

---

**Мишки Гамми**

---

**Сервис “Управление таймлайнами и сбор ресурсов”**

**План разработки продукта – ИСУВИ**

**Version 0.5**

## История изменений

Дата	Версия	Описание	Авторы
17.04.2024	0.1	Написана основная информация (1-3 разделы)	Все
24.04.2024	0.2	Обновлен раздел 3 (роли по RUP), доделан раздел 4	Все
15.05.2024	0.3	Доработка раздела 4.2 - 4.4	Юра + Егор
22.05.2024	0.4	Добавлены 2 диаграммы Ганта + 4.2.6 раздел	Юра + Егор
29.05.2024	0.5	Обновили диаграмму Ганта (разделение на дни + мерж с таблицей) + 4.2.6 раздел	Егор

# Содержание

<b>1. Введение.....</b>	<b>4</b>
1.1 Назначение.....	4
1.2 Область применения.....	4
1.3 Определения и аббревиатуры.....	4
1.4 Ссылки.....	4
1.5 Обзор документа.....	4
<b>2. Обзор продукта.....</b>	<b>5</b>
2.1 Назначение, цели и контекст продукта.....	5
2.2 Влияющие факторы и ограничения.....	5
2.3 Ожидаемые результаты проекта.....	5
2.4 План развития данного документа.....	5
<b>3. Организация проекта.....</b>	<b>5</b>
3.1 Структура команды разработки.....	6
3.2 Внешние интерфейсы.....	6
3.3 Роли и обязанности.....	6
<b>4. Процесс управления.....</b>	<b>6</b>
4.1 Оценка сроков разработки проекта.....	6
4.2 План проекта.....	7
4.2.1 План фаз.....	7
4.2.2 Цели итераций.....	10
4.2.3 Релизы.....	14
4.2.4 Штатное расписание.....	16
4.2.5 Ресурсы проекта.....	23
4.2.5.1 План по сотрудникам.....	23
4.2.5.2 План поиска сотрудников.....	23
4.2.6 Бюджет.....	23
4.3 Мониторинг и контроль проекта.....	23
4.3.1 План управления требованиями.....	23
4.3.2 План управления расписанием.....	24
4.3.3 План управления бюджетом.....	24
4.3.4 План управления качеством.....	24
4.3.5 План отчетности.....	24
4.4 План управления рисками.....	25
4.5 План завершения проекта.....	25

# Software Development Plan

## (План разработки продукта)

### 1. Введение

*[Введение представляет собой обзор на весь документ в целом и включает в себя следующие разделы - назначение, область применения, определения и аббревиатуры, ссылки и обзор.]*

#### 1.1 Назначение

*[Укажите назначение данного документа.]*

Назначением данного документа является описание плана разработки ИСУВИ, а также указать разделение ролей и обязанности сотрудников.

#### 1.2 Область применения

*[Приведите краткое описание области применения данного документа, к какому(им) проекту(ам) он относится, кем будет использоваться и т.д.]*

SDP описывает план разработки, тестирования и развёртывания на оборудовании заказчика. Данный документ относится к проекту “Информационная система для управления временными изменениями”, разрабатываемому командой студентов Университета ИТМО.

#### 1.3 Определения и аббревиатуры

*[Укажите значение терминов и аббревиатур, которые употребляются в данном документе. Возможно указание ссылки на Глоссарий проекта.]*

Глоссарий ИСУВИ:  Gloss

#### 1.4 Ссылки

*[Перечислите список названия документов, на которые ссылаетесь в данном, укажите их источники.]*

Vision:  Vision

SRS:  SRS

UC:  UC

RL:  RL

#### 1.5 Обзор документа

*[Приведите краткое описание остальных разделов документа.]*

1. Введение: смысл данного документа и полезные ссылки.

2. Обзор продукта: назначение проекта, цели его разработки и артефакты, которые будут получены.
3. Организация проекта: структура команды проекта.
4. Процесс управления: примерные сроки и стоимость разработки проекта.

