

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра информационных систем управления

МАРТЫНЮК Валерия Валентиновна

**ОПТИМИЗАЦИЯ РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ НА
ОСНОВЕ МОНИТОРИНГА СТАТУСА ЗАДАЧ**

Дипломная работа

Научный руководитель:
старший преподаватель
И. Г. Орешко

Допущена к защите

« ____ » _____ 20__ г.

Заведующий кафедрой информационных систем управления,
доктор технических наук, доцент А.М. Недзьведь

Минск, 2025

ОГЛАВЛЕНИЕ

Перечень условных обозначений, символов и терминов	4
Введение.....	8
Глава 1 СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ПЛАНИРОВАНИЯ	10
1.1 Исследование инструментов для управления задачами и рабочим временем.....	10
1.2 Обзор классических методов планирования	11
1.2.1 Метод сетевого графика: визуализация и анализ зависимостей.....	11
1.2.2 Метод критического пути(CPM): принципы и математическая модель.....	12
1.2.3 Метод оценки и пересмотра программ (PERT): вероятностная природа задач.....	13
1.2.4 Метод потенциалов (MPM): обратный анализ графа.....	14
1.2.5 Диаграмма Ганта	15
1.3 Математические основы сетевого планирования	16
1.3.1 Использование ориентированных ациклических графов (DAG)....	17
1.4 Сравнение методов по ключевым критериям	20
1.5 Современные программные инструменты планирования	22
1.5.1 Microsoft Project.....	22
1.5.2 GanttPRO	23
1.5.3 ClickUp	24
1.5.4 Общая сравнительная характеристика.....	25
1.6 Обоснование выбора метода для реализации	26
1.7 Выводы по первой главе.....	26
Глава 2 ПРОЕКТИРОВАНИЕ ИНСТРУМЕНТА ПЛАНИРОВАНИЯ.....	28
2.1 Практическое применение приведенных методов	28
2.1.1 Определение критического пути	28
2.1.2 Определение параметров работ сетевого графика	29
2.2 Требования к приложению.....	30
2.3 Проектирование структуры проекта	31
2.4 Выводы по второй главе.....	33
Глава 3 РЕАЛИЗАЦИЯ ИНСТРУМЕНТА ПЛАНИРОВАНИЯ.....	34
3.1 Основные элементы веб-приложения	34

3.2 Используемые технологии и архитектура веб-приложения.....	36
3.3 Реализация элементов веб-приложения.....	37
3.4 Реализация функционала веб-приложения.....	37
3.5 Тестирование приложения	39
3.6 Выводы к третьей главе.....	40
Заключение	41
Список использованных источников	43
Приложение А Листинг основного файла.....	44
Приложение Б Пример хранения задач	46

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

API – программный интерфейс приложения (Application Programming Interface),

REST – передача репрезентативного состояния (Representational State Transfer),

CPM – метод критического пути (Critical Path Method),

PERT – метод оценки и пересмотра программ (Program Evaluation and Review Technique),

MPM – метод потенциалов (Material Point Method).

РЕФЕРАТ

Структура и объём дипломной работы

46 страниц, 17 рисунков, 1 таблица, 2 приложения, 8 источников

Ключевые слова: ДИАГРАММА ГАНТА, УПРАВЛЕНИЕ ПРОЕКТАМИ, WEB-APPLICATION, PYTHON, FLASK.

Объект исследования – методы и инструменты планирования проектов.

Предмет исследования – разработка веб-приложения для визуализации и управления задачами на основе диаграммы Ганта.

Цель исследования – создание автоматизированного инструмента для оптимизации планирования и контроля выполнения проектов.

Методы исследования – анализ литературы, проектирование архитектуры, программирование, тестирование.

Полученные результаты и их новизна – разработано веб-приложение с функционалом построения диаграммы Ганта, расчета критического пути и мониторинга задач.

Достоверность материалов и результатов – подтверждена тестированием и сравнением с аналогами.

Область возможного практического применения – управление проектами в IT, строительстве, маркетинге и других сферах.

РЭФЕРАТ

Структура і аб'ём дыпломнай працы

46 старонак, 17 малюнкаў, 1 табліца, 2 дадатака, 8 крыніц

Ключавыя словы: ДЫЯГРАМА ГАНТА, КІРАВАННЕ ПРАЕКТАМІ, WEB-APPLICATION, PYTHON, FLASK.

Аб'ект даследавання – метады і інструменты планавання праектаў.

Прадмет даследавання – распрацоўка вэб-прыкладання для візуалізацыі і кіравання задачамі на аснове дыяграмы Ганта.

Мэта даследавання – стварэнне аўтаматызаванага інструмента для аптымізацыі планавання і кантролю выканання праектаў.

Метады даследавання – аналіз літаратуры, праектаванне архітэктур, праграмаванне, тэставанне.

Атрыманыя вынікі і іх навізна – распрацавана вэб-прыкладанне з функцыяналам пабудовы дыяграмы Ганта, разліку крытычнага шляху і маніторынгу задач.

Даставернасць матэрыялаў і вынікаў – пацверджана тэставаннем і параўнаннем з аналагамі.

Вобласць магчымага практычнага прымянення – кіраванне праектамі ў ІТ, будаўніцтве, маркетынгу і іншых галінах.

SUMMARY

Structure and scope of the diploma work

46 pages, 17 figures, 1 table, 2 appendix, 8 references

Keywords: GANTT CHART, PROJECT MANAGEMENT, WEB-APPLICATION, PYTHON, FLASK.

Object of study – methods and tools for project planning.

Subject of study – development of a web application for task visualization and management based on the Gantt chart.

Research goal – creation of an automated tool for optimizing project planning and control.

Research methods – literature analysis, architecture design, programming, testing.

Results and novelty – a web application with Gantt chart visualization, critical path calculation, and task monitoring was developed.

Reliability of materials and results – confirmed by testing and comparison with analogues.

Practical application – project management in IT, construction, marketing, and other fields.

ВВЕДЕНИЕ

В условиях динамично развивающихся технологий и возрастающей сложности проектов эффективное управление задачами и временем сотрудников становится критически важным для успешной реализации проектов. Современные организации сталкиваются с проблемами несвоевременного выполнения задач, неравномерной загрузки персонала и отсутствия прозрачности в мониторинге прогресса. Традиционные методы, такие как таблицы и ручное планирование, не обеспечивают необходимой гибкости и наглядности. В связи с этим возрастает потребность в автоматизированных инструментах визуализации и управления проектами [1].

Разработанное в рамках дипломной работы веб-приложение для планирования и мониторинга задач на основе диаграммы Ганта позволяет не только оптимизировать рабочие процессы, но и минимизировать риски задержек за счет автоматизированного расчета критического пути и анализа загрузки ресурсов. Актуальность темы обусловлена необходимостью внедрения современных цифровых решений в управление проектами для повышения производительности и прозрачности рабочих процессов.

Целью работы является разработка веб-приложения для визуализации и управления проектами с использованием диаграммы Ганта, обеспечивающего автоматизацию планирования, контроль выполнения задач и анализ эффективности распределения ресурсов.

Для достижения поставленной цели решаются следующие задачи:

1. Провести анализ существующих методов и инструментов управления проектами (сетевые графики, PERT, диаграмма Ганта).
2. Определить требования к функциональности и архитектуре веб-приложения.
3. Реализовать алгоритм расчета критического пути и построение диаграммы Ганта.
4. Реализовать клиент-серверное приложение с использованием современных технологий (Flask, SQLite).
5. Протестировать работоспособность приложения и оценить его эффективность.

Хронологические рамки исследования охватывают период разработки и тестирования приложения. Выбор данного периода обусловлен актуальностью темы и необходимостью внедрения цифровых решений в управление проектами в современных условиях.

В работе применялись следующие методы:

- Теоретические: анализ научной литературы, сравнение методов планирования.
- Практические: проектирование архитектуры приложения, разработка алгоритмов, программирование на Python и JavaScript, тестирование.
- Экспериментальные: оценка производительности системы, анализ эффективности внедрения.

Внедрение разработанного приложения позволяет:

- Сократить время на планирование задач за счет автоматизации.
- Уменьшить количество ошибок при распределении ресурсов.
- Повысить прозрачность контроля выполнения проектов.

При эксплуатации веб-приложения должны соблюдаться стандартные требования информационной безопасности:

- Защита персональных данных.
- Использование HTTPS для шифрования передаваемой информации.
- Регулярное резервное копирование данных для предотвращения потерь.

Дипломная работа состоит из трех глав. В целях выбора оптимального подхода к визуализации и управлению задачами, в первой главе представлено теоретическое обоснование существующих методов временного планирования и их применимость в условиях современных проектов, во второй — проектированием веб-приложения, в третьей — практическая реализация веб-приложения.

Разработанное решение может быть использовано в компаниях различных сфер деятельности для повышения эффективности управления проектами и ресурсами.

ГЛАВА 1

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ПЛАНИРОВАНИЯ

1.1 Исследование инструментов для управления задачами и рабочим временем

Методы планирования представляют собой графические модели проектов, в которых работы и события проектных задач изображаются в виде графов. С помощью сетевой модели можно определить общую продолжительность проекта, рассчитать время выполнения отдельных задач и выявить критические пути – последовательности задач, не имеющих резерва времени.

Современные инструменты планирования проектов представляют собой специализированные программные решения и методологии, предназначенные для организации, контроля и оптимизации выполнения проектных задач. Они предоставляют комплекс функциональных возможностей для визуализации работ, управления ресурсами и временными параметрами (графики Ганта, сетевые диаграммы, PERT-анализ), а также для автоматизации расчетов критических показателей проекта.

Профессиональные системы планирования проектов основаны на мощных вычислительных алгоритмах и современных технологиях визуализации данных. Они обеспечивают:

- автоматизированное построение временных графиков
- расчет критического пути
- анализ ресурсной загрузки
- контроль выполнения задач

Среди наиболее распространенных решений можно выделить Microsoft Project, Primavera P6, JIRA, Asana, Trello и другие. Каждая из этих систем предлагает уникальный набор функций, интерфейсов и подходов к управлению проектами, что позволяет подобрать оптимальное решение для различных типов проектов и организаций.

