1. Составим декомпозицию первого уровня (Рисунок 1).

  
Рисунок 1 – Декомпозиция первого уровня

Далее составим декомпозицию второго уровня для некоторых объектов (Рисунок 2).

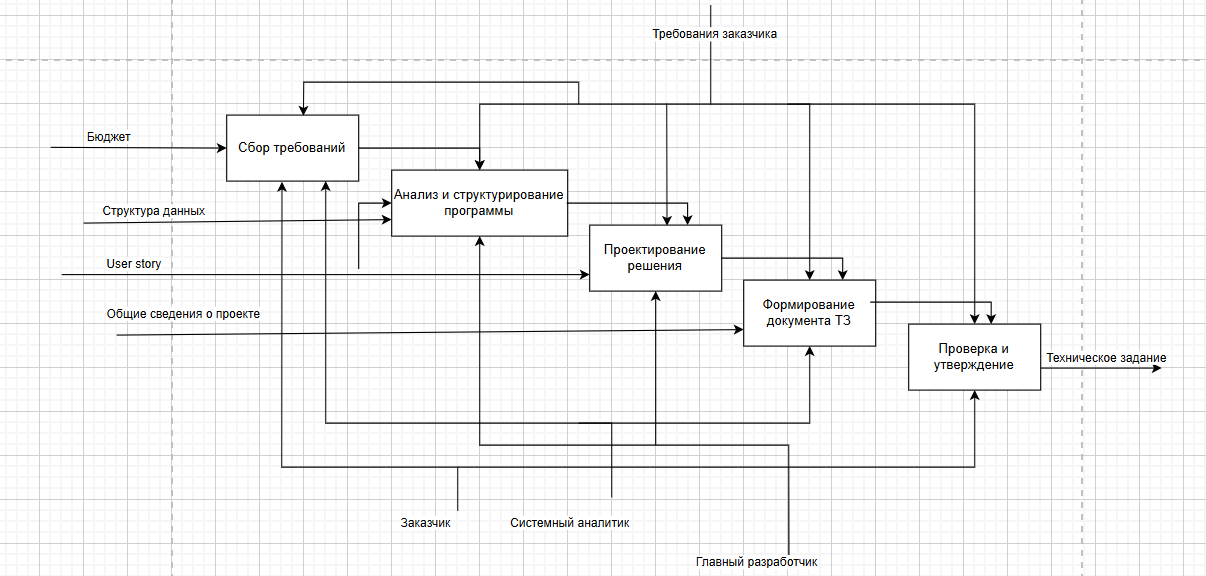


Рисунок 2 – Декомпозиция второго уровня

Теперь составим структурную декомпозицию работ (Рисунок 3).

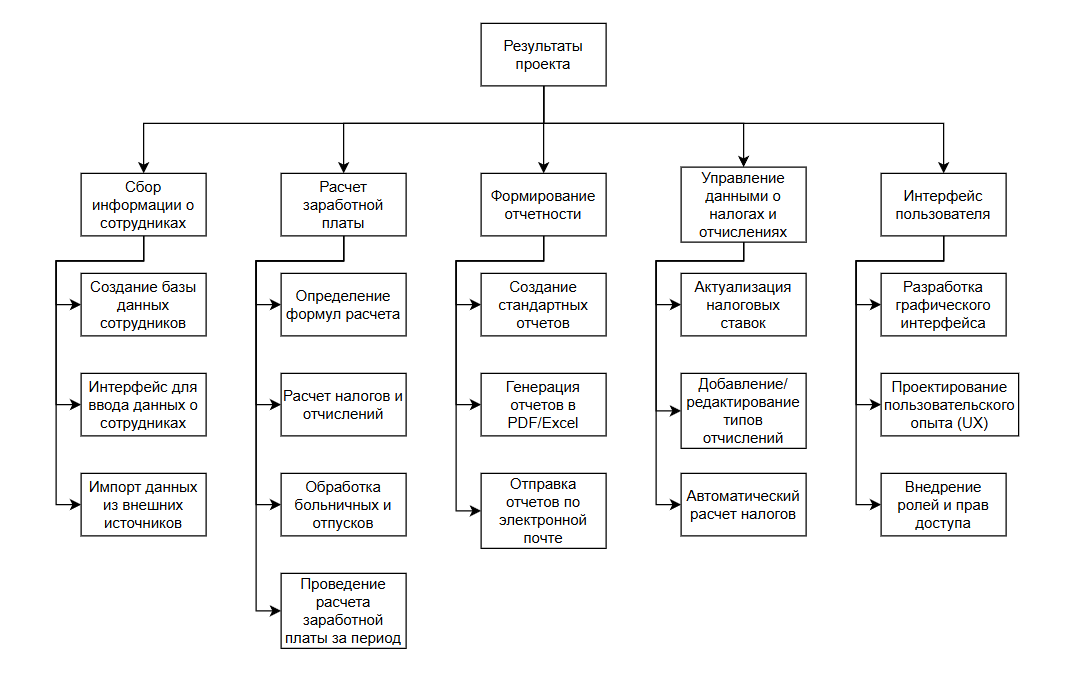


Рисунок 3 – Структурная декомпозиция

1. Проведем оценку трудовых затрат:
2. Нормативный метод

Нормативный метод основывается на стандартах и нормативов, которые обычно используются в аналогичных проектах. Включаем типичные трудозатраты для основных задач, основанных на стандартных временных нормах.

Требуется рассчитать трудоемкость выполнения проектных работ по созданию программного компонента информационной системы «Расчет заработной платы».

Решение.

п.1. Для подготовительных операций, связанных с описанием задачи программного проекта, достаточно задействовать одного специалиста в течение трех дней по 8 часов. Тогда:  
Тпо = 3 × 8 = 24,0 чел./ч.

п.2. Изменения в постановке задачи возможны, но незначительны. Проект может выполнять специалист с опытом 3–5 лет. Следовательно:  
В = 1,3; K = 1,1.

п.3. Тип задачи — учетная система. Степень новизны — типовые решения (группа В). Степень сложности — задачи учета, отчетности и статистики (группа 2). Используется язык программирования высокого уровня (например, Python). По таблице 24 определяем коэффициент:  
q = 1500; с = 1,00.

п.4. Рассчитываем условное число команд:  
Q = 1500 × 1,00 = 1500.

п.5. Рассчитываем трудоемкость описания задачи:  
То = 1500 × 1,3 / 50 × 1,1 = 42,9 чел./ч.

п.6. Трудоемкость подготовки блок-схем:  
Тбс = 1500 / 50 × 1,1 = 33,0 чел./ч.

п.7. Трудоемкость написания программного обеспечения:  
Тн = 1500 × 1,5 / 50 × 1,1 = 49,5 чел./ч.

п.8. Трудоемкость документирования (по аналогии с практикой):  
Тд = 40,0 чел./ч.

п.9. Дополнительные трудоемкости:  
Та = 1500 / (50 × 1,1) = 27,27 чел./ч  
Тп = 1500 / 50 = 30,0 чел./ч  
Тот = 1500 × 4,2 / 50 × 1,1 = 138,6 чел./ч

п.10. Общая трудоемкость проекта:  
Т = 24,0 + 42,9 + 33,0 + 49,5 + 40,0 + 27,27 + 30,0 + 138,6 = 385,27 чел./ч.

п.11. Перевод в человеко-дни и человеко-месяцы:  
Человеко-дней: 385,27 / 8 = 48,16 чел./дн.  
Человеко-месяцев: 48,16 / 21 ≈ 2,3 чел./мес.

п.12. При оплате специалиста по 700 руб./ч:  
Стоимость проекта: 385,27 × 700 = 269 689 руб.

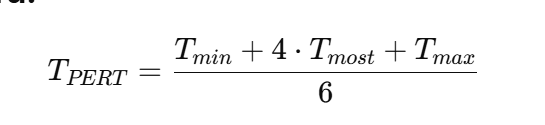
**1. Обоснование выбора коэффициентов**

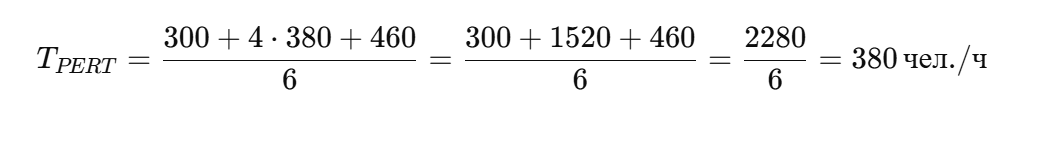
Для оценки трудоемкости программного компонента «Расчет заработной платы» были выбраны следующие параметры на основе анализа характера задачи и опыта прошлых проектов:

• Тип задачи — учетная, так как проект связан с расчетом, хранением и отчетностью по заработной плате.  
• Степень новизны — типовые решения (группа В), так как подобные компоненты широко распространены и могут разрабатываться на основе существующих практик.  
• Сложность — группа 2 (учет и отчетность), так как используются стандартные методы без сложных алгоритмов оптимизации.  
• Язык — высокого уровня (например, Python), что позволяет ускорить разработку и упростить поддержку кода.  
• Коэффициент изменений (B) выбран как 1.3 — умеренная вероятность изменений в постановке задачи.  
• Квалификация специалиста (K) — 1.1, что соответствует опыту работы от 3 до 5 лет.