## **2. Обзор продукта**

### **2.1 Назначение, цели и контекст продукта**

*[Кратко опишите назначение проекта, цели его разработки, какие артефакты будут получены в результате работы над ним].*

Данный проект решает задачу контроля за изменениями таймлайнов во вселенной сериала “Локи”, в котором происходят временные катаклизмы. Проект направлен на систематизацию отслеживания таких происшествий, так как их может происходить очень много и контролировать их без помощи информационной системы очень сложно. Более того, систематизация процесса контроля позволяет эффективно собирать полезные ресурсы с самоуничтожающихся таймлайнов для последующей перепродажи, а также оперативно следить за состоянием станка, который может перегрузиться. В результате работы над проектом ожидается создание информационной системы в виде клиент-серверного приложения.

### **2.2 Влияющие факторы и ограничения**

*[Перечислите ограничения, накладываемые на данный план разработки проекта - бюджет, персонал, сроки и т.д.]*

- Нулевой бюджет
- Персонал 3 человека (студенты)
- Отсутствие бесплатного кофе и печенья, ДМС, компенсации спорта и питания
- Ограниченный ресурс времени, которое можно выделить на разработку в условиях отсутствия бюджета

### **2.3 Ожидаемые результаты проекта**

*[Перечислите артефакты, которые будут созданы в результате проекта.]*

- Исходный код серверной части программного продукта
- Исходный код клиентской части программного продукта

### **2.4 План развития данного документа**

*[Приведите план версий данного документа, на каких этапах они должны создаваться, при каких условиях нужно изменить данный документ.]*

Данный документ будет меняться в случае изменения требований заказчиком, а также:

- Изменение задач по реализации интерфейсов дизайна в соответствии с поправками заказчиков
- Изменение очередности выполнения задач в связи с изменением приоритетов

## **3. Организация проекта**

### 3.1 Структура команды разработки

*[Опишите структуру команды проекта - список сотрудников с указанием ролей, которые они занимают.]*

- **Бабалин Юрий** – Database Designer / Software Architect / Implementer / Deployment Manager
- **Зайцев Максим** – Software Architect / Implementer / User-Interface Designer / System Analyst
- **Кривонос Егор** – Database Designer / Software Architect / Implementer / Deployment Manager / System Analyst / Business-Process Analyst / Requirements Specifier / Tester

### 3.2 Внешние интерфейсы

*[Опишите, как команда связана с внешним миром - укажите контактных лиц и каналы связи.]*

Исаев Илья, Клименков Сергей, Николаев Владимир, Цопа Евгений – доверенные лица стейкхолдеров, осуществляют приемочное тестирование