В данной работе особое внимание уделено диаграмме Ганта как наиболее наглядному и востребованному инструменту визуализации проектных планов. Ее цифровая реализация требует тщательного анализа существующих решений, что позволит разработать оптимальное веб-приложение с учетом современных требований к функциональности.

Чтобы подтвердить практическую релевантность теоретических методов, рассмотрим современные программные решения, реализующие основные принципы сетевого планирования, их ключевые особенности и обоснуем выбор технологической платформы для реализации разрабатываемого решения.

1.2 Обзор классических методов планирования

Временное планирование представляет собой ключевой элемент управления проектами, обеспечивающий согласованность действий и контроль над сроками реализации. С формированием формализованных подходов к управлению сложными проектами возникла потребность в строгих методах координации работ, что привело к появлению классических методов сетевого планирования — СРМ, PERT, МРМ и традиционных сетевых графиков.

Указанные методы позволяют не только формализовать логические зависимости между задачами, но и осуществлять расчёт временных параметров — раннего и позднего начала, длительности, резервов и критических путей. Это обеспечивает выявление рисков нарушения сроков и повышение управляемости проекта. Методы охватывают как детерминированные, так и стохастические модели, что особенно актуально при высокой степени неопределённости.

Каждый из подходов имеет свои преимущества: СРМ — для стабильных проектов с фиксированной структурой, PERT — при неопределённых сроках, МРМ — для анализа остаточного времени. При этом все методы основаны на теории графов и математическом моделировании, что делает их применимыми в автоматизированных системах управления.

1.2.1 Метод сетевого графика: визуализация и анализ зависимостей

Сетевой график служит универсальной моделью для представления структуры проекта, наглядно отображая взаимосвязи между задачами. Этот метод лежит в основе как СРМ, так и PERT подходов, обеспечивая визуализацию сложных проектных зависимостей. В графической модели узлы обозначают ключевые события проекта, а соединяющие их дуги — работы и их последовательность. Особенно полезен такой подход для проектов с параллельными процессами и сложными перекрестными связями между различными этапами работ [11].

В структуре сетевого графа:

- Узлы (вершины) символизируют ключевые события или вехи проекта;
- Дуги (связи между узлами) представляют собой работы или задачи проекта;
- Направление дуг отражает логическую последовательность выполнения работ.

Основное преимущество сетевых графиков заключается в их способности четко демонстрировать логику выполнения проекта. Они позволяют не

только увидеть общую картину, но и провести детальный анализ, включая определение критического пути и расчет временных резервов. Наглядность представления помогает быстро выявлять потенциальные проблемы и находить оптимальные решения.

Однако при работе с крупными проектами сетевые графики сталкиваются с естественными ограничениями. Когда количество задач превышает несколько сотен, графическое представление становится перегруженным и теряет свою наглядность. Чрезмерное количество связей и узлов превращает график в сложную для восприятия паутину, что затрудняет его практическое использование. Именно поэтому в современных условиях сетевые графики чаще всего применяются в сочетании со специализированным программным обеспечением, которое помогает преодолеть эти ограничения за счет автоматизированного построения, масштабирования и интерактивной работы с элементами графа.

1.2.2 Метод критического пути(CPM): принципы и математическая модель

Метод критического пути (Critical Path Method, CPM) используется для определения минимального времени выполнения проекта и определения критических задач, от которых зависит срок завершения. Эти задачи нельзя отложить без задержки всего проекта, что делает их приоритетными для контроля [11].

Главной аналитической составляющей метода является определение критического пути — наиболее продолжительной последовательности взаимозависимых задач, где любая задержка выполнения работ автоматически приводит к увеличению общего срока реализации всего проекта. Это делает критический путь важнейшим объектом контроля при управлении проектами.

С математической точки зрения, CPM базируется на расчете двух групп параметров:

1. Ранние сроки (ES - Early Start) и поздние сроки (LS - Late Start) начала выполнения задач

2. Величины временных резервов ($Slack = LS - ES$)

Особое значение имеют задачи с нулевым резервом времени, которые и формируют критический путь проекта. Вычислительный процесс включает два последовательных этапа:

1. Прямой проход по графу (от начального к конечному событию) для определения ранних сроков

2. Обратный проход (от конечного к начальному событию) для вычисления поздних сроков

Важной характеристикой СРМ является его детерминированная природа — метод оперирует фиксированными, заранее определенными оценками продолжительности работ, что обеспечивает точность расчетов, но требует достоверных исходных данных. Эта особенность делает метод особенно эффективным для проектов со стабильными и предсказуемыми параметрами выполнения работ.

1.2.3 Метод оценки и пересмотра программ (PERT): вероятностная природа задач

Метод оценки и пересмотра программ (Program Evaluation and Review Technique, PERT) представляет собой продвинутый инструмент сетевого планирования, первоначально разработанный для управления сложными оборонными и аэрокосмическими проектами, где точные временные оценки выполнения работ зачастую отсутствовали. В отличие от детерминированного подхода СРМ, PERT специально создан для работы в условиях высокой неопределенности, что делает его особенно ценным для инновационных и исследовательских проектов.

Ключевой особенностью метода является использование трех временных оценок для каждой задачи [4]:

1. Оптимистичная оценка (t_0) — минимально возможная продолжительность при идеальных условиях;
2. Наиболее вероятная оценка (t_m) — реалистичный срок выполнения;
3. Пессимистичная оценка (t_p) — максимальная продолжительность при неблагоприятных обстоятельствах.

На основе этих оценок согласно Ларсону и Грею [4], оценка длительности в PERT использует взвешенное среднее трех значений (t_e), которое рассчитывается по формуле (1.1), где применяется взвешенное среднее с акцентом на наиболее вероятное значение:

$$t_e = \frac{t_0 + 4t_m + t_p}{6} \quad (1.1)$$

Данная формула основана на свойствах бета-распределения, которое эффективно моделирует асимметричные вероятностные распределения, характерные для реальных проектных ситуаций. Такой подход позволяет более точно отражать неопределенность, присущую многим видам работ.

PERT предоставляет дополнительные аналитические возможности:

- Расчет дисперсии для каждой задачи;

- Определение вероятности завершения проекта в заданные сроки с использованием теории вероятностей;

- Анализ чувствительности проекта к изменениям параметров задач.

Благодаря этим особенностям, метод PERT получил широкое распространение не только в оборонной промышленности, но и в других областях с высокой степенью неопределенности — научных исследованиях, IT-разработках, строительстве уникальных объектов. Его применение особенно оправдано на ранних стадиях проектов, когда точные оценки продолжительности работ отсутствуют, но необходимо построить реалистичный прогноз сроков реализации.

1.2.4 Метод потенциалов (MPM): обратный анализ графа

Метод потенциалов, известный также как метод Роя (Metra Potential Method), представляет собой усовершенствованную версию классического СРМ-подхода с акцентом на обратном анализе временных параметров проекта. В отличие от традиционного СРМ, где расчеты ведутся в двух направлениях (прямом и обратном), МРМ фокусируется на концепции "потенциалов событий" — величин, характеризующих максимальное время, которое может быть затрачено на выполнение оставшейся части проекта от данного события до его завершения [6].

Ключевые особенности метода:

- Обратный анализ — расчеты начинаются от конечного события проекта и движутся к начальному, что позволяет оценить временные резервы на каждом участке;

- Потенциал события — ключевой параметр, показывающий допустимую продолжительность оставшегося пути;

- Гибкость при изменениях — метод особенно эффективен при частых корректировках сроков, так как позволяет быстро пересчитывать параметры без полного повторного анализа всего проекта.

Преимущества МРМ перед классическим СРМ:

- Более удобен для анализа итерационных проектов с циклическими процессами;

- Позволяет быстрее адаптироваться к изменениям сроков выполнения отдельных задач;

- Дает наглядное представление о временных резервах на каждом участке проекта;

Алгоритм метода потенциалов:

1. Построить начальный допустимый план (например, методом северо-западного угла или минимального элемента).

2. Ввести потенциалы:

Для всех базисных ячеек (i,j) решить систему уравнений:

$$u_i + v_j = c_{ij} \quad (1.2)$$

3. Вычислить оценки (разности) для всех свободных ячеек:

$$\Delta_{ij} = c_{ij} - (u_i + v_j) \quad (1.3)$$

4. Проверка оптимальности:

Если все дельта больше или равны нулю, то текущий план оптимален — стоп

5. В противном случае:

1. Построить замкнутый цикл с чередующимися "+" и "-".
2. Найти минимальное значение среди "-" ячеек.
3. Перераспределить поставки вдоль цикла.

6. Повторить шаги 2–5, пока план не станет оптимальным.

Математическая основа метода позволяет не только определять критические пути, но и оценивать степень влияния изменений в отдельных задачах на общую продолжительность проекта. Это делает МРМ особенно ценным инструментом в условиях нестабильной среды, где требования и сроки часто подвергаются корректировкам.

1.2.5 Диаграмма Ганта

Диаграмма Ганта, несмотря на свою визуальную простоту, является мощным инструментом временного планирования, который эффективно дополняет классические методы сетевого анализа (СРМ, PERT, МРМ). В отличие от строго формализованных подходов, она предоставляет интуитивно понятное представление проекта, сочетая в себе наглядность, гибкость и аналитические возможности.

Например, если сравнивать со сложными сетевыми графиками (СРМ, PERT), которые требуют специальных знаний для интерпретации, диаграмма Ганта — это горизонтальная шкала времени, где каждая задача представлена прямоугольником. Длина прямоугольника соответствует длительности задачи, а положение на шкале — её срокам. Это делает диаграмму Ганта доступной не только для менеджеров проектов, но и для всех участников команды, включая нетехнических специалистов [5].