**2. Оценка по методу PERT**

Метод PERT позволяет учитывать неопределенность оценки, используя три значения: оптимистичное (Tmin), наиболее вероятное (Tmost) и пессимистичное (Tmax):



Для компонента расчета ЗП значения были выбраны экспертно:  
• Tmin = 300 чел./ч  
• Tmost = 380 чел./ч  
• Tmax = 460 чел./ч  
Следовательно:  


**3. Метод аналогии**

В документе-источнике приведен пример проектирования программного продукта для учета библиотечных услуг. Его трудоемкость составила 220 чел./ч. Поскольку задача расчета заработной платы сложнее — требуется обработка финансовых данных, проверка на соответствие законодательству, возможны интеграции с кадровыми и бухгалтерскими системами — оценка трудоемкости по аналогии увеличивается на 70%:  
Tаналогии = 220 × 1.7 = 374 чел./ч

1. **Стоимость работы конкретного специалиста**

В рамках проекта «Расчет заработной платы» для корректной постановки задачи, проверки логики начислений и тестирования системы может потребоваться участие специалиста по предметной области — бухгалтера.

Предположим, участие бухгалтера охватывает следующие этапы:

* Участие в формализации требований (веха 1): 10 часов
* Проверка алгоритма расчётов и формул: 8 часов
* Участие в приёмочном тестировании: 6 часов

Общее время: 10 + 8 + 6 = 24 часа

Если принять среднюю ставку бухгалтера, выступающего как внешний эксперт или консультант, на уровне 1200 руб./ч, то расчёт стоимости будет следующим:

Стоимость=24 ч×1200 руб./ч=28 800 руб

**3. Определить вехи проекта с помощью метода целевых ориентиров.**

Метод целевых ориентиров (или этапно-временной метод) позволяет разбить реализацию программного проекта на ключевые этапы (вехи), каждая из которых отражает достижение важного промежуточного результата. Этот метод широко применяется при управлении проектами в сфере информационных технологий и способствует контролю сроков, бюджета и качества исполнения.

Для программного компонента «Расчет заработной платы» выделим основные вехи проекта. Каждая веха будет определяться как логически завершённый этап работ, после которого может быть принято решение о продолжении, корректировке или завершении проекта.

**Вехи проекта**

Веха 1. Формализация требований и постановка задачи.

На данном этапе производится сбор информации о текущих бизнес-процессах расчета заработной платы, взаимодействие с заказчиком, определение границ автоматизации и фиксация функциональных требований. Результатом является утверждённое техническое задание.

Веха 2. Проектирование архитектуры программного компонента.

Разработка архитектуры решения, в том числе схемы взаимодействия модулей, форматов хранения данных, интерфейсов с внешними системами (например, бухгалтерия, кадры). На выходе — проектная документация и структура базы данных.

Веха 3. Реализация ключевого функционала (ядра).

Программирование основной логики расчета заработной платы, включая ввод, проверку и обработку данных, расчёт начислений, налогов, удержаний и пр. Завершение вехи означает реализацию базовой функциональности.

Веха 4. Разработка интерфейсов пользователя и отчётности.

Создание графического интерфейса для взаимодействия с системой, формирование шаблонов отчётов (по сотрудникам, отделам, фондам). Этап завершается успешной сборкой интерфейсного модуля и отчётных форм.

Веха 5. Интеграция с внешними системами.

Разработка модулей обмена данными с кадровыми и бухгалтерскими системами, настройка API или файлового экспорта/импорта. Результат — проверка корректной передачи данных между системами.

Веха 6. Тестирование и отладка.

Проведение модульного и интеграционного тестирования, отладка найденных ошибок, верификация расчётных формул. Завершение этапа — отсутствие критических ошибок и успешное прохождение сценариев тестирования.

Веха 7. Документирование и обучение пользователей.

Подготовка технической и пользовательской документации, инструкции, проведение обучения персонала предприятия. Выход — утвержденный пакет документов и завершённые обучающие мероприятия.

Веха 8. Ввод в эксплуатацию и сопровождение.

Развертывание системы в продуктивной среде, запуск в опытную эксплуатацию, сбор обратной связи и подготовка к технической поддержке. Проект считается завершенным после подписания акта приёмки.

1. **Рассчитать временную продолжительность работ с помощью нормативного метода и метода параметрической оценки. Использовать вышеприведенные формулы из методических указаний по ЛР.**

Тип задачи: учетная система.

Степень новизны: типовые решения (группа В).

Группа сложности: 2 (учет, отчетность, статистика).

Уровень языка программирования: высокий (например, Python).

Коэффициенты: B = 1.3; K = 1.1; q = 1500; с = 1.00 (по таблице 4.1).

Число разработчиков: 1 человек.

Продолжительность рабочего дня: 8 часов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Уровень программирования | Группа сложности | А | Б | В | Г |
| Язык программирования высокого уровня | 1 | 1,38 | 1,26 | 1,15 | 0,69 |
| Язык программирования высокого уровня | 2 | 1,30 | 1,19 | 1,08 | 0,65 |
| Язык программирования высокого уровня | 3 | 1,20 | 1,10 | 1,00 | 0,60 |
| Язык программирования низкого уровня | 1 | 1,58 | 1,45 | 1,32 | 0,79 |
| Язык программирования низкого уровня | 2 | 1,49 | 1,37 | 1,24 | 0,74 |
| Язык программирования низкого уровня | 3 | 1,38 | 1,26 | 1,15 | 0,69 |

Условное число команд проекта:

Q = q × c = 1500 × 1.00 = 1500

Описание задачи (То):

То = (Q × B) / 50 × K = (1500 × 1.3) / 50 × 1.1 = 42.9 чел./ч

Подготовка блок-схем (Тбс):

Тбс = Q / 50 × K = 1500 / 50 × 1.1 = 33.0 чел./ч

Написание программного обеспечения (Тн):

Тн = Q × 1.5 / 50 × K = 1500 × 1.5 / 50 × 1.1 = 49.5 чел./ч

Общее время на создание продукта (Тобщ):

Тобщ = сумма всех этапов = 385.27 чел./ч

**- Перевод во временные единицы**

Продолжительность в человеко-днях (8 часов в день):

Tдни = 385.27 / 8 = 48.16 чел./дн

Продолжительность в человеко-неделях (5 рабочих дней в неделе):

Tнедели = 48.16 / 5 = 9.63 недель

Продолжительность в человеко-месяцах (21 рабочий день в месяце):

Tмес = 48.16 / 21 = 2.3 мес

**- Параметрическая оценка (PERT)**

PERT-метод учитывает три сценария оценки трудоемкости:  
Tmin = 300 чел./ч, Tmost = 380 чел./ч, Tmax = 460 чел./ч

Формула: Tpert = (Tmin + 4 × Tmost + Tmax) / 6 = (300 + 4×380 + 460) / 6 = 380 чел./ч

Tpert в днях: 380 / 8 = 47.5 чел./дн

Tpert в неделях: 47.5 / 5 = 9.5 недель

Tpert в месяцах: 47.5 / 21 = 2.26 мес

1. **Определить последовательность работ с использованием  
   сетевых диаграмм и метода графов (Диаграмма Ганта)**

Временная диаграмма показывает время начала и окончания каждого  
этапа и его длительность. Сетевая диаграмма отображает зависимости  
между различными этапами проекта (Рисунок 5).

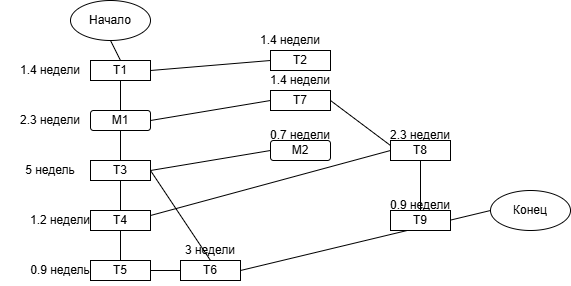


Рисунок 4 – Сетевая диаграмма этапов

| Этап | Наименование | | Длительность (нед.) |
| --- | --- | --- | --- |
| T1 | Анализ требований и постановка задачи | | 1,4 |
| M1 | Согласование требований и утверждение ТЗ | | 2,3 |
| T2 | Проектирование архитектуры системы | | 4,5 |
| T3 | Разработка алгоритмов и блок-схем | | 3,5 |
| T4 | Реализация базовой логики (начисления, налоги, удержания) | | 4,5 |
| T5 | Реализация интерфейса и отчетов | | 0,9 |
| T6 | Интеграция с внешними системами (1С, кадровая база) | | 3,0 |
| T7 | Модульное и интеграционное тестирование | | 8,0 |
| M2 | Промежуточный контроль и проверка корректности расчетов | | 0,7 |
| T8 | Подготовка технической и пользовательской документации | | 8,2 |
| T9 | Ввод в эксплуатацию, обучение пользователей, финальный аудит, подписание актов | | 1,5 |
|  | |
|  | |

- Последовательность выполнения работ (логика)

Работы связаны логически и имеют следующие зависимости:

T1 → M1 → T2 → T3

T3 → T4, T5 → T6

T4, T5, T6 → T7 → M2

M2 → T8 → T9

- Критический путь

Критический путь — это наиболее длительная по времени последовательность взаимозависимых задач. Он определяет минимально возможную продолжительность проекта.