### 3.3 Роли и обязанности

*[Приведите описание ролей в проекте и перечислите их обязанности.]*

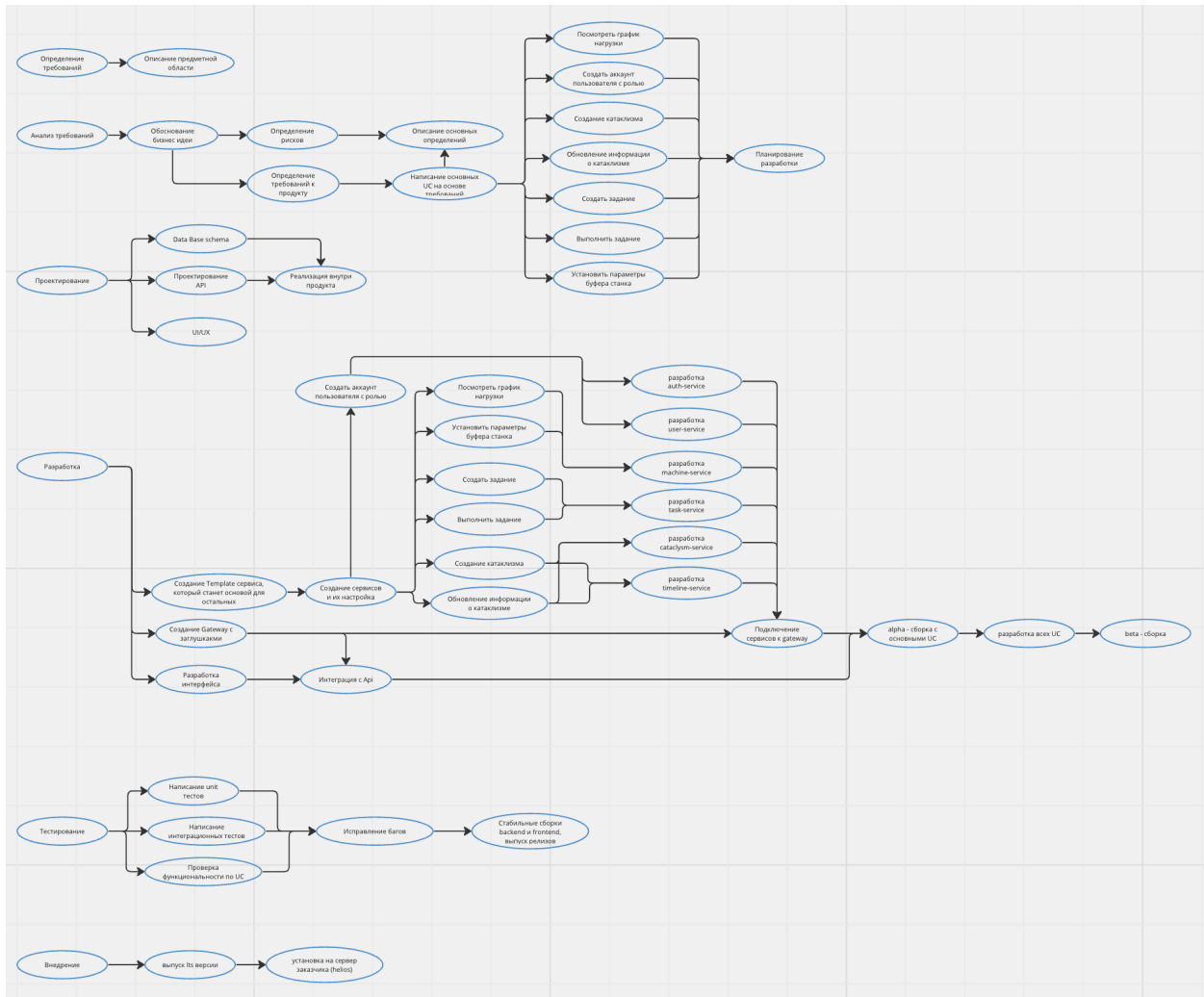
- **Database Designer** – Роль дизайнера базы данных проектирует таблицы, индексы, представления, ограничения, триггеры, хранимые процедуры, табличные пространства или параметры хранения и другие специфические для базы данных конструкции, необходимые для хранения, извлечения и удаления постоянных объектов.
- **Software Architect** – Архитектор программного обеспечения руководит и координирует техническую деятельность и артефакты в рамках проекта.
- **Implementer** – Исполнитель отвечает за разработку и тестирование компонентов в соответствии с принятыми в проекте стандартами для интеграции в более крупные подсистемы.
- **System Analyst** – Системный аналитик руководит и координирует работу по выявлению требований и моделированию сценариев использования, определяя функциональность системы и ее границы, например, устанавливая, какие субъекты и сценарии использования существуют и как они взаимодействуют.
- **Business-Process Analyst** – Аналитик бизнес-процессов руководит и координирует моделирование сценариев использования бизнеса, определяя границы моделируемой организации, например, устанавливая, какие бизнес-актеры и сценарии использования бизнеса существуют и как они взаимодействуют.
- **Requirements Specifier** – Роль спецификатора требований детализирует спецификацию части функциональности системы путем описания аспекта требований одного или нескольких вариантов использования и других вспомогательных требований к программному обеспечению.
- **User-Interface Designer** – Дизайнер пользовательского интерфейса координирует работу по созданию прототипов и дизайна пользовательского интерфейса.
- **Tester** – Тестировщик отвечает за основную деятельность по тестированию, которая заключается в проведении необходимых тестов и регистрации результатов тестирования.
- **Deployment Manager** – Роль менеджера по развертыванию отвечает за переход продукта к пользователям.