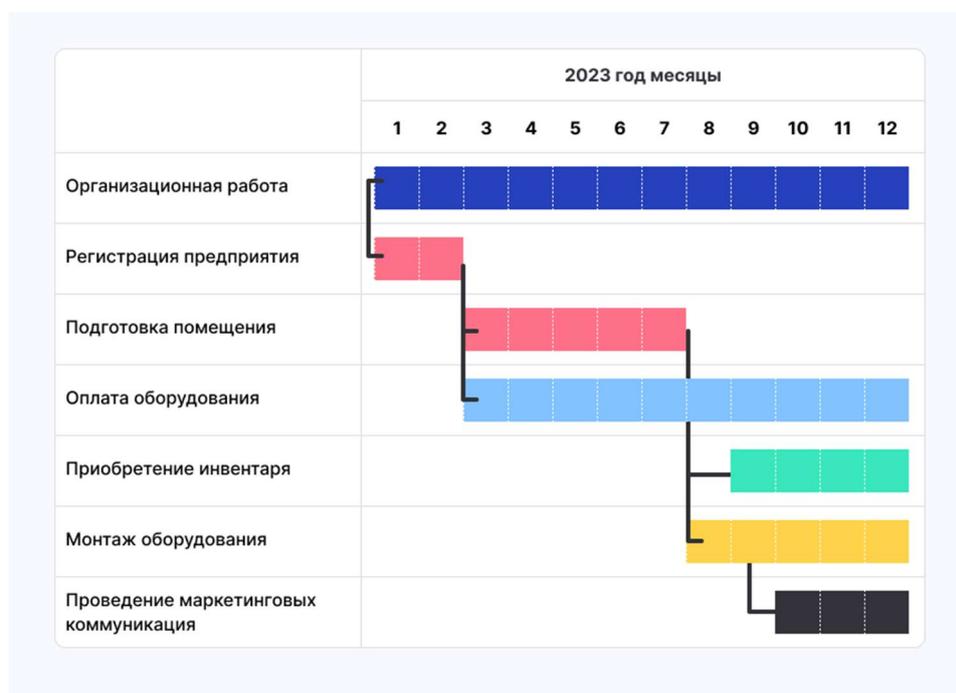


Рисунок 1.1 – Пример диаграммы Ганта

Модель диаграммы Ганта поддерживает динамическое изменение сроков (перетаскивание задач, корректировка длительности), различные виды зависимостей (FS, SS, FF, SF), аналогично СРМ, многопроектное управление — возможность отображения нескольких проектов на одной временной шкале. В отличие от статичных сетевых графиков, диаграмма Ганта позволяет оперативно вносить изменения и сразу видеть их влияние на общий срок проекта.

1.3 Математические основы сетевого планирования

Методы сетевого планирования опираются на строгий математический аппарат, включающий элементы теории графов, дискретной математики и вероятностного анализа. Такие подходы позволяют формализовать структуру проекта в виде ориентированного ациклического графа, на основе которого осуществляется вычисление ключевых временных характеристик: раннего и позднего начала, длительности задач, резервов времени и критических путей.

Ключевое значение имеет определение критического пути — последовательности задач с нулевым резервом времени, определяющей минимальную продолжительность проекта. Расчёты основаны на алгоритмах прямого и обратного прохода по графу, позволяющих оценить временные риски и обеспечить управляемость сроками.

Таким образом, математические методы обеспечивают точность и воспроизводимость планирования, а также создают основу для прогнозной аналитики в управлении проектами.

1.3.1 Использование ориентированных ациклических графов (DAG)

Основой всех сетевых методов является представление проекта в виде ориентированного ациклического графа. В нём вершины (события) и дуги (работы) упорядочены так, чтобы отсутствовали циклы. Такой граф можно топологически отсортировать, что позволяет проводить вычисления с начала и с конца, определяя ранние и поздние сроки событий.

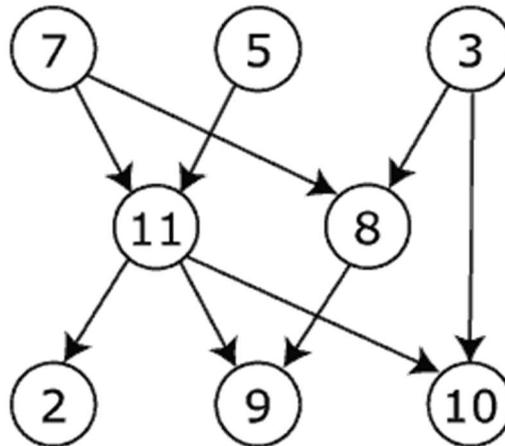


Рисунок 1.2 – Пример ациклического графа

В виде таких графов представляются сетевые модели, где каждый узел заменяет задачу, а ребра показывают порядок выполнения. Такая модель позволяет визуализировать данные и управлять сложными рабочими процессами, гарантируя обработку данных в верной последовательности.

1.3.2 Расчет ранних и поздних сроков, резервов времени

Для каждой задачи в проекте рассчитываются:

- ранний старт (ES) – самый ранний момент, когда операция может начаться, учитывая завершенность предыдущих,
- поздний старт (LS) – максимальное время начала работы без нарушения сроков завершения проекта,
- раннее окончание (EF) – момент завершения операции, если она начата в максимально раннее возможное время,
- позднее окончание (LF) – самое позднее время окончания работы, не вызывающее срыва общего срока проекта,
- общий резерв – максимальное время, на которое выполнение операции может быть отложено без нарушения сроков завершения проекта,

- свободный резерв – максимальное время, на которое операция может быть отложена без влияния на ранний старт последующих операций.

Эти параметры позволяют определить гибкость в исполнении задач и выявить критические элементы графа, которые не могут быть задержаны без сдвига всего проекта.

1.3.3 Алгоритм определения критического пути

Определение критического пути проводится в два этапа:

1. Прямой проход: расчёт ранних сроков начала и окончания задач.
2. Обратный проход: расчёт поздних сроков и резервов.

При этом применяются формулы:

$$ES_j = \max(ES_i + D_{ij}) \quad (1.4)$$

$$LS_i = \min(LS_j - D_{ij}) \quad (1.5)$$

где D_{ij} — длительность работы от события i к j . Путь, по которому резерв равен нулю, считается критическим [11].

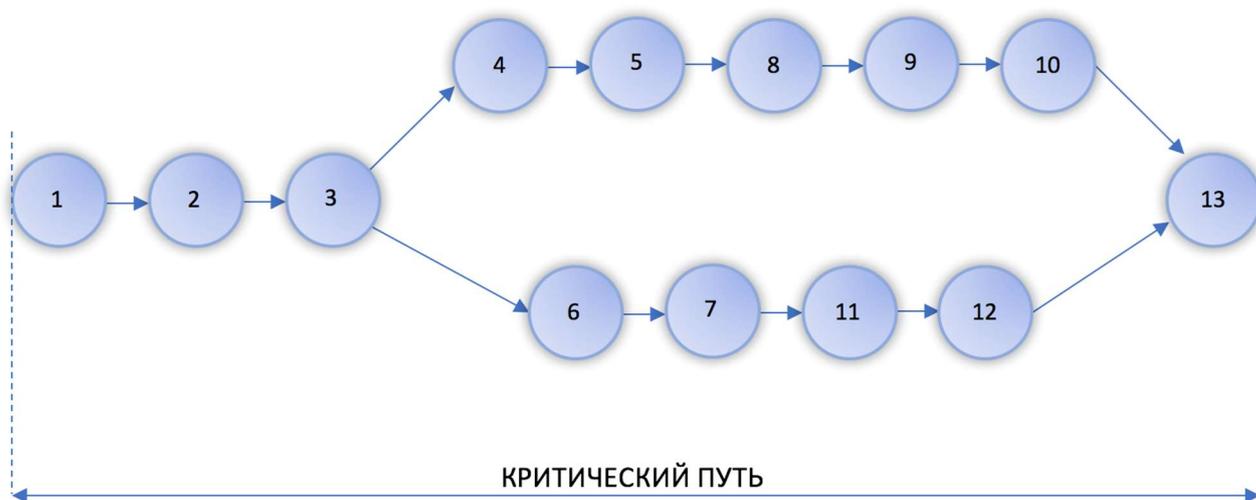


Рисунок 1.3 – Пример критического пути

1.3.4 Вероятностная модель PERT: бета-распределение, дисперсия

В PERT предполагается, что длительности задач имеют бета-распределение. Его асимметричная форма отражает неопределенность в оценке сроков. Среднее значение и дисперсия используются для расчёта ожидаемой продолжительности проекта и оценки риска превышения сроков [5].

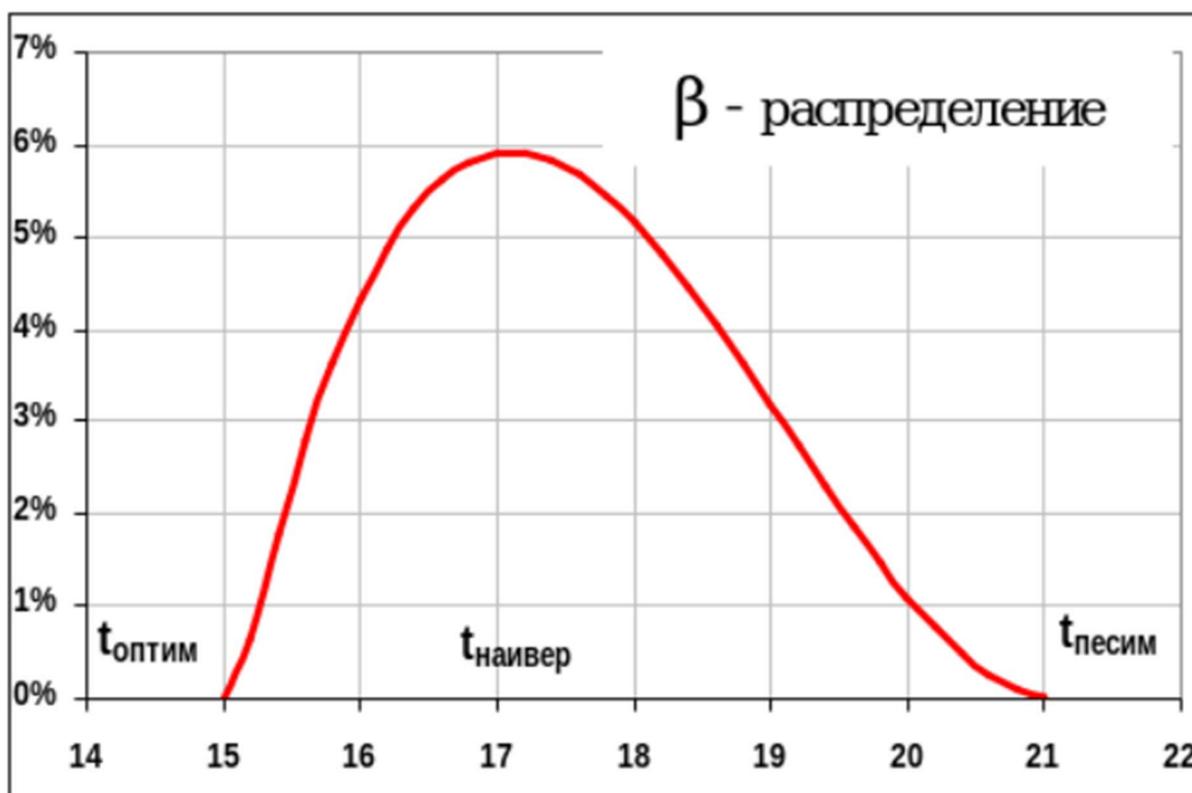


Рисунок 1.4 – Бета-распределение

Бета-распределение определяется бета-функцией, которая вычисляется по формуле (1.6):

$$B(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx \quad (1.6)$$

Где $\alpha, \beta > 0$ произвольные фиксированные параметры.