В данном случае критический путь:  
T1 → T4 → T5 → T7   
Общая продолжительность по критическому пути:  
1,4 + 4,5 + 0,9 + 1,2 + 8,0 = 16 недель

- Интерпретация сетевой диаграммы

* Эллипсы (M1, M2 ) — контрольные точки / события проекта.
* Прямоугольники (T1–T9) — работы, выполняемые командой.
* Стрелки показывают логическую зависимость между задачами.
* Указана длительность в неделях, соответствующая расчетам на основе нормативного метода (≈16 недель).

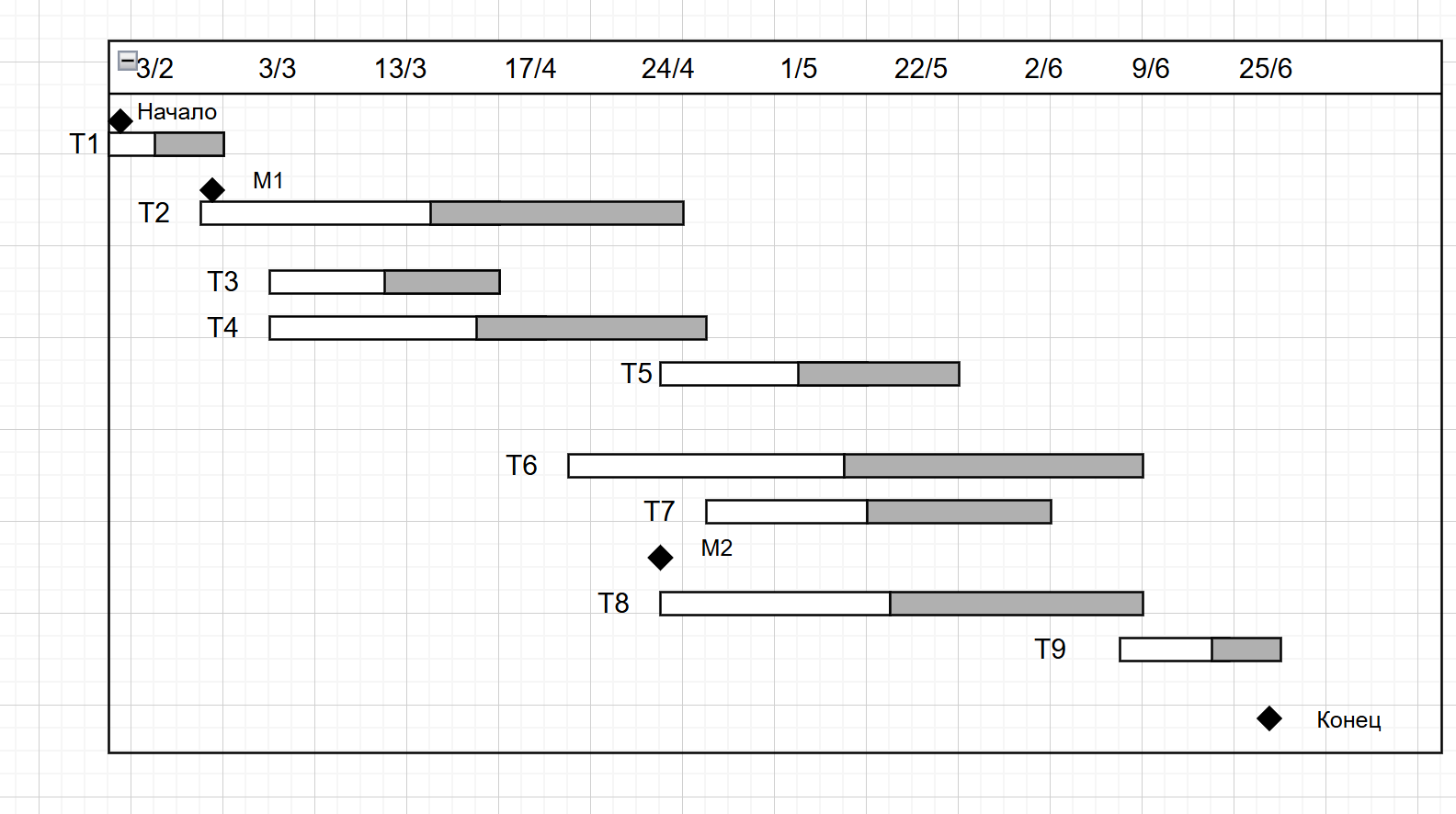


Рисунок 5 – Диаграмма Ганта

1. Провести корректировку календарного плана с учетом ограничений, используя методы оптимизации и ресурсное выравнивание.

Параллельные задачи:

* Запустить T5 (0.9 нед.) параллельно с T4 (после завершения T3).
* Начать T6 (3.0 нед.) сразу после T5, не дожидаясь завершения T4.
* Частичное тестирование (T7) можно начать после готовности первых модулей.

Ресурсное выравнивание:

* Увеличить команду на T7 (тестирование) и T8 (документация).
* Автоматизировать 30% тестов (сокращение T7 на 2 недели).

Ожидаемый результат:  
Сокращение критического пути до 12–14 недель

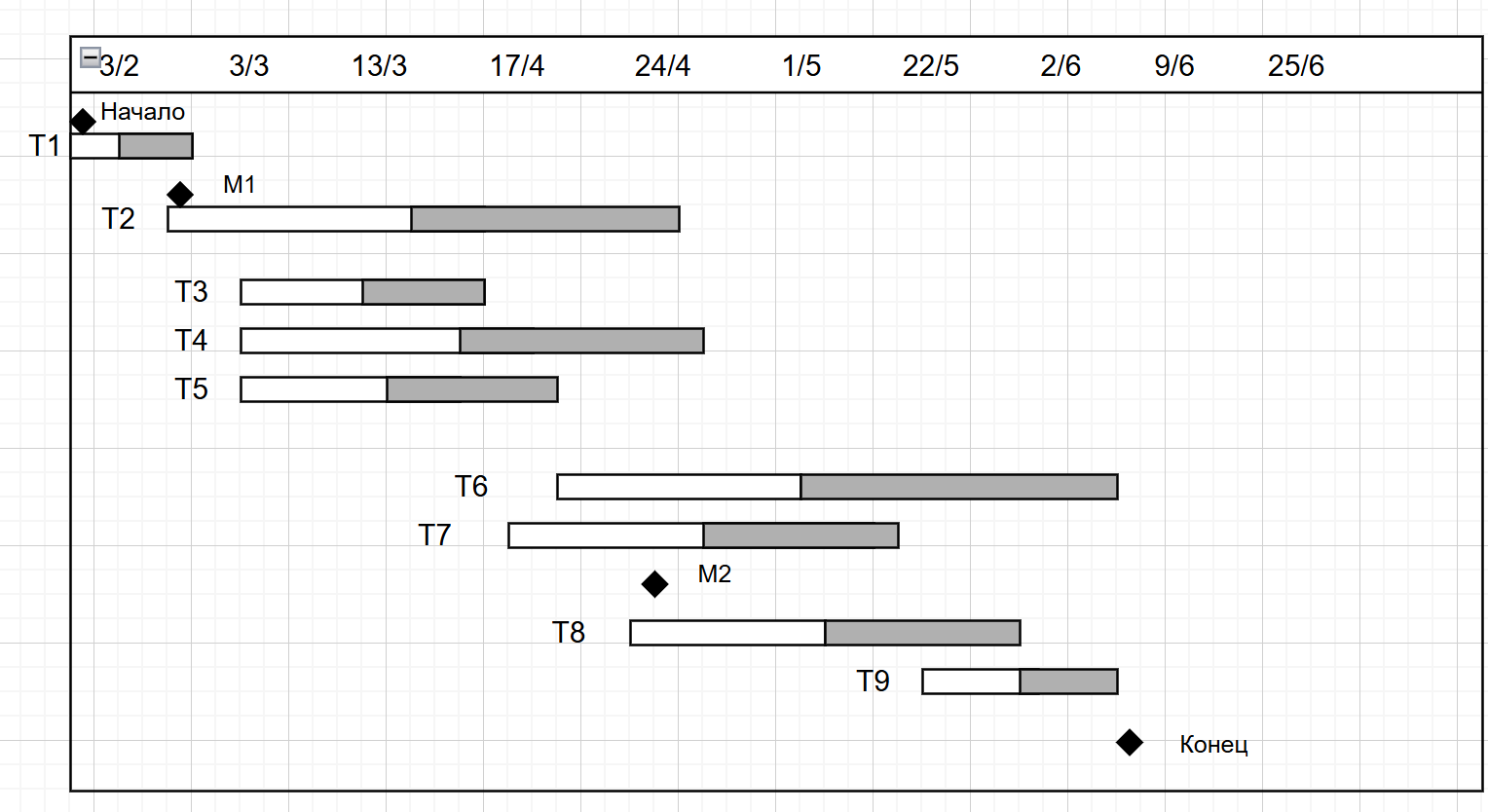


Рисунок 6 – обновленная диаграмма Ганта

1. Провести корректировку календарного плана с учетом рисков, используя методы упреждения и реагирования на риски.