## 4. Процесс управления

### 4.1 Оценка сроков разработки проекта

[Укажите примерные сроки и стоимость разработки проекта.]

4.2 План проекта



4.2.1 План фаз

[Опишите, какие фазы будет включать в себя разработка проекта с указанием их длительности и основных вех.]

Фаза	Количество итераций	Начало	Конец
Название фазы		Ориентировочная дата начала фазы	Ориентировочная дата конца фазы

Фаза	Описание	Вехи
<i>Название фазы</i>	<i>Краткое описание основных видов деятельности в ходе данной фазы</i>	<i>Список состояний проекта (созданных артефактов), которые должны быть пройдены для окончания данной фазы</i>

Фаза	Количество итераций	Начало	Конец
Определение требований	1	07.02.24	28.02.24
Анализ требований	6	28.02.24	12.06.24
	1	01.09.24	30.09.24
Проектирование	4	08.05.24	21.06.24
Разработка	15	01.06.24	31.10.24
Тестирование	5	15.10.24	30.11.24
Внедрение	2	01.12.24	29.12.24



Фаза	Описание	Вехи
Определение требований	Описание системы, а также представления как должна работать система, также определение ограничений	Vision
Анализ требований	Обоснование бизнес идеи, определение требований и рисков, так же написание основных UC и описание основных терминов. Далее переход к разработке	SRS, UC, SDP, BC, Gloss, RL
Проектирование	Проектирование частей проекта, тк схема базы данных, арі устройство системы внутри, для дальнейшей разработки. Разработка UI/UX для дальнейшей разработки интерфейса	UI/UX, API (Swagger UI), DB schema, схема микросервисов
Разработка	Разработка общего шаблона сервиса, для дальнейшего его наследования. Так же создание Gateway-service со статическими ответами ар для дальнейшей интеграции. На основе UC разрабатываются все сервисы системы, а так же используются все технологии, которые были определены при анализе системы. Когда сервисы будет частично написаны, можно делать интеграцию с gateway-service и выпускать alpha и далее beta версий продукта при написании всех сервисов по основным UC.	.jar сервисов, docker-compose.yml, template-service в gitlab, сборка интерфейса
Тестирование	Написание unit тестов, а также интеграционных тестов с	Стабильная сборка проекта

	проверкой работы системы при различных ответах внешних сервисов.Проверка сценариев из тз и исправление их работы при некорректной работе	.jar + docker-compose.yml + ui
Внедрение	Финальная сборки и развертывание на оборудовании заказчика	Развернутое приложение, которое можно передать заказчику

#### 4.2.2 Цели итераций

*[Перечислите цели каждой итерации.]*

Фаза	Номер итерации	Описание	Вехи
<i>Название фазы</i>		<i>Краткое описание основных видов деятельности в ходе данной итерации</i>	<i>Список состояний проекта (созданных артефактов), которые должны быть пройдены для окончания данной итерации</i>

Фаза	Номер итерации	Описание	Вехи
Определение требований	1	Описание системы, а также представления как должна работать система, также определение ограничений	Vision
Анализ требований	1	Обоснование бизнес идеи	BC

	2	Описание рисков	RL
	3	Определение требований к продукту	SRS
	4	Написание UC	UC
	5	Описание основных определений	GL
	6	Планирование разработки	SDP
Проектирование	1	Создание интерфейса	UI\UX (figma)
	2	Проектирование Базы данных	DB schema
	3	Проектирование api	API (Swagger UI)
	4	Создание схемы микросервисов	Схема продукта
Разработка	1	Разработка интерфейса	Интерфейс
	2	Создание gateway-service с заглушками api	.jar, docker-compose.yml, Dockerfile