Формула для вычисления среднего значения была упомянута выше (Формула 1.1). Оно представляет собой ни что иное, как средневзвешенное значение, где, наиболее вероятная оценка длительности имеет вес 4 раза больший, чем оптимистична и пессимистичная оценки.

Для более глубокого анализа продолжительности задач рассчитывают значение дисперсии (формула 1.7), она говорит об уровне разброса оптимистичного, пессимистичного и наиболее вероятного значений от их среднего.

$$\sigma^2 = \left(\frac{t_0 - t_p}{6} \right)^2 \quad (1.7)$$

1.4 Сравнение методов по ключевым критериям

Несмотря на общую концептуальную основу, классические методы сетевого планирования существенно различаются по своим характеристикам, функциональным возможностям и области применения. Выбор конкретного метода зависит от множества факторов: степени определённости сроков выполнения задач, сложности проекта, требований к визуализации, необходимости учёта рисков, а также доступности вычислительных ресурсов и программных средств.

Сравнение целесообразно проводить по таким параметрам, как уровень детализации расчетов, адаптивность к изменениям проекта, удобство визуализации и поддержка автоматизированных инструментов. Некоторые методы более чувствительны к исходным данным, другие — лучше масштабируются при увеличении числа задач.

В данной части будет проведен сравнительный анализ рассмотренных ранее методов по основным критериям, определяющим их практическую применимость в реальных условиях.

1.4.1 Уровень детализации и точность

Метод критического пути (СРМ) и метод потенциалов (МРМ) обеспечивают высокую точность планирования при работе с фиксированными параметрами задач. Эти методы идеально подходят для проектов со стабильными условиями, где все временные оценки могут быть определены заранее с высокой точностью.

Метод оценки и пересмотра программ (PERT) учитывает вероятностную природу длительностей задач, используя три временные оценки: оптимистичную, пессимистичную и наиболее вероятную. Такой подход позволяет рассчитать ожидаемую длительность и дисперсию для каждой задачи, что делает PERT более точным в условиях неопределенности.

Сетевые графики, применяемые в СРМ и PERT, обеспечивают полную визуализацию всех логических зависимостей между задачами, что делает их незаменимыми для анализа сложных проектных структур. Однако при большом количестве задач такие графики могут становиться громоздкими и трудными для восприятия.

Диаграмма Ганта, хотя и уступает в точности отображения сложных зависимостей, превосходит сетевые графики в наглядности представления временных параметров и удобстве контроля сроков выполнения задач.

1.4.2 Гибкость и адаптивность к изменениям

PERT демонстрирует высокую гибкость при работе в условиях неопределенности, позволяя корректировать временные оценки по мере поступления новых данных. Это делает его особенно ценным для проектов с высокой степенью неопределенности сроков выполнения задач.

CPM и MPM менее адаптивны к изменениям, так как предполагают фиксированную структуру работ. Однако они остаются оптимальным выбором для проектов со стабильными параметрами, где важна точность расчетов и контроль за критическим путем. MPM особенно удобен для анализа временных резервов и оценки влияния изменений сроков на общую продолжительность проекта.

Диаграмма Ганта обеспечивает высокую гибкость в оперативном управлении сроками благодаря возможности визуального перетаскивания задач и изменения их длительности. Это делает ее удобным инструментом для проектов, где планы часто требуют корректировки.

1.4.3 Учет рисков и неопределенности

PERT является лидером в области учета рисков благодаря своей вероятностной природе. Метод позволяет оценивать вероятность завершения проекта в заданные сроки, используя дисперсию длительностей задач. Это особенно важно для проектов с высокой неопределенностью временных параметров.

CPM и MPM не предназначены для работы с вероятностными параметрами, что ограничивает их применение в условиях неопределенности. Однако они остаются эффективными для проектов с четко определенными параметрами, где основное внимание уделяется оптимизации сроков и ресурсов.

Диаграмма Ганта может быть дополнена элементами риск-менеджмента, например, с помощью цветового кодирования для выделения задач с высоким уровнем риска или отметки критических участков проекта.

1.4.4 Поддержка автоматизации и визуализации

Современные инструменты управления проектами обеспечивают автоматическую визуализацию как диаграмм Ганта, так и сетевых графиков. Это значительно упрощает процесс планирования и контроля, особенно для крупных проектов.

Диаграмма Ганта остается наиболее удобным инструментом для визуализации временных параметров. Ее интерактивные реализации позволяют:

- оперативно корректировать сроки выполнения задач
- назначать и перераспределять ресурсы
- отслеживать текущий прогресс проекта

Кроме того, диаграмма Ганта легко интегрируется с популярными системами управления проектами, что делает ее универсальным решением для командной работы.

Сетевые графики, используемые в СРМ и PERT, также поддерживаются многими профессиональными программными продуктами для управления проектами, обеспечивая детальный анализ зависимостей между задачами.

1.5 Современные программные инструменты планирования

Современные системы управления проектами интегрируют классические методы (СРМ, диаграммы Ганта, анализ критического пути) с визуализацией, управлением ресурсами и коммуникациями.

Главным достоинством таких инструментов является удобство интерфейса и возможность коллективной работы, позволяющая оперативно отслеживать структуру проекта, взаимосвязи задач, степень их выполнения и влияние на сроки. Расчёты выполняются автоматически, а любые изменения мгновенно отражаются в планах, что способствует адаптивности и прозрачности управления ресурсами и сроками.

К числу ведущих решений относятся Microsoft Project — корпоративный стандарт, облачный сервис GanttPRO и гибридная платформа ClickUp. Все они реализуют ключевые принципы сетевого планирования, включая расчёт критического пути, построение диаграмм Ганта и оценку загрузки ресурсов.

В последующих разделах представлен обзор и сравнительный анализ этих инструментов в контексте теоретических моделей, рассмотренных ранее.

1.5.1 Microsoft Project

MS Project — система управления проектами, реализующая СРМ, диаграмму Ганта и расширенные аналитические функции. Позволяет управлять ресурсами, бюджетами, сроками и критическим путём [12].

Name	Assigned to	Duration	Depends on ...	Effort remaining	Start	Fi
1 Determine mentor categories		35 days		192 hours	8/9/2019	9,
2 Validate potential job classifications	Elva Hebert	4 days		32 hours	9/9/2019	9,
3 Determine mentor qualifications	JR	5 days	2		8/9/2019	8,
4 Define mentor roles and responsibilities	NT JR	10 days	2 3	160 hours	9/13/2019	9,
5 Mentor program defined		0 days	4		9/26/2019	9,
6 Identify Mentors		25 days		272 hours	9/9/2019	11
7 Request mentor volunteers	Felipe Lawhorr	10 days	2 3	80 hours	9/13/2019	9,
8 Screen mentors	RG GG	10 days	7	160 hours	9/27/2019	11
9 Accept mentors	NT	2 days	8 5	32 hours	10/11/2019	11
10 Mentors-identified		0 days	9		9/9/2019	9,

Рисунок 1.5 – Пример проекта в MS Project

Он остается лидером в корпоративном сегменте. Его ключевые особенности:

- Полная реализация методологии СРМ с расчетом:
 - Ранних/поздних сроков начала и окончания задач;
 - Полных и свободных резервов времени;
 - Критического пути и его динамического пересчета.
- Расширенные возможности:
 - Многоуровневое планирование задач;
 - Анализ "что-если" (сценарное моделирование).

К ограничениям можно отнести высокую стоимость лицензий, сложность освоения для начинающих и ориентацию на классические методы управления.

1.5.2 GanttPRO

GanttPRO представляет собой облачное решение, позволяющее строить диаграммы Ганта, задавать зависимости, отслеживать прогресс [8].

Преимущества:

- Интуитивно понятный интерфейс;
- Реализация базовых принципов СРМ;
- Возможность командной работы в реальном времени;
- Доступность с любых устройств.

Особенности:

- Упрощенный расчет критического пути;
- Ограниченные возможности ресурсного планирования;
- Готовые шаблоны для различных отраслей.

Он идеально подходит для малых и средних проектов или команд, работающих по гибридным методологиям, так как основывается на визуальном управлении без сложных расчетов.