- Выявляем риски для основных задач

|  |  |  |  |
| --- | --- | --- | --- |
| Задача | Потенциальные риски | Вероятность | Влияние |
| Т6 (Интеграция) | Несовместимость API с 1С | Высокая | Критическое |
| Т7 (Тестирование) | Обнаружение критических багов | Средняя | Высокое |
| Т3 (Алгоритмы) | Ошибки в логике расчетов | Низкая | Критическое |
| М1 (Согласование) | Задержки от клиента | Высокая | Среднее |

- Методы упреждения

a) Для T6 (Интеграция):

* Провести предварительный анализ API 1С до начала этапа.
* Разработать mock-сервисы для тестирования интеграции без реальных систем.

b) Для T7 (Тестирование):

* Внедрить автоматизированное тестирование на ранних этапах (параллельно с разработкой).
* Заложить дополнительные 3 дня в сроки на исправление багов.

c) Для M1 (Согласование):

* Назначить ответственного за коммуникацию с клиентом.
* Прописать в договоре штрафные санкции за задержки.

- Методы реагирования (Contingency)

a) План "Если → То":

Риск Действие

Задержка T6 Подключить внешнего интеграционного специалиста

Критические баги в T7 Выделить резервную команду тестировщиков

Ошибки в T3 Запустить экспресс-аудит кода senior-разработчиком

b) Резервы времени/ресурсов:

- Добавить буфер 10% к длительности критических задач (T6, T7).

- Закрепить 2 разработчика в резерве для срочных исправлений.

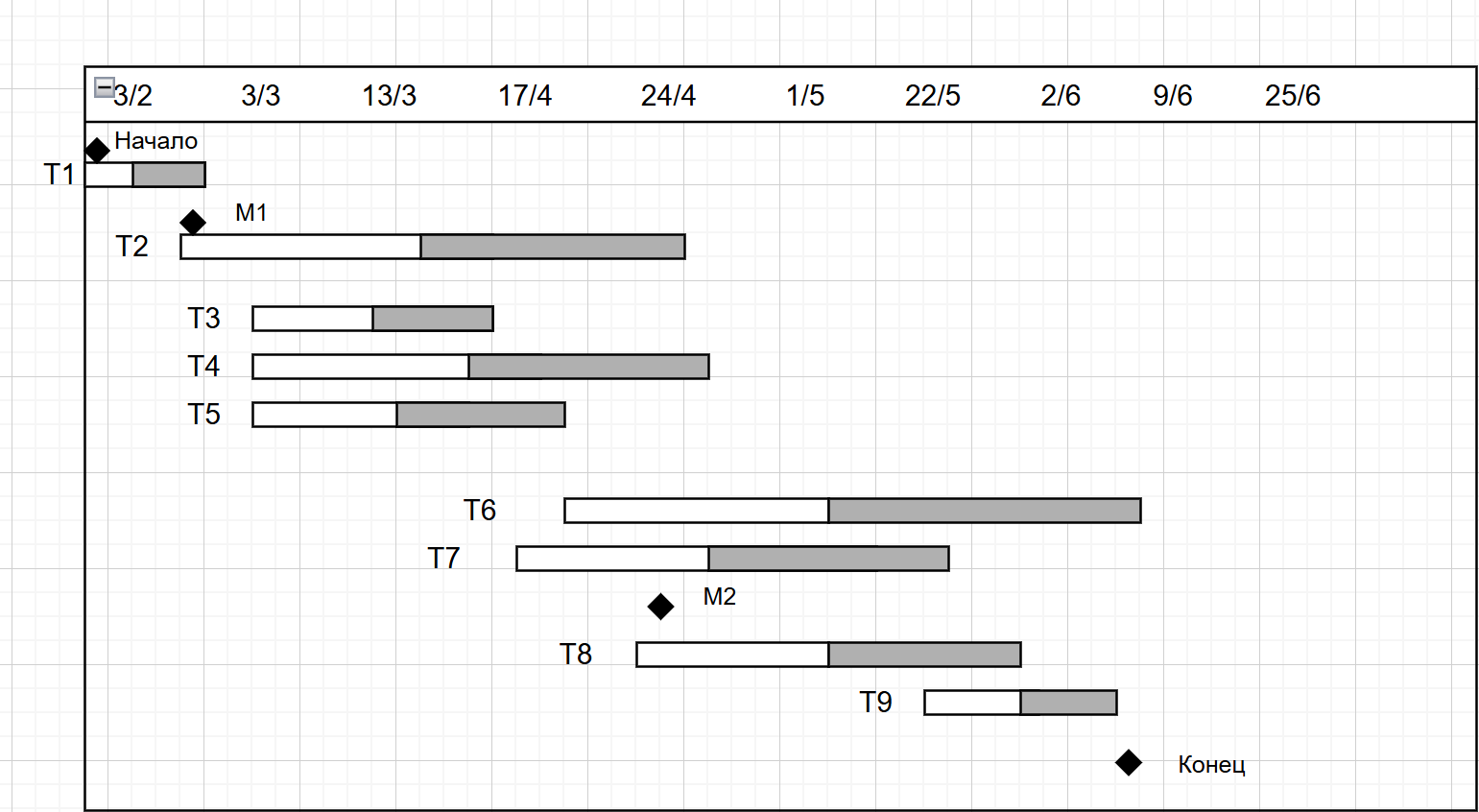
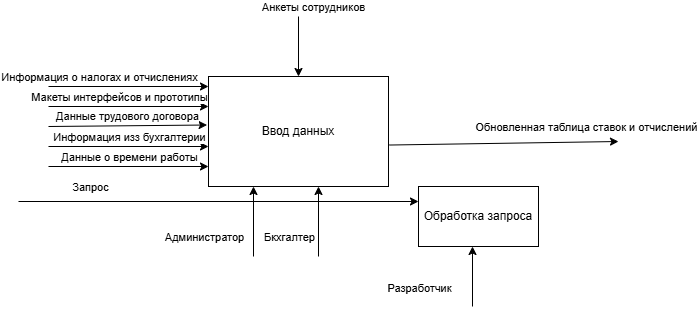


Рисунок 7 – Обновленная диаграмма Ганта

1. Провести окончательное документальное оформление календарного плана, используя метод согласования, метод принятия плановых решений.

Составим декомпозицию первого уровня (Рисунок 1).

  
Рисунок 8 – Декомпозиция первого уровня

Далее составим декомпозицию второго уровня для некоторых объектов (Рисунок 2).

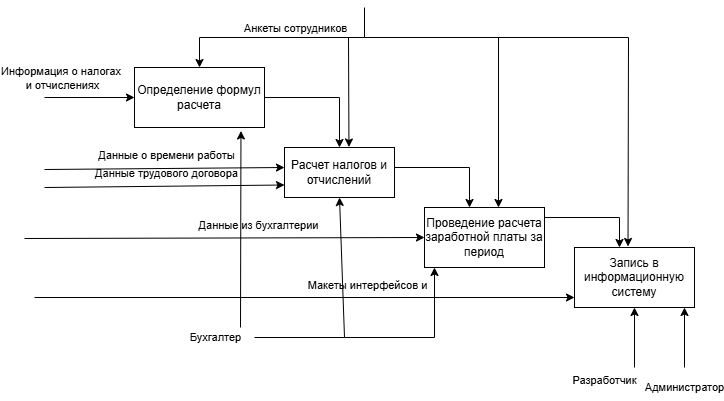


Рисунок 9 – Декомпозиция второго уровня

Теперь составим структурную декомпозицию работ (Рисунок 3).

