Разработчики: Саранцева Алина, Садриев Реназ, Закревский Владислав, Проснеков Игнатий, Ершов Егор
* Дизайнеры: Колосова Ксения, Калюгина Мария
* Тестировщики: Янсон Виталия, Сидоренко Маргарита, Караджаев Максим, Шибанова Алёна

**Разработка проекта**

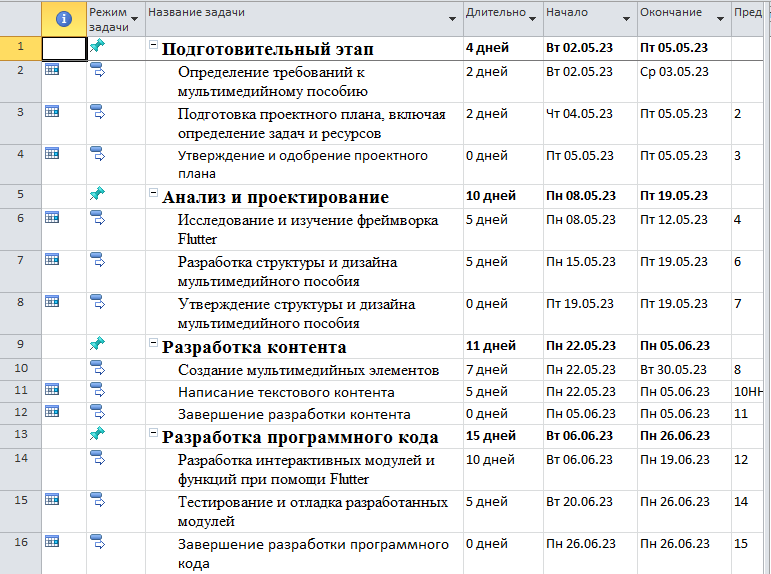