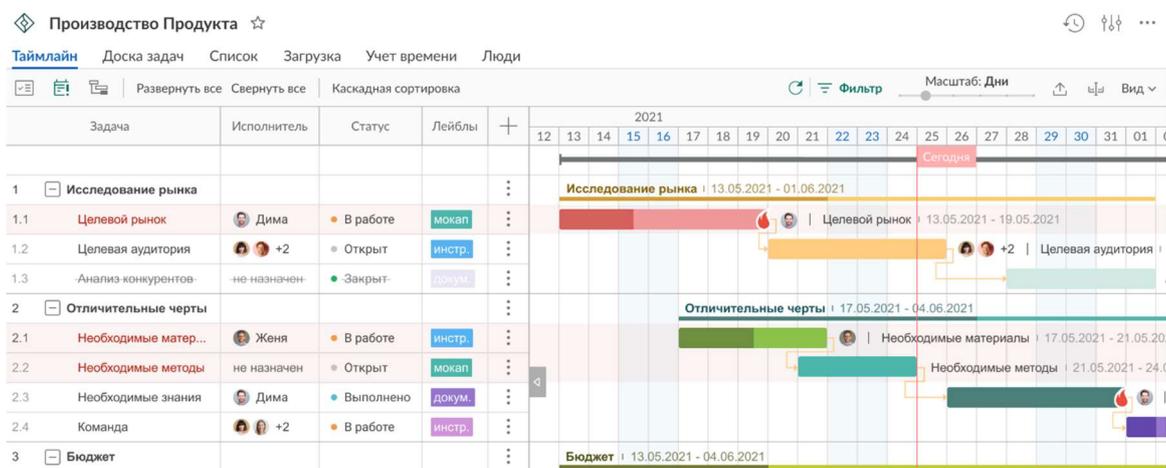


Рисунок 1.6 – Пример проекта в GanttPRO

1.5.3 ClickUp

ClickUp представляет собой универсальный облачный сервис, который сочетает подходы классического и гибкого управления проектами. Поддерживает задачи, диаграммы, доски Kanban и критический путь [7].

Уникальные возможности:

- Гибкое представление данных (доски, списки, диаграммы Ганта);
- Встроенные инструменты совместной работы;
- Автоматизация рутинных операций.

Стоит отметить особенности реализации СРМ, используется упрощенный расчет зависимостей, визуальное выделение критического пути и интеграция с Agile-инструментами, такими как Trello, Asana и т.д.

Преимущества:

- Легкость в освоении;
- Гибкость настройки рабочих пространств;
- Богатые возможности интеграции.

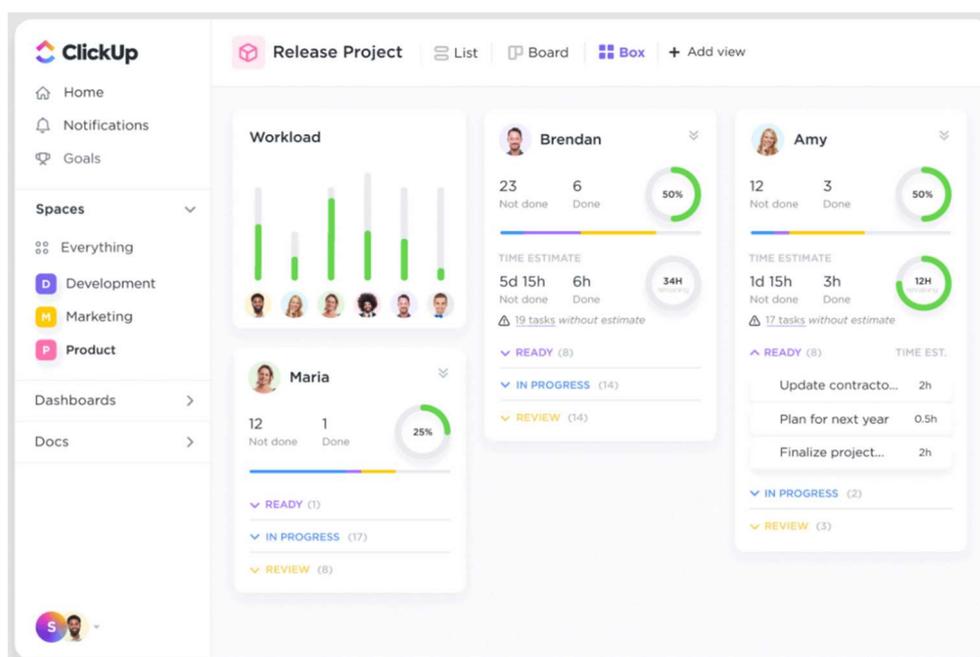


Рисунок 1.7 – Пример проекта в ClickUp

1.5.4 Общая сравнительная характеристика

Для наглядного сравнения функциональных возможностей ведущих систем управления проектами сведем ключевые параметры в таблицу 1.1:

Таблица 1.1 – Сравнение приложений планирования

Критерий	MS Project	GanttPRO	ClickUp
Полная реализация СРМ	Да	Частично	Частично
Расчет критического пути	Полный	Базовый	Упрощенный
Управление ресурсами	Многоуровневое	Базовое	Базовое
Визуализация данных	Базовая	Средняя	Высокая
Сценарное моделирование	Да	Нет	Ограниченно
Стоимость	Высокая	Средняя	Низкая
Сложность обучения	Высокая	Низкая	Средняя
Интеграционные возможности	Средние	Средние	Высокие

Проведенное сравнение трех популярных систем управления проектами выявило их ключевые различия в функциональности. Microsoft Project остается наиболее мощным профессиональным решением с полной поддержкой СРМ и многоуровневым управлением ресурсами, однако требует значительных финансовых вложений и времени на освоение. GanttPRO предлагает оптимальный баланс между функциональностью и удобством использования, делая акцент на визуализации данных. ClickUp выделяется своей доступностью и гибкостью, хотя и имеет ограничения в аналитических возможностях [3].

Выбор конкретного инструмента должен основываться на потребностях проекта: для сложных корпоративных решений предпочтительнее MS Project, для средних по размерам команд и необъемным проектам – GanttPRO, а для небольших команд и маленьких проектов – ClickUp. Все три системы демонстрируют разные подходы к реализации методов сетевого планирования, что подтверждает важность тщательного анализа требований перед выбором программного обеспечения для управления проектами [13].

1.6 Обоснование выбора метода для реализации

В рамках дипломной работы основной задачей является визуализация загрузки сотрудников и контроль статуса задач, что делает диаграмму Ганта предпочтительным инструментом для решения поставленной задачи в рамках данной работы.

Однако простая визуализация недостаточна без логической структуры и временных взаимосвязей между задачами. Поэтому дополнительно предполагается интеграция элементов сетевого анализа, прежде всего расчёта критического пути (СРМ) и логических зависимостей между задачами. Это позволяет обеспечить реалистичное и адаптивное планирование, отражающее как статическое, так и динамическое состояние проекта.

Выбранный подход сочетает преимущества диаграммы Ганта с точностью сетевых алгоритмов, что обеспечивает высокую информативность и удобство в практическом использовании.

1.7 Выводы по первой главе

Проведенное исследование методов сетевого планирования демонстрирует изменение подходов к управлению проектами – от классических алгоритмов СРМ и PERT до современных интерактивных решений на основе диаграмм Ганта. Анализ выявил, что каждый метод обладает уникальными характеристиками, определяющими его область эффективного применения.

Метод критического пути (СРМ) сохраняет свою актуальность для проектов с четко определенными параметрами, обеспечивая математическую точность расчетов временных показателей. Его алгоритм, основанный на теории графов, позволяет точно определять критические пути и временные резервы, что особенно важно для сложных проектов. В то же время вероятностный метод PERT остается незаменимым инструментом для работы при возможных изменениях, когда требуется учет возможных вариаций сроков выполнения задач.

Особое внимание в исследовании уделено диаграмме Ганта как наиболее востребованному инструменту визуального представления проектных планов. Ее ключевое преимущество заключается в сочетании аналитических возможностей (поддержка расчета критического пути, временных зависимостей) с визуализацией и простотой восприятия. Современные цифровые реализации диаграмм Ганта преодолели традиционные ограничения этого метода, добавив возможности динамического редактирования, автоматического пересчета параметров и интеграции с другими системами управления.

Сравнительный анализ профессиональных решений (MS Project, GanttPRO, ClickUp) показал, что современные инструменты успешно комбинируют различные методы планирования, адаптируя их к потребностям разных категорий пользователей. При этом наблюдается четкая тенденция к упрощению интерфейсов и повышению интерактивности без потери аналитических возможностей.

На основе проведенного анализа можно заключить, что диаграмма Ганта с интеграцией элементов сетевого анализа представляет собой оптимальный инструмент для визуального и аналитического сопровождения проекта. Такой подход позволяет объединить математическую строгость сетевых методов с удобством визуального представления, создавая эффективный инструмент для управления проектами различной сложности. Особую ценность представляет возможность адаптации классических алгоритмов к современным веб-технологиям, что открывает новые перспективы для развития инструментов проектного управления [10].

ГЛАВА 2

ПРОЕКТИРОВАНИЕ ИНСТРУМЕНТА ПЛАНИРОВАНИЯ

В данной главе рассматривается процесс проектирования веб-приложения, включая применение приведенных выше математических методов, анализ требований к инфраструктуре и модульной структуре системы. Особое внимание уделяется методам сетевого планирования, включая алгоритмы расчета критического пути и временных параметров работ. Практическая часть охватывает проектирование основных элементов интерфейса, интеграцию клиентской и серверной частей приложения, а также вопросы обеспечения производительности решения. В результате разрабатывается полнофункциональное веб-приложение, позволяющее эффективно управлять проектами с использованием современных технологий визуализации данных.

2.1 Практическое применение приведенных методов

В следующих пунктах рассмотрим используемые математические термины и методы планирования. Определим их полезность в рамках дипломной работы. Адаптируем их для будущего применения в рамках исполняемой компьютерной программы.

2.1.1 Определение критического пути

Метод поиска критического пути называется СРМ, относится к группе детерминированных методов сетевого планирования. Операции, принадлежащие критическому пути, называются критическими. Остальные операции имеют резерв времени, характеризуемый максимальной задержкой операции, при которой продолжительность проекта не изменяется. Критические операции имеют нулевой резерв.

Введем следующие обозначения:

- S_i – Самое раннее время наступления события i ;
- E_i – Самое позднее время наступления события i ;
- D_{ij} – Длительность процесса между узлами i и j .