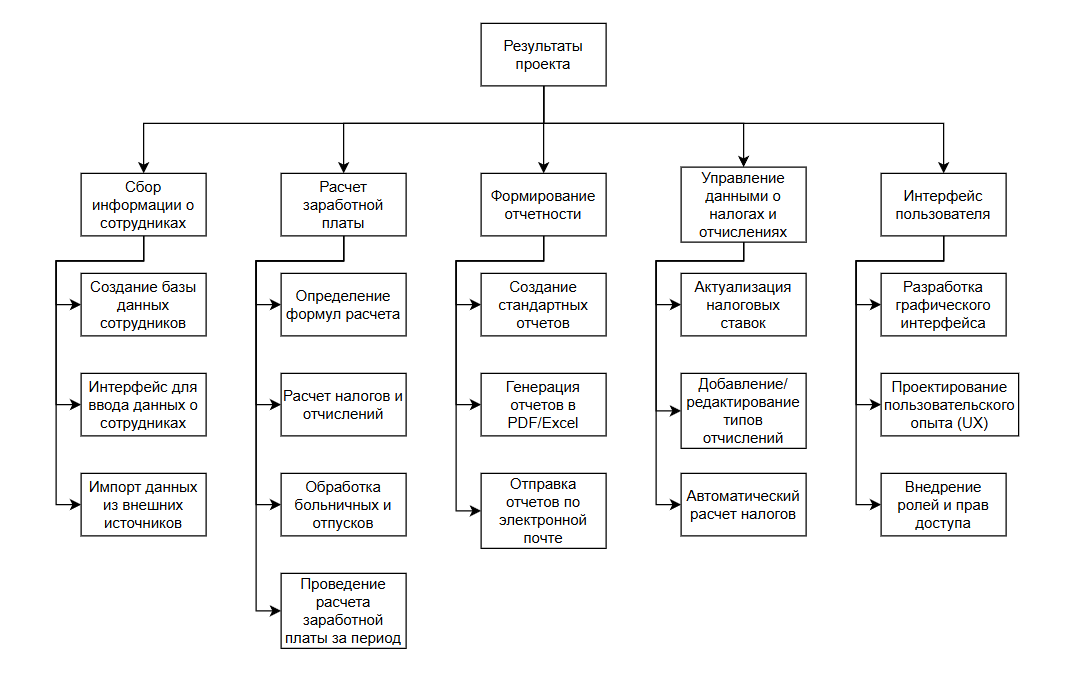


Рисунок 10 – Структурная декомпозиция

Проведем оценку трудовых затрат:

Нормативный метод

Нормативный метод основывается на стандартах и нормативов, которые обычно используются в аналогичных проектах. Включаем типичные трудозатраты для основных задач, основанных на стандартных временных нормах.

Требуется рассчитать трудоемкость выполнения проектных работ по созданию программного компонента информационной системы «Расчет заработной платы».

Решение.

п.1. Для подготовительных операций, связанных с описанием задачи программного проекта, достаточно задействовать одного специалиста в течение трех дней по 8 часов. Тогда:  
Тпо = 3 × 8 = 24,0 чел./ч.

п.2. Изменения в постановке задачи возможны, но незначительны. Проект может выполнять специалист с опытом 3–5 лет. Следовательно:  
В = 1,3; K = 1,1.

п.3. Тип задачи — учетная система. Степень новизны — типовые решения (группа В). Степень сложности — задачи учета, отчетности и статистики (группа 2). Используется язык программирования высокого уровня (например, Python). По таблице 24 определяем коэффициент:  
q = 1500; с = 1,00.

п.4. Рассчитываем условное число команд:  
Q = 1500 × 1,00 = 1500.

п.5. Рассчитываем трудоемкость описания задачи:  
То = 1500 × 1,3 / 50 × 1,1 = 42,9 чел./ч.

п.6. Трудоемкость подготовки блок-схем:  
Тбс = 1500 / 50 × 1,1 = 33,0 чел./ч.

п.7. Трудоемкость написания программного обеспечения:  
Тн = 1500 × 1,5 / 50 × 1,1 = 49,5 чел./ч.

п.8. Трудоемкость документирования (по аналогии с практикой):  
Тд = 40,0 чел./ч.

п.9. Дополнительные трудоемкости:  
Та = 1500 / (50 × 1,1) = 27,27 чел./ч  
Тп = 1500 / 50 = 30,0 чел./ч  
Тот = 1500 × 4,2 / 50 × 1,1 = 138,6 чел./ч

п.10. Общая трудоемкость проекта:  
Т = 24,0 + 42,9 + 33,0 + 49,5 + 40,0 + 27,27 + 30,0 + 138,6 = 385,27 чел./ч.

п.11. Перевод в человеко-дни и человеко-месяцы:  
Человеко-дней: 385,27 / 8 = 48,16 чел./дн.  
Человеко-месяцев: 48,16 / 21 ≈ 2,3 чел./мес.

п.12. При оплате специалиста по 700 руб./ч:  
Стоимость проекта: 385,27 × 700 = 269 689 руб.

1Обоснование выбора коэффициентов

Для оценки трудоемкости программного компонента «Расчет заработной платы» были выбраны следующие параметры на основе анализа характера задачи и опыта прошлых проектов:

• Тип задачи — учетная, так как проект связан с расчетом, хранением и отчетностью по заработной плате.  
• Степень новизны — типовые решения (группа В), так как подобные компоненты широко распространены и могут разрабатываться на основе существующих практик.  
• Сложность — группа 2 (учет и отчетность), так как используются стандартные методы без сложных алгоритмов оптимизации.  
• Язык — высокого уровня (например, Python), что позволяет ускорить разработку и упростить поддержку кода.  
• Коэффициент изменений (B) выбран как 1.3 — умеренная вероятность изменений в постановке задачи.  
• Квалификация специалиста (K) — 1.1, что соответствует опыту работы от 3 до 5 лет.

Оценка по методу PERT

Метод PERT позволяет учитывать неопределенность оценки, используя три значения: оптимистичное (Tmin), наиболее вероятное (Tmost) и пессимистичное (Tmax):

Формула: Tpert = (Tmin + 4 × Tmost + Tmax) / 6

Для компонента расчета ЗП значения были выбраны экспертно:  
• Tmin = 300 чел./ч  
• Tmost = 380 чел./ч  
• Tmax = 460 чел./ч  
Следовательно:  
Tpert = (300 + 4×380 + 460) / 6 = 380 чел./ч

Метод аналогии

В документе-источнике приведен пример проектирования программного продукта для учета библиотечных услуг. Его трудоемкость составила 220 чел./ч. Поскольку задача расчета заработной платы сложнее — требуется обработка финансовых данных, проверка на соответствие законодательству, возможны интеграции с кадровыми и бухгалтерскими системами — оценка трудоемкости по аналогии увеличивается на 70%:  
Tаналогии = 220 × 1.7 = 374 чел./ч

Стоимость работы конкретного специалиста

В рамках проекта «Расчет заработной платы» для корректной постановки задачи, проверки логики начислений и тестирования системы может потребоваться участие специалиста по предметной области — бухгалтера.

Предположим, участие бухгалтера охватывает следующие этапы:

* Участие в формализации требований (веха 1): 10 часов
* Проверка алгоритма расчётов и формул: 8 часов
* Участие в приёмочном тестировании: 6 часов

Общее время: 10 + 8 + 6 = 24 часа

Если принять среднюю ставку бухгалтера, выступающего как внешний эксперт или консультант, на уровне 1200 руб./ч, то расчёт стоимости будет следующим:

Стоимость=24 ч×1200 руб./ч=28 800 руб

Определить вехи проекта с помощью метода целевых ориентиров.

Метод целевых ориентиров (или этапно-временной метод) позволяет разбить реализацию программного проекта на ключевые этапы (вехи), каждая из которых отражает достижение важного промежуточного результата. Этот метод широко применяется при управлении проектами в сфере информационных технологий и способствует контролю сроков, бюджета и качества исполнения.

Для программного компонента «Расчет заработной платы» выделим основные вехи проекта. Каждая веха будет определяться как логически завершённый этап работ, после которого может быть принято решение о продолжении, корректировке или завершении проекта.

Вехи проекта

Веха 1. Формализация требований и постановка задачи.

На данном этапе производится сбор информации о текущих бизнес-процессах расчета заработной платы, взаимодействие с заказчиком, определение границ автоматизации и фиксация функциональных требований. Результатом является утверждённое техническое задание.

Веха 2. Проектирование архитектуры программного компонента.

Разработка архитектуры решения, в том числе схемы взаимодействия модулей, форматов хранения данных, интерфейсов с внешними системами (например, бухгалтерия, кадры). На выходе — проектная документация и структура базы данных.

Веха 3. Реализация ключевого функционала (ядра).

Программирование основной логики расчета заработной платы, включая ввод, проверку и обработку данных, расчёт начислений, налогов, удержаний и пр. Завершение вехи означает реализацию базовой функциональности.

Веха 4. Разработка интерфейсов пользователя и отчётности.

Создание графического интерфейса для взаимодействия с системой, формирование шаблонов отчётов (по сотрудникам, отделам, фондам). Этап завершается успешной сборкой интерфейсного модуля и отчётных форм.

Веха 5. Интеграция с внешними системами.

Разработка модулей обмена данными с кадровыми и бухгалтерскими системами, настройка API или файлового экспорта/импорта. Результат — проверка корректной передачи данных между системами.

Веха 6. Тестирование и отладка.

Проведение модульного и интеграционного тестирования, отладка найденных ошибок, верификация расчётных формул. Завершение этапа — отсутствие критических ошибок и успешное прохождение сценариев тестирования.

Веха 7. Документирование и обучение пользователей.