Рисунок 1 – Назначение задач

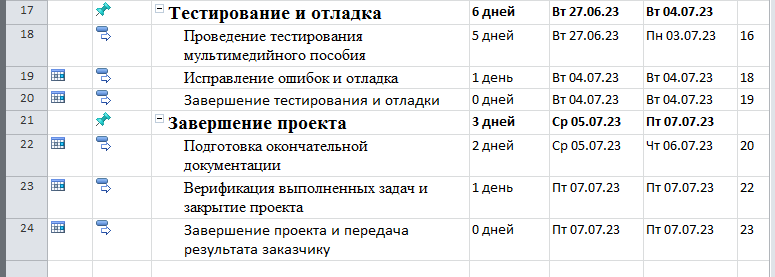


Рисунок 2 – Назначение задач

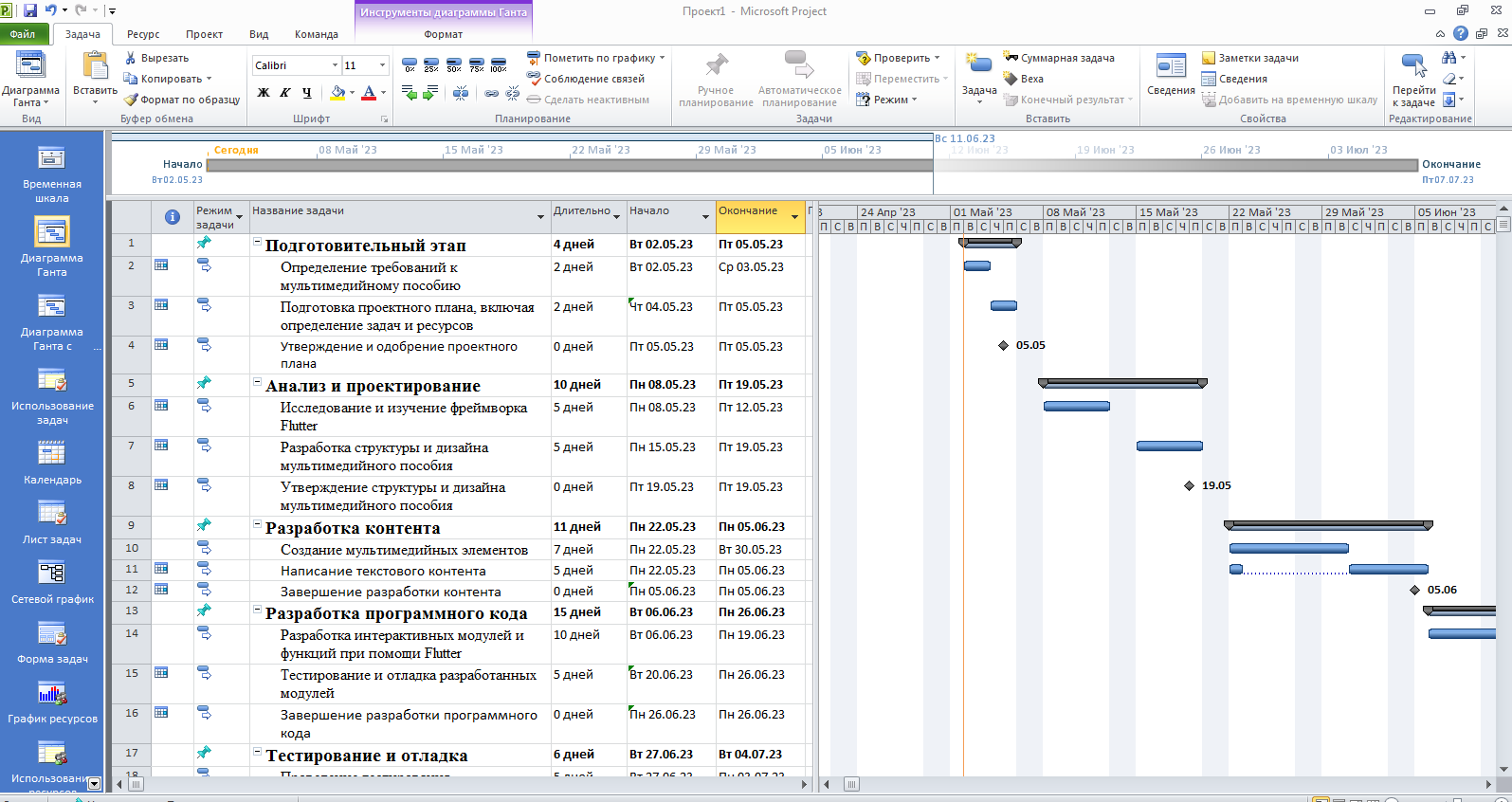


Рисунок 3 – Планирование времени

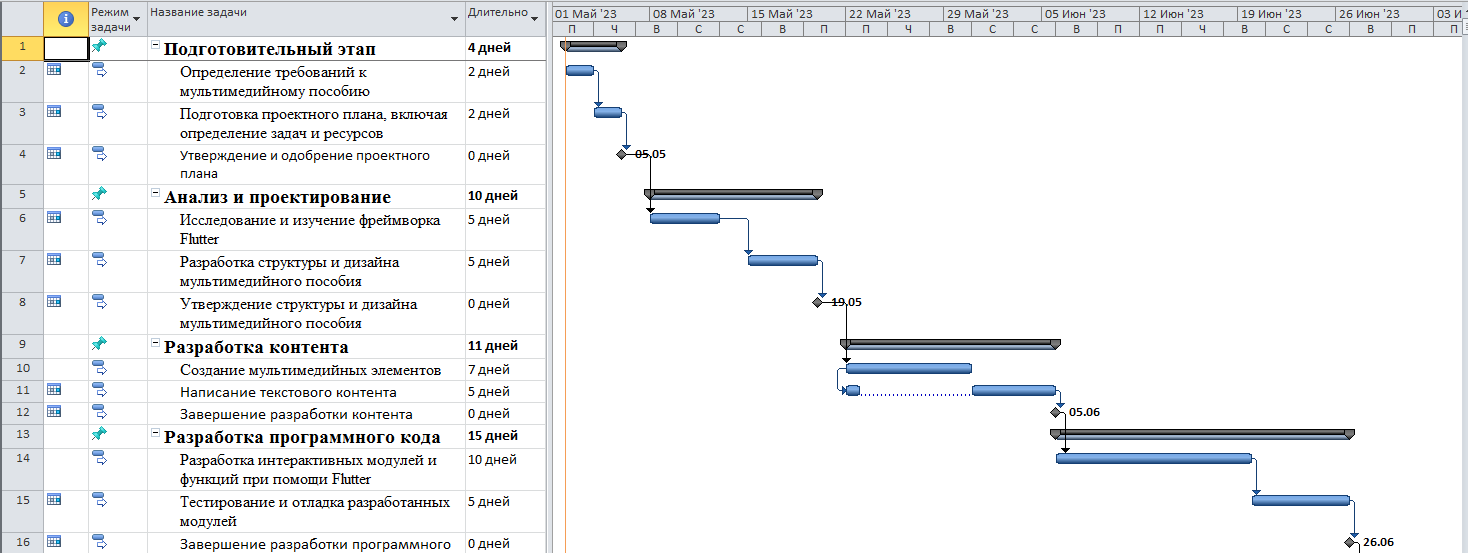


Рисунок 4 –Установление связей между задачами