Расчет пути состоит из двух этапов: проход по графу вперед, вычисляет самые ранние времена наступления событий, и проход назад, вычисляет самое позднее время наступления тех же событий. Проход вперед начинается на узле 1 и заканчивается на последнем узле n .

Рассмотрим последовательность действий для определения сроков раннего и позднего наступления событий:

1. установить самое раннее время наступления события для первого узла равным нулю;
2. для узла j определяем узлы $l, m \dots r$, непосредственно связанные с узлом j работами $[l;j], [m;j], [r;j]$;
3. рассчитать самое раннее наступление события j по формуле (2.1):

$$S_j = \max \{S_l + D_{lj}, S_m + D_{mj}, \dots, S_r + D_{rj}\} \quad (2.1)$$

4. повторять п.2 и п.3 до достижения узла n ;
5. проход в обратную сторону начинается в узле n и заканчивается в узле 1. На начальном шаге принимаем $E_n = S_n$, что указывает на совпадение самого раннего и позднего времени завершения проекта;
6. на каждом шаге для узла j определяются узлы $l, m \dots r$, непосредственно связанные с узлом j работами $[j;l], [j;m], [j;r]$;
7. рассчитать самое позднее время наступления события j по формуле (2.2):

$$E_j = \max \{E_l + D_{jl}, E_m + D_{jm}, \dots, E_r + D_{jr}\} \quad (2.2)$$

8. проход назад завершить при вычислении E_l для узла 1.
- Процесс $[i;j]$ можно назвать критическим, если будет выполнено три условия:
- $E_i = S_j$ – самое позднее время наступления события i совпадает с самым ранним временем наступлением события j ;
 - $S_i = E_j$ – самое раннее время наступление события i совпадает с самым поздним наступлением события j ;
 - $S_j - S_i = E_j - E_i = D_{ij}$ – разность между самыми ранними и самыми поздними временами наступления событий равны между собой и продолжительности работы $[i;j]$.

2.1.2 Определение параметров работ сетевого графика

Кроме ранних и поздних сроков свершения событий определяются ранние и поздние сроки начала и окончания работ. Смысл определения этих параметров тот же, что и у предыдущих, т.е. определяются возможности смещения при выполнении отдельных работ проекта.

Ранний срок начала работы. Совпадает с самым ранним сроком свершения предшествующего события:

$$S_{ij}' = S_i \quad (2.3)$$

Ранний срок окончания работы равен раннему сроку свершения предшествующего события плюс продолжительность этой работы.

$$FS_{ij} = S_i + D_{ij} \quad (2.4)$$

Поздний срок окончания работы совпадает с поздним сроком свершения последующего события.

$$E_{ij}' = E_j \quad (2.5)$$

Поздний срок начала работы равен позднему сроку свершения последующего события за вычетом продолжительности этой работы

$$FE_{ij} = E_j - D_{ij} \quad (2.6)$$

Общий запас времени операции – показывает, как можно увеличить время операции при условии, что срок выполнения всего проекта не изменится.

$$TF_{ij} = E_j - S_i - D_{ij} \quad (2.7)$$

Свободный резерв времени операции – показывает часть полного резерва времени, на которое можно увеличить продолжительность работы, не изменив при этом раннего срока ее конченого события.

$$FF_{ij} = S_j - S_i - D_{ij} \quad (2.8)$$

Свободный резерв должен быть меньше общего запаса времени.

Для некритических процессов можно выделить следующие правила:

- если $FF_{ij} = TF_{ij}$, тогда процесс $[i;j]$ может выполняться в любое время внутри максимального интервала (S_i, E_j) без нарушения отношений следования;
- если $FF_{ij} < TF_{ij}$, тогда без нарушения отношения следования данный процесс может начаться со сдвигом, не превышающим FF_{ij} , относительно самого раннего момента начала процесса i . Сдвиг начала процесса на величину времени, превышающую FF_{ij} (но не более TF_{ij}), должен сопровождаться равным сдвигом относительно j всех процессов, начинающихся с события j .

2.2 Требования к приложению

К требованиям следует отнести следующие пункты.

1. **Открытость:** Система должна допускать замену и обновление компонентов без значительных изменений.
2. **Модульность:** Возможность изменения отдельных частей приложения, таких как интерфейс, серверная логика и база данных, без нарушения общей функциональности.
3. **Безопасность:** Все данные о задачах и проектной информации должны быть защищены, а доступ к ним возможен только через авторизованные каналы.
4. **Унифицируемость:** Для упрощения работы с приложением используется общий механизм для управления задачами, прогрессом и зависимостями, что способствует эффективному обновлению и поддержке.
5. **Производительность:** Система должна быть способна обрабатывать значительное количество запросов для отображения и редактирования диаграмм Ганта, не перегружая сервер.
6. **Масштабируемость:** Приложение должно быть легко масштабируемо в случае увеличения числа пользователей или задач.
7. **Целостность данных:** В случае сбоя системы или сервера приложение должно гарантировать сохранение данных без потерь.

2.3 Проектирование структуры проекта

Этапы проектирования:

1. **Сбор требований:** Определение ключевых функциональных возможностей системы, таких как расчёт сроков выполнения задач, визуализация взаимосвязей и интеграция с другими инструментами.
2. **Разработка модели данных:** Определение сущностей, таких как задачи, зависимости, сроки начала и завершения, а также атрибутов для каждой из сущностей.
3. **Создание прототипов:** Разработка схем и макетов интерфейса для визуализации данных диаграммы Ганта.

База данных – SQLite.

```
CREATE TABLE task (  
    id INTEGER NOT NULL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    start VARCHAR(10) NOT NULL,  
    "end" VARCHAR(10) NOT NULL,  
    progress INTEGER NOT NULL,  
    dependencies VARCHAR(100)  
);
```

Рисунок 2.1 – Создание базы данных

Установим формат ответа backend на GET запрос от frontend приложения. В будущем приложении будем использовать следующую структуру. Возвращается массив задач со следующими полями:

- Зависимости – содержит id зависимых задач;
- Дата начала;
- Дата окончания;
- Уникальный идентификатор;
- Название и краткое описание;
- Прогресс (в процентах).

Установим схему работы самого веб-приложения.

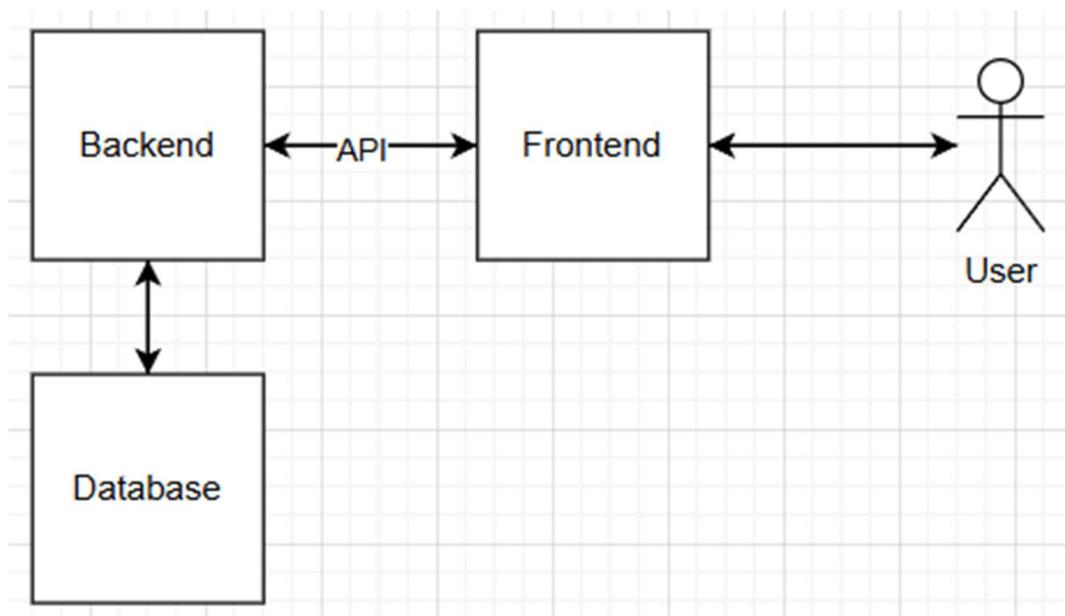


Рисунок 2.2 – Схема работы приложения

2.4 Выводы по второй главе

В данной главе был детально рассмотрен процесс проектирования инструмента планирования на основе методов сетевого анализа. Основное внимание уделено практической реализации алгоритмов расчета критического пути (СРМ) и временных параметров работ, что составляет математическую основу разрабатываемого приложения. Были формализованы ключевые формулы для определения ранних и поздних сроков выполнения задач, а также расчета временных резервов, что обеспечивает точность планирования и выявление критически важных операций.

Особое значение имела разработка требований, где акцент сделан на такие важные аспекты как модульность, безопасность данных и масштабируемость системы. Это позволило создать гибкую основу для дальнейшей реализации приложения.

В процессе проектирования была разработана четкая структура базы данных и API, определены основные сущности и их взаимосвязи. Представленные схемы работы системы и примеры взаимодействия компонентов наглядно демонстрируют продуманность архитектурных решений.

Результатом проектирования стала комплексная модель веб-приложения, сочетающая математическую строгость методов сетевого планирования с современными подходами к разработке программного обеспечения. Это создает прочную основу для практической реализации инструмента в следующей главе работы.

ГЛАВА 3

РЕАЛИЗАЦИЯ ИНСТРУМЕНТА ПЛАНИРОВАНИЯ

3.1 Основные элементы веб-приложения

При открытии главной страницы веб-приложения пользователь видит диаграмму, построенную на основе его данных, поле добавления новых задач и кнопку смены темы (светлая/тёмная).