Подготовка технической и пользовательской документации, инструкции, проведение обучения персонала предприятия. Выход — утвержденный пакет документов и завершённые обучающие мероприятия.

Веха 8. Ввод в эксплуатацию и сопровождение.

Развертывание системы в продуктивной среде, запуск в опытную эксплуатацию, сбор обратной связи и подготовка к технической поддержке. Проект считается завершенным после подписания акта приёмки.

Рассчитать временную продолжительность работ с помощью нормативного метода и метода параметрической оценки. Использовать вышеприведенные формулы из методических указаний по ЛР.

Тип задачи: учетная система.

Степень новизны: типовые решения (группа В).

Группа сложности: 2 (учет, отчетность, статистика).

Уровень языка программирования: высокий (например, Python).

Коэффициенты: B = 1.3; K = 1.1; q = 1500; с = 1.00 (по таблице 4.1).

Число разработчиков: 1 человек.

Продолжительность рабочего дня: 8 часов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Уровень программирования | Группа сложности | А | Б | В | Г |
| Язык программирования высокого уровня | 1 | 1,38 | 1,26 | 1,15 | 0,69 |
| Язык программирования высокого уровня | 2 | 1,30 | 1,19 | 1,08 | 0,65 |
| Язык программирования высокого уровня | 3 | 1,20 | 1,10 | 1,00 | 0,60 |
| Язык программирования низкого уровня | 1 | 1,58 | 1,45 | 1,32 | 0,79 |
| Язык программирования низкого уровня | 2 | 1,49 | 1,37 | 1,24 | 0,74 |
| Язык программирования низкого уровня | 3 | 1,38 | 1,26 | 1,15 | 0,69 |

Условное число команд проекта:

Q = q × c = 1500 × 1.00 = 1500

Описание задачи (То):

То = (Q × B) / 50 × K = (1500 × 1.3) / 50 × 1.1 = 42.9 чел./ч

Подготовка блок-схем (Тбс):

Тбс = Q / 50 × K = 1500 / 50 × 1.1 = 33.0 чел./ч

Написание программного обеспечения (Тн):

Тн = Q × 1.5 / 50 × K = 1500 × 1.5 / 50 × 1.1 = 49.5 чел./ч

Общее время на создание продукта (Тобщ):

Тобщ = сумма всех этапов = 385.27 чел./ч

- Перевод во временные единицы

Продолжительность в человеко-днях (8 часов в день):

Tдни = 385.27 / 8 = 48.16 чел./дн

Продолжительность в человеко-неделях (5 рабочих дней в неделе):

Tнедели = 48.16 / 5 = 9.63 недель

Продолжительность в человеко-месяцах (21 рабочий день в месяце):

Tмес = 48.16 / 21 = 2.3 мес

- Параметрическая оценка (PERT)

PERT-метод учитывает три сценария оценки трудоемкости:  
Tmin = 300 чел./ч, Tmost = 380 чел./ч, Tmax = 460 чел./ч

Формула: Tpert = (Tmin + 4 × Tmost + Tmax) / 6 = (300 + 4×380 + 460) / 6 = 380 чел./ч

Tpert в днях: 380 / 8 = 47.5 чел./дн

Tpert в неделях: 47.5 / 5 = 9.5 недель

Tpert в месяцах: 47.5 / 21 = 2.26 мес

**Определить последовательность работ с использованием  
сетевых диаграмм и метода графов (Диаграмма Ганта)**

Временная диаграмма показывает время начала и окончания каждого  
этапа и его длительность. Сетевая диаграмма отображает зависимости  
между различными этапами проекта (Рисунок 11).

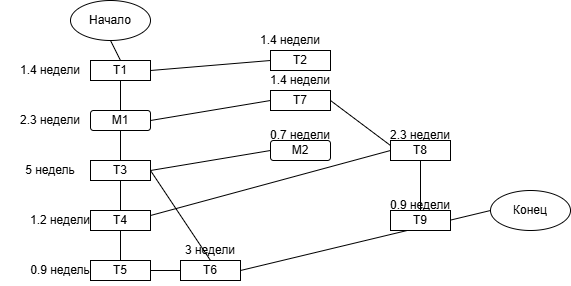


Рисунок 11 – Сетевая диаграмма этапов

| Этап | Наименование | | Длительность (нед.) |
| --- | --- | --- | --- |
| T1 | Анализ требований и постановка задачи | | 1,4 |
| M1 | Согласование требований и утверждение ТЗ | | 2,3 |
| T2 | Проектирование архитектуры системы | | 4,5 |
| T3 | Разработка алгоритмов и блок-схем | | 3,5 |
| T4 | Реализация базовой логики (начисления, налоги, удержания) | | 4,5 |
| T5 | Реализация интерфейса и отчетов | | 0,9 |
| T6 | Интеграция с внешними системами (1С, кадровая база) | | 3,0 |
| T7 | Модульное и интеграционное тестирование | | 8,0 |
| M2 | Промежуточный контроль и проверка корректности расчетов | | 0,7 |
| T8 | Подготовка технической и пользовательской документации | | 8,2 |
| T9 | Ввод в эксплуатацию, обучение пользователей, финальный аудит, подписание актов | | 1,5 |
|  | |
|  | |

- Последовательность выполнения работ (логика)

Работы связаны логически и имеют следующие зависимости:

T1 → M1 → T2 → T3

T3 → T4, T5 → T6

T4, T5, T6 → T7 → M2

M2 → T8 → T9

- Критический путь

Критический путь — это наиболее длительная по времени последовательность взаимозависимых задач. Он определяет минимально возможную продолжительность проекта.

В данном случае критический путь:  
T1 → T4 → T5 → T7   
Общая продолжительность по критическому пути:  
1,4 + 4,5 + 0,9 + 1,2 + 8,0 = 16 недель

- Интерпретация сетевой диаграммы

* Эллипсы (M1, M2 ) — контрольные точки / события проекта.
* Прямоугольники (T1–T9) — работы, выполняемые командой.
* Стрелки показывают логическую зависимость между задачами.
* Указана длительность в неделях, соответствующая расчетам на основе нормативного метода (≈16 недель).

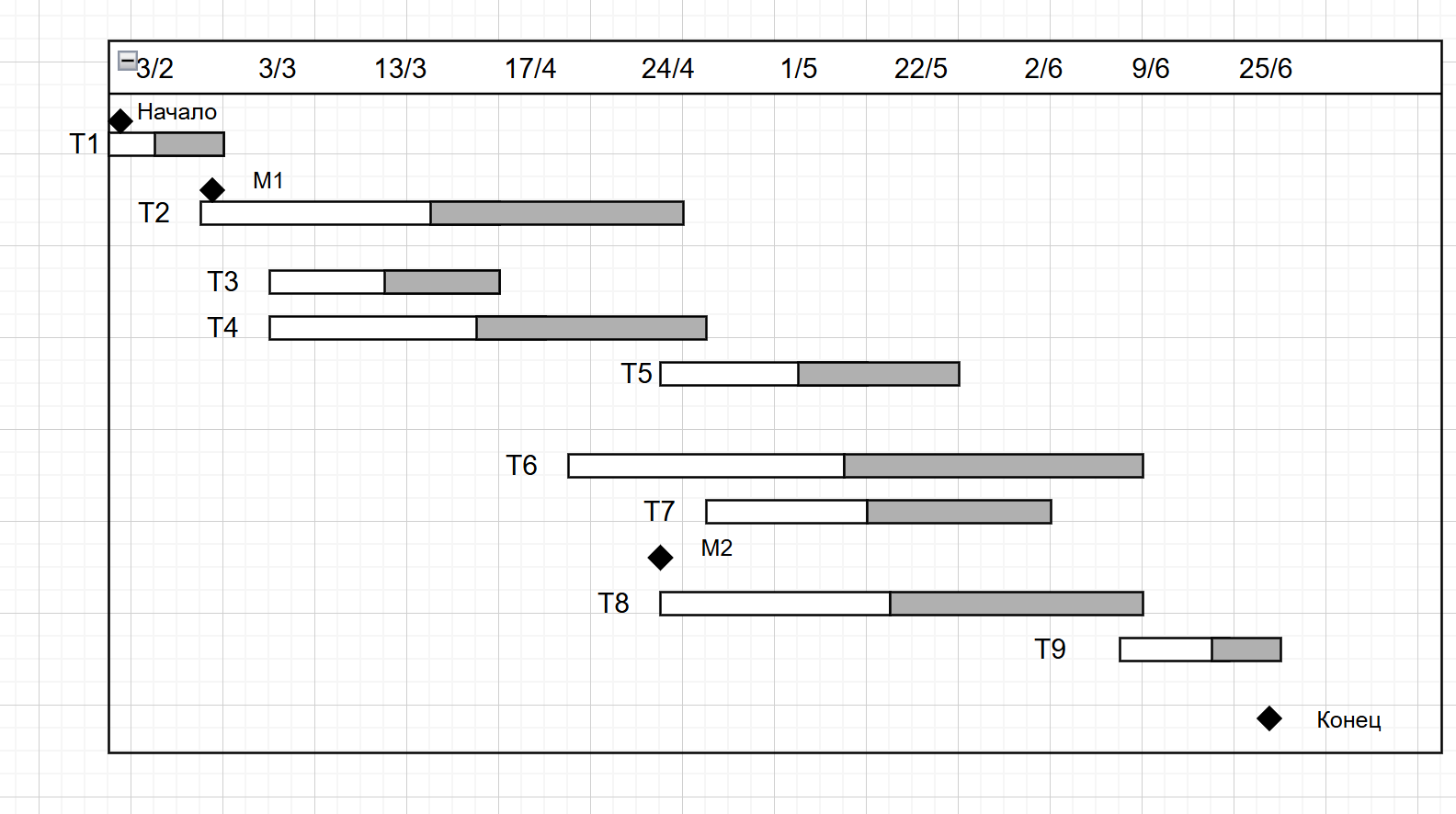


Рисунок 12 – Диаграмма Ганта

1. Провести корректировку календарного плана с учетом ограничений, используя методы оптимизации и ресурсное выравнивание.

Параллельные задачи:

* Запустить T5 (0.9 нед.) параллельно с T4 (после завершения T3).
* Начать T6 (3.0 нед.) сразу после T5, не дожидаясь завершения T4.
* Частичное тестирование (T7) можно начать после готовности первых модулей.

Ресурсное выравнивание:

* Увеличить команду на T7 (тестирование) и T8 (документация).
* Автоматизировать 30% тестов (сокращение T7 на 2 недели).

Ожидаемый результат:  
Сокращение критического пути до 12–14 недель

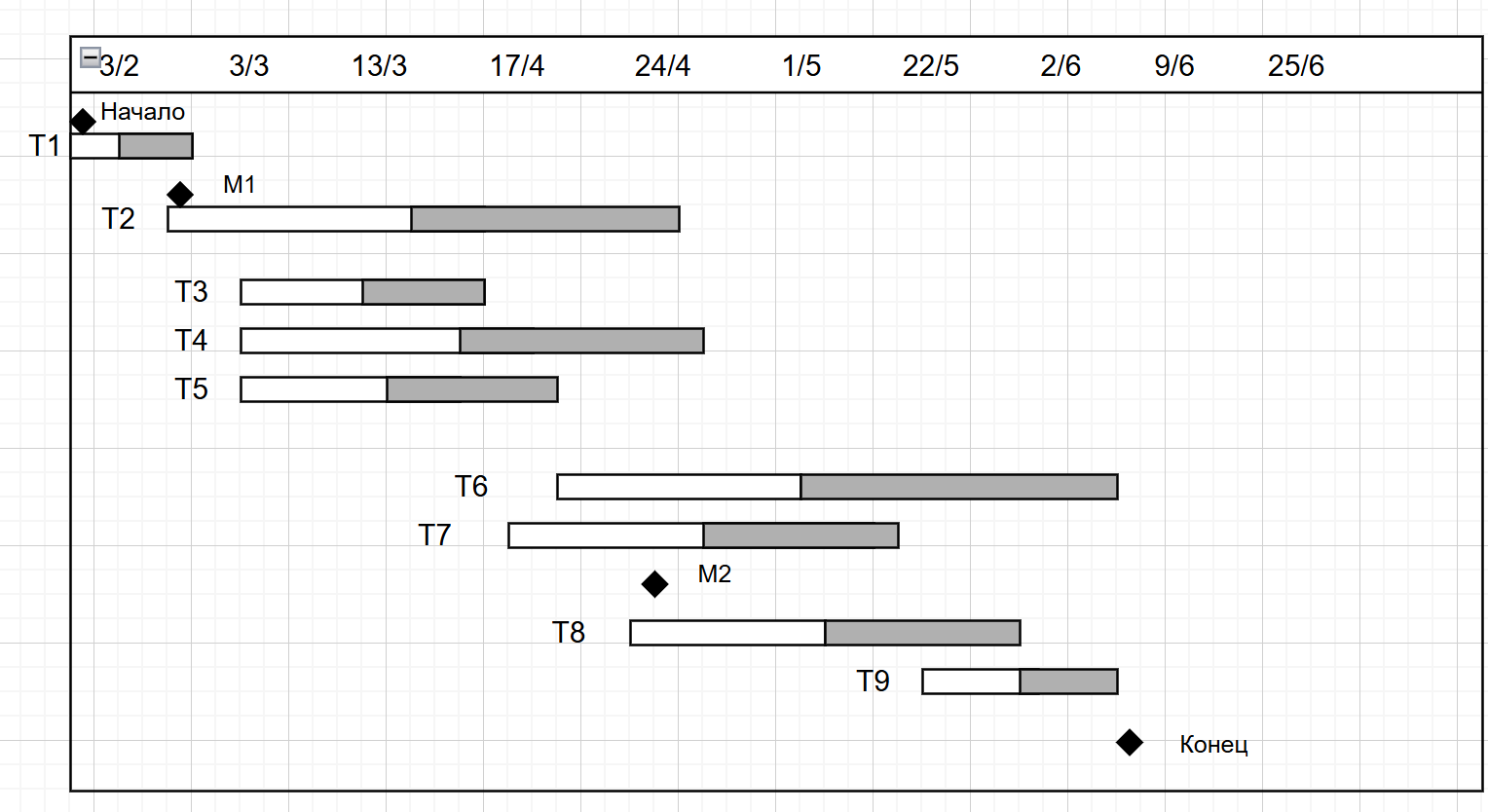


Рисунок 13 – обновленная диаграмма Ганта

Провести корректировку календарного плана с учетом рисков, используя методы упреждения и реагирования на риски.

- Выявляем риски для основных задач

|  |  |  |  |
| --- | --- | --- | --- |
| Задача | Потенциальные риски | Вероятность | Влияние |
| Т6 (Интеграция) | Несовместимость API с 1С | Высокая | Критическое |
| Т7 (Тестирование) | Обнаружение критических багов | Средняя | Высокое |
| Т3 (Алгоритмы) | Ошибки в логике расчетов | Низкая | Критическое |
| М1 (Согласование) | Задержки от клиента | Высокая | Среднее |

- Методы упреждения

a) Для T6 (Интеграция):

* Провести предварительный анализ API 1С до начала этапа.
* Разработать mock-сервисы для тестирования интеграции без реальных систем.

b) Для T7 (Тестирование):

* Внедрить автоматизированное тестирование на ранних этапах (параллельно с разработкой).
* Заложить дополнительные 3 дня в сроки на исправление багов.

c) Для M1 (Согласование):

* Назначить ответственного за коммуникацию с клиентом.
* Прописать в договоре штрафные санкции за задержки.

- Методы реагирования (Contingency)

a) План "Если → То":

Риск Действие

Задержка T6 Подключить внешнего интеграционного специалиста

Критические баги в T7 Выделить резервную команду тестировщиков

Ошибки в T3 Запустить экспресс-аудит кода senior-разработчиком

b) Резервы времени/ресурсов:

- Добавить буфер 10% к длительности критических задач (T6, T7).

- Закрепить 2 разработчика в резерве для срочных исправлений.

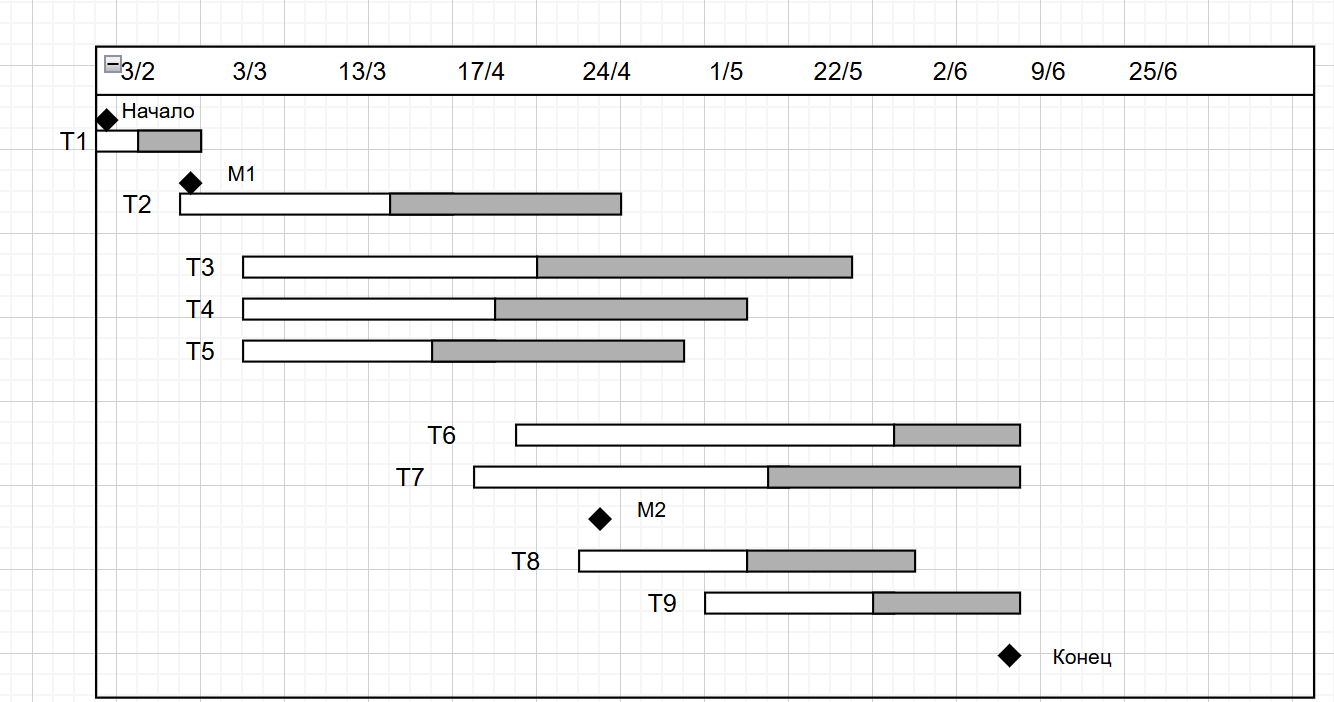


Рисунок 14 – Обновленная диаграмма Ганта

 Метод согласования

Шаги:

1. Рассылка документации ключевым стейкхолдерам:
   * Заказчику
   * Руководителям подразделений
   * Основным исполнителям
2. Внесение комментариев через:
   * Систему электронного документооборота (EDMS)
   * Комментарии в Excel/Google Sheets
   * Отдельный протокол замечаний
3. Проведение согласовательного совещания (очно/онлайн):
   * Продолжительность: 1-1.5 часа
   * Повестка:
     + Обсуждение спорных сроков
     + Корректировка распределения ресурсов
     + Фиксация зон ответственности

Распределение задач по ролям (Рисунок 15)

Исполнитель Роль Назначенные задачи Длительность (нед.)

Анна Разработчик T2, T6, T7, T5 4.5 + 3.0 + 8.0 + 0.9 = 16.4

Кирилл Системный аналитик T1, T3 1.4 + 3.5 = 4.9

Максим Тестировщик T7 , T9 4.0 + 1.5 = 5.5

Богдан Системный администратор T6 3.0

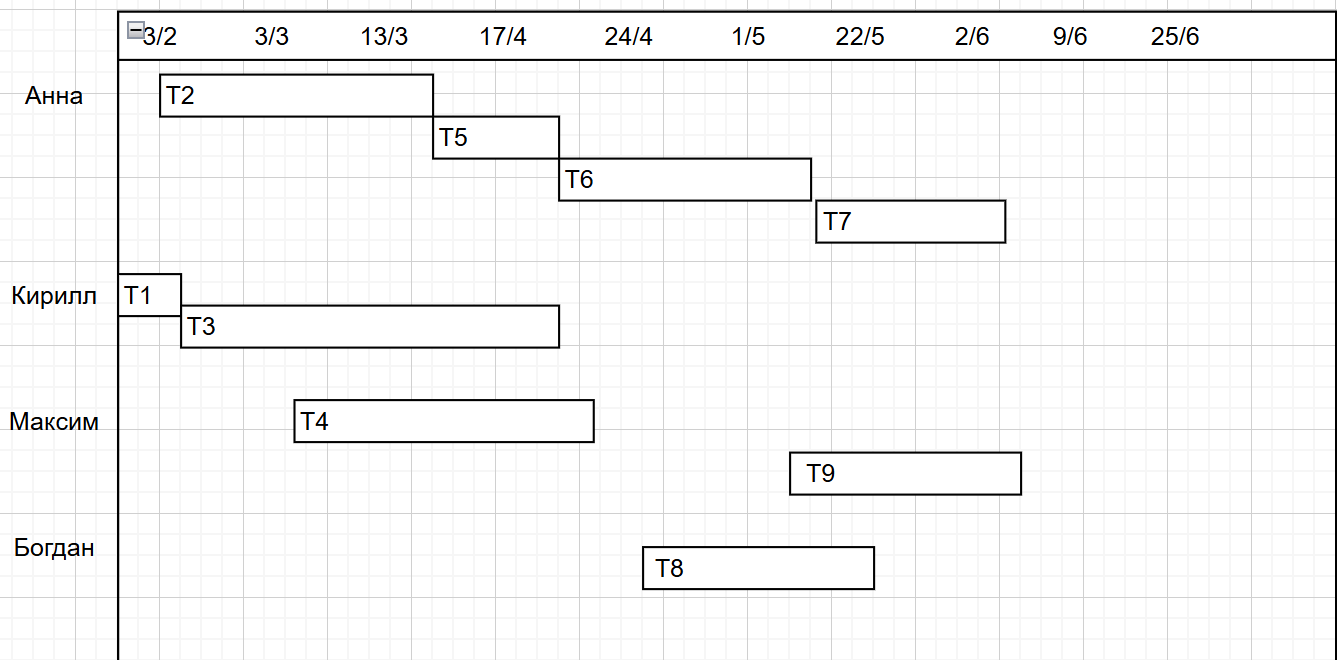


Рисунок 15 - Временная диаграмма распределения работников по этапам

 Расчет трудозатрат и затрат

| Роль | Ставка (руб./нед.) | Нагрузка (нед.) | Стоимость |
| --- | --- | --- | --- |
| Разработчик (Анна) | 60 000 | 13.5 | 810 000 руб. |
| Системный аналитик (Кирилл) | 50 000 | 8.9 | 445 000 руб. |
| Тестировщик (Максим) | 45 000 | 6.4 | 288 000 руб. |
| Системный администратор (Богдан) | 40 000 | 3.0 | 120 000 руб. |
| Итого |  | 31.8 | 1 663 000 руб. |

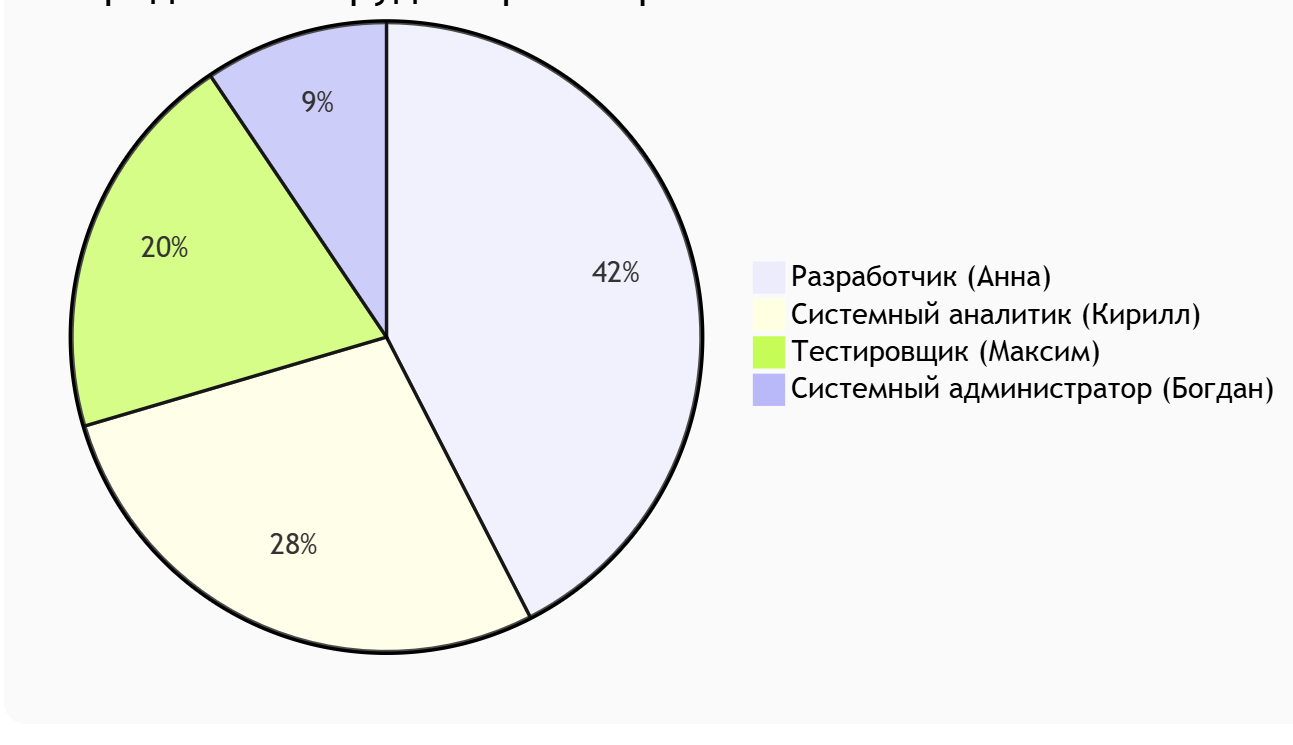


Рисунок 6 – Диаграмма трудозатрат