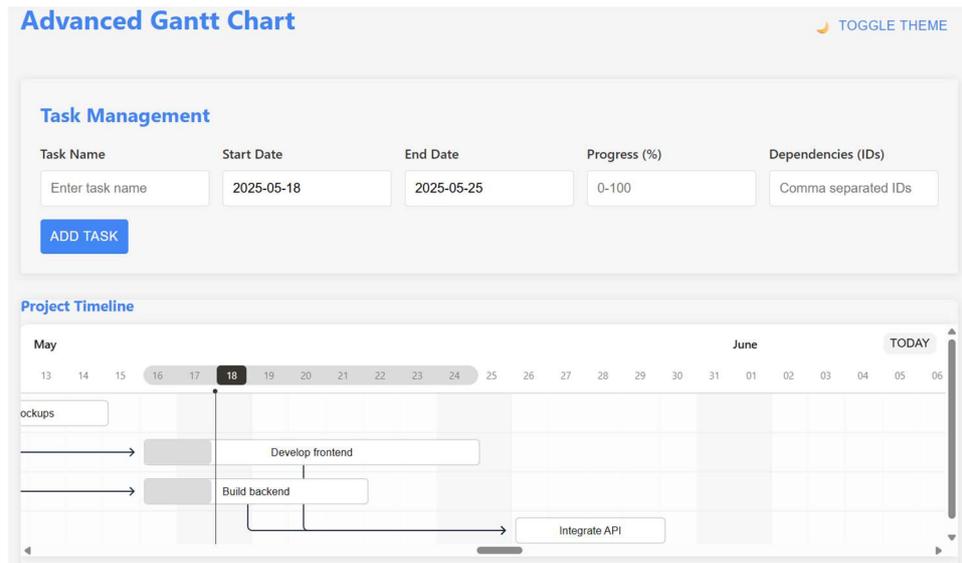


Рисунок 3.1 – Вид главной страницы приложения (светлая тема)

Сведения о задачах, которые пользователь может получить:

1. Текущие задания и их прогресс;
2. Дата начала и окончания каждого из заданий;
3. Текущая дата;
4. Зависимости между заданиями.

Теперь вкратце рассмотрим, как отображается каждый из пунктов.

Задание отображаются в виде прямоугольного блока, на котором прописано название. Также каждый блок имеет две части: готовая часть и та, что еще в прогрессе. При наведении на блок задания, выделяется период, на который оно рассчитано, и высвечивается вся доступная информация, как показано на рисунке 3.2:

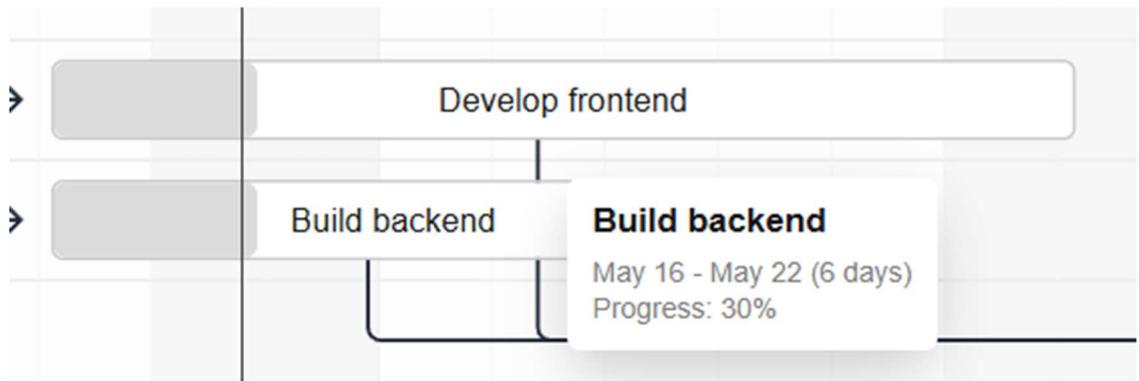


Рисунок 3.2 – Информация о задании.

Также на рисунке 3.1 серой полоской отмечена текущая дата и зависимости между заданиями.

Рассмотрим действия, которые доступны пользователю:

1. Изменение продолжительности задания;
2. Изменение вида отображения;
3. Перемещение на текущую дату.

Для изменения длительности задачи пользователь может «растянуть» её границы — это активирует интерактивные маркеры, аналогичные тем, что используются в графических редакторах (Рисунок 3.3).

Кроме того, можно перемещать задание на поле при нажатии и перетягивании. При таких изменениях сроки задания изменяются.

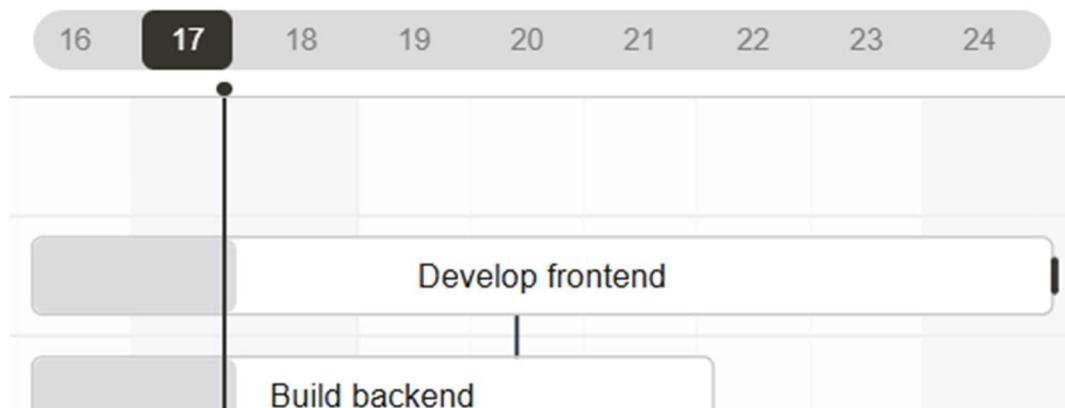


Рисунок 3.3 – Изменение продолжительности задания.

Нажатие на кнопку “Mode”, даёт возможность выбрать наиболее удобный вид отображения диаграммы. Доступно 7 видов отображения:

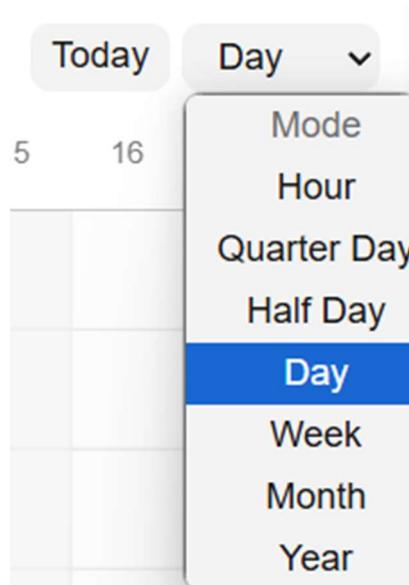


Рисунок 3.4 – Доступные виды отображения.

Нажатие на кнопку “Today”, перемещает на текущую дату.

3.2 Используемые технологии и архитектура веб-приложения

Разрабатываемое веб-приложение строится на основе трёхуровневой архитектуры, включающей:

1. Клиентскую часть (frontend) — реализованную с использованием HTML, CSS и JavaScript;
2. Серверную часть (backend) — построенную на Python с использованием Flask как лёгкого веб-фреймворка;
3. Систему управления базами данных — с применением SQLite и ORM-библиотеки SQLAlchemy для взаимодействия между сервером и базой данных.

Серверная часть. Сервер реализует API для управления задачами, датами, зависимостями и прогрессом выполнения. Используются:

1. Flask — фреймворк для маршрутизации и логики [9];
2. SQLAlchemy — ORM для работы с базой данных;
3. SQLite — встраиваемая СУБД, обеспечивающая надёжность и транзакционность.

Клиентская часть. Для отображения диаграмм используется JavaScript. Реализованы возможности:

- отображения и редактирования задач в графическом интерфейсе;
- переключения между тёмной и светлой темой;
- визуализации связей между задачами;

- использования drag-and-drop и изменения длительности заданий.

3.3 Реализация элементов веб-приложения

Выбор JavaScript (JS) для работы с веб-приложением обусловлен несколькими факторами:

1. Клиентская сторона: JS является основным языком для разработки клиентской стороны веб-приложений. Он выполняется непосредственно в браузере пользователя, что позволяет создавать интерактивные и динамические пользовательские интерфейсы.

2. Большое сообщество и экосистема: JavaScript имеет одно из самых больших и активных сообществ разработчиков. Существует множество ресурсов, библиотек, фреймворков и инструментов, которые упрощают разработку веб-приложений.

3. Широкая поддержка и совместимость: JavaScript является стандартом веб-разработки и поддерживается всеми современными браузерами, что обеспечивает совместимость приложения с различными платформами и устройствами.

Были разработаны классы Arrow, Popup и Bar, расчёты в которых использовали указанные ранее алгоритмы. На основе этих классов впоследствии был создан класс Gantt, реализующий экземпляр диаграммы.



Рисунок 3.5 – Реализованные классы

3.4 Реализация функционала веб-приложения

Серверная часть приложения организована по принципам модульности и разделения функционала, что обеспечивает гибкость и масштабируемость. Рассмотрим структуру более подробно с примерами реализации.

Основным исполняемым файлом в приложении является файл “app.py”. Там импортируются модели, инициализируются основные компоненты и начинается выполнение приложения.

```
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///tasks.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

class Task(db.Model):
    __tablename__ = 'tasks'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    start = db.Column(db.String(10), nullable=False)
    end = db.Column(db.String(10), nullable=False)
    progress = db.Column(db.Integer, nullable=False)
    dependencies = db.Column(db.String(100), nullable=True)

with app.app_context():
    db.create_all()
    Task.query.delete()
    db.session.commit()
```

Рисунок 3.6 – Инициализация основных компонентов и базы данных

Реализация API для получения, добавления и обновления задач с обработкой ошибок приведена в приложении А.

Для управления пользовательским интерфейсом реализованы методы на JavaScript. Такими методами являются смена темы, выбор даты при выборе задачи, появление сообщений об ошибке и другое. Например, на рисунке 3.8 приведена реализация выбора темы и загрузки. При переключении темы с помощью функции toggleTheme происходит не только визуальное изменение, но и сохранение выбора пользователя в localStorage, что обеспечивает сохранение предпочтений между сессиями.

```

// Theme Management
function toggleTheme() {
  const isDark = body.getAttribute('data-theme') === 'dark';
  body.setAttribute('data-theme', isDark ? '' : 'dark');
  localStorage.setItem('theme', isDark ? 'light' : 'dark');
  themeIcon.textContent = isDark ? '🌙' : '☀️';
}

function loadTheme() {
  const savedTheme = localStorage.getItem('theme') || 'light';
  body.setAttribute('data-theme', savedTheme === 'dark' ? 'dark' : '');
  themeIcon.textContent = savedTheme === 'dark' ? '☀️' : '🌙';
}

```

Рисунок 3.7 – Реализация выбора темы

Особое внимание уделено аспектам доступности и продуманности интерфейса. Реализация предусматривает возможность легкого добавления новых атрибутов для изменения внешнего вида.

Код написан с учетом возможного масштабирования. Например, можно разработать систему уведомлений, добавив очередь сообщений или приоритизацию. Механизм тем легко дополняется поддержкой системных предпочтений операционной системы. Работа с датами может быть расширена дополнительной валидацией и интеграцией с календарем праздников.

В целом, эта реализация демонстрирует современный подход к разработке клиентской части веб-приложений, где особое внимание уделяется не только функциональности, но и аспектам производительности, доступности и удобства сопровождения кода. Каждый компонент системы может модифицироваться или заменяться независимо, что соответствует разработанным требованиям и упрощает процесс разработки и тестирования.

3.5 Тестирование приложения

Для проверки отдельных модулей и функций приложения была реализована система юнит-тестов с использованием Jest - популярного средства разработки для тестирования JavaScript-кода. Особое внимание уделялось тестированию:

- Парсинга и форматирования дат (модуль `date_utils`);
- Корректности расчетов временных параметров задач;
- Логике обработки зависимостей между задачами;
- Преобразования данных между клиентской и серверной частями.

```
test('Calculate task duration correctly', () => {
  const start = new Date('2025-05-10');
  const end = new Date('2025-05-15');
  const duration = date_utils.diff(end, start, 'day');
  expect(duration).toBe(5);
});
```

Рисунок 3.8 – Пример теста для проверки расчета временных параметров

Также проводилось комплексное тестирование взаимодействия между различными компонентами системы. Особое внимание уделялось интеграции клиентской части (Frontend) с серверным API, что обеспечивало корректный обмен данными между браузером и сервером. Тестировалась работа модуля диаграммы Ганта с хранилищем данных, проверялась целостность информации при различных операциях.

3.6 Выводы к третьей главе

В данной главе была реализована ключевая функциональность веб-приложения для работы с диаграммами Ганта. Основное внимание уделено разработке интуитивно понятного интерфейса с поддержкой темной/светлой темы и возможностью интерактивного редактирования задач через drag-and-drop.

Технологический стек (Flask, SQLite, JavaScript) был подобран в соответствии с поставленной в ходе работы задачей. По результатам реализации мы получили готовое решение, полностью согласованное с поставленными требованиями. Особое значение имеет реализация алгоритмов расчета временных параметров и тестирование функциональности, что гарантирует корректность работы приложения.

Разработанное решение успешно сочетает визуальную наглядность диаграмм Ганта с точностью расчетов сетевого планирования, предлагая пользователям удобный инструмент для управления проектами.

ЗАКЛЮЧЕНИЕ

В рамках данной дипломной работы было разработано веб-приложение для визуализации и управления проектами с использованием интерактивной диаграммы Ганта. Проведенная работа включала несколько ключевых этапов:

1. Анализ современных методов сетевого планирования (CPM, PERT, MPM) и обоснование выбора диаграммы Ганта как оптимального решения для наглядного представления временных параметров проекта.

2. Разработка полнофункционального веб-приложения, включающего:

- Серверную часть на Python с использованием средства разработки Flask;
- Клиентскую часть с интерактивной диаграммой;
- Базу данных SQLite с ORM SQLAlchemy.

Основные функциональные возможности реализованного решения включают создание и редактирование задач с указанием временных параметров, установку зависимостей между задачами, автоматическое построение и обновление диаграммы, визуальное выделение критического пути, а также цветовую индикацию прогресса выполнения.

Особое внимание было уделено интерфейсу приложения. Разработан интуитивно понятный внешний вид с возможностью drag-and-drop редактирования, реализована адаптивная верстка для разных устройств, добавлены визуальные подсказки для пользователя.

Практическая значимость работы заключается в создании функционального инструмента для проектного управления, который может использоваться как самостоятельное решение для малых и средних проектов, так и в качестве учебного пособия для изучения методов сетевого планирования. Модульная архитектура приложения позволяет рассматривать его как базовый компонент для дальнейшего развития и интеграции с другими системами.

Перспективы развития проекта включают несколько направлений:

- Добавление функций управления ресурсами и бюджетирования;
- Реализацию многопользовательского режима с разграничением прав доступа;
- Интеграцию с популярными API (Google Calendar, Trello, Jira);
- Внедрение элементов искусственного интеллекта для прогнозирования сроков выполнения задач.

Проведенная работа наглядно демонстрирует эффективность сочетания классических методов сетевого планирования с современными веб-технологиями. Разработанное приложение подтверждает, что даже сложные методы

управления проектами могут быть реализованы в удобной и доступной форме, сохраняя при этом свою аналитическую ценность и точность расчетов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Зуб А. Т. Управление проектами: учебник и практикум. — М.: Юрайт, 2019. — 390 с.
2. Козлов А.В. Сетевое планирование и управление проектами: учебное пособие. — СПб.: Питер, 2018. — 288 с.
3. Куперштейн В. И. Microsoft Project 2013 в управлении проектами. — СПб.: БХВ-Петербург, 2014. — 384 с.
4. Ларсон Э., Грей К. Управление проектами. — М.: Дело и Сервис, 2016. — 928 с.
5. Плескунов М. А. Задачи сетевого планирования: учебное пособие. — Екатеринбург: УрФУ, 2014. — 104 с.
6. Руденко Л. И. Основы управления проектами: практический курс. — Симферополь: Таврическая академия, 2018. — 144 с.
7. ClickUp. Веб-сервис управления проектами [Электронный ресурс]. — Режим доступа: <https://clickup.com/>. Date of access: 24.09.2024
8. GanttPRO. Облачное решение для построения диаграмм Ганта [Электронный ресурс]. — Режим доступа: <https://ganttpro.com/> . Date of access: 24.09.2024
9. Greenberg М. Разработка веб-приложений с использованием Flask на языке Python / пер. с англ. А. Н. Киселёвой. — М.: ДМК Пресс, 2014. — 320 с.
10. Heagney J. Fundamentals of Project Management. — 5th ed. — New York: AMACOM, 2016. — 256 p.
11. Kerzner Н. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. — 12th ed. — Hoboken, NJ: Wiley, 2017. — 1296 p.
12. Microsoft Project. Программное обеспечение для управления проектами [Электронный ресурс]. — Режим доступа: <https://www.microsoft.com/project/>. Date of access: 24.09.2024
13. Portny S. E., Portny J. L. Project Management For Dummies. — 6th ed. — Hoboken, NJ: Wiley, 2022. — 448 p.
14. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). — 7th ed. — Newtown Square, PA: Project Management Institute, 2021. — 370 p.

ЛИСТИНГ ОСНОВНОГО ФАЙЛА

```
from flask import Flask, request, jsonify, render_template
from flask_sqlalchemy import SQLAlchemy
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///tasks.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)
class Task(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    start = db.Column(db.String(10), nullable=False)
    end = db.Column(db.String(10), nullable=False)
    progress = db.Column(db.Integer, nullable=False)
    dependencies = db.Column(db.String(100), nullable=True)
with app.app_context():
    db.create_all()
    Task.query.delete()
    db.session.commit()
@app.route('/api/tasks', methods=['GET'])
def get_tasks():
    tasks = Task.query.all()
    return jsonify([ {
        'id': task.id,
        'name': task.name,
        'start': task.start,
        'end': task.end,
        'progress': task.progress,
        'dependencies': task.dependencies
    } for task in tasks])
@app.route('/api/tasks', methods=['POST'])
def add_task():
    data = request.json
    task = Task(
        name=data['name'],
        start=data['start'],
        end=data['end'],
```

```

        progress=data['progress'],
        dependencies=','.join(data.get('dependencies', []))
    )
    db.session.add(task)
    db.session.commit()
    return jsonify({
        'id': task.id,
        'name': task.name,
        'start': task.start,
        'end': task.end,
        'progress': task.progress,
        'dependencies': task.dependencies.split(',') if task.dependencies else []
    }), 201
@app.route('/api/tasks/<int:task_id>', methods=['PUT'])
def update_task(task_id):
    data = request.json
    task = Task.query.get(task_id)
    if not task:
        return jsonify({'message': 'Task not found'}), 404
    task.name = data['name']
    task.start = data['start']
    task.end = data['end']
    task.progress = data['progress']
    task.dependencies = ','.join(data.get('dependencies', []))
    db.session.commit()
    return jsonify({
        'id': task.id,
        'name': task.name,
        'start': task.start,
        'end': task.end,
        'progress': task.progress,
        'dependencies': task.dependencies.split(',') if task.dependencies else []
    })
@app.route('/', methods=['GET'])
def frontend():
    return render_template('index.html')
if __name__ == '__main__':
    app.run(debug=True)

```

ПРИМЕР ХРАНЕНИЯ ЗАДАЧ

```
[
  {
    "dependencies": "",
    "end": "2025-05-15",
    "id": 1,
    "name": "Design mockups",
    "progress": 40,
    "start": "2025-05-10"
  },
  {
    "dependencies": "1",
    "end": "2025-05-25",
    "id": 2,
    "name": "Develop frontend",
    "progress": 20,
    "start": "2025-05-16"
  },
  {
    "dependencies": "1",
    "end": "2025-05-22",
    "id": 3,
    "name": "Build backend",
    "progress": 30,
    "start": "2025-05-16"
  },
  {
    "dependencies": "2,3",
    "end": "2025-05-30",
    "id": 4,
    "name": "Integrate API",
    "progress": 0,
    "start": "2025-05-26"
  }
]
```