	3	Создание template-service для дальнейшего создания сервисов	Шаблон сервиса в gitlab
	4	Создание сервисов и настройка	.jar, docker-compose.yml, Dockerfile
	5	Разработка auth-service	.jar, docker-compose.yml, Dockerfile
	6	Разработка user-service	.jar, docker-compose.yml, Dockerfile
	7	Разработка machine-service	.jar, docker-compose.yml, Dockerfile
	8	Разработка task-service	.jar, docker-compose.yml, Dockerfile
	9	Разработка cataclysm-service	.jar, docker-compose.yml, Dockerfile
	10	Разработка timeline-service	.jar, docker-compose.yml, Dockerfile

	11	Подключение сервисов к gateway-service	.jar, docker-compose.yml, Dockerfile
	12	Интеграция с api	готовый интерфейс
	13	Публикация alpha версии с основными UC	.jar - всех сервисов, готовый интерфейс
	14	Разработка всех UC	.jar - всех сервисов, готовый интерфейс
	15	Выпуск beta версии	.jar - всех сервисов, готовый интерфейс
Тестирование	1	Проверка всех UC	Рабочая сборка .jar и docker-compose.yml файлами
	2	Написание unit тестов	
	3	Написание интеграционных тестов	
	4	Исправление багов	сборка .jar и docker-compose.yml файлами с исправленными ошибками
	5	Стабильная сборка	Стабильная .jar и docker-compose.yml

			файлами с исправленными ошибками
Внедрение	1	Выпуск Its версии	jar и docker-compose.yml
	2	Развертывание на мощностях заказчика	Развернутое приложение

#### 4.2.3 Релизы

*[Перечислите релизы проекта, их назначение и даты выпуска]*

Номер версии	Описание релиза	Дата выпуска
0.0.1-alpha	Основной функционал системы на основе UC	30.06.24
0.0.1-beta	Написание первой части дополнительных UC-ов	06.09.24
0.0.2-beta	Написание второй части дополнительных UC-ов	13.09.24
0.0.3-beta	Полная реализация продукта	20.09.24
0.0.4-beta	Написание unit тестов	27.09.24

0.0.5-beta	Написание интеграционных тестов	04.10.24
1.0.0-release	Сборка готового продукта	11.10.24
1.0.1-release	Исправление багов после проверки всех УС	18.10.24
1.0.2-release	Стабильная сборка продукта	25.10.24
1.0-lts	Версия развернутая на оборудовании заказчика	01.11.24
1.1-lts	Версия развернутая на оборудовании заказчика после правок	08.11.24

#### 4.2.4 Штатное расписание

*[Данный раздел содержит расписание разработки проекта, с указанием трудозатрат на реализацию каждой из фаз и итераций.]*

Роль	Ставка (для расчета стоимости реализации)
Database Designer	55\$-65\$ / час - senior
Software Architect	60\$-70\$ / час - senior
Implementer	20\$-30\$ / час - middle
System Analyst	20\$-25\$ / час - middle
Business-Process Analyst	20\$-25\$ / час - middle
Requirements Specifier	15\$-25\$ / час - middle
User-Interface Designer	25\$-30\$ / час - middle
Tester	15\$-20\$ / час - middle
Deployment Manager	20\$-25\$ / час - middle



Название задачи	Data base Designer	Soft ware Architect	Impl ementer	Syst em Analyst	Busine ss-Pro cess Analyst	Req uire ment s Specifier	User-Interf ace Designer	Te st er	Depl oym ent Manager	Общие трудозатраты на задачу (от всех ролей (часы))	Стоимость реализации	
Определение требований												
Сбор требований	0	0	0	15	45	0	0	0	0	60	\$ 1,350.00	123,201.00 Р
Анализ требований												
Анализ бизнес-идеи	0	0	0	0	60	0	0	0	0	60	\$ 1,350.00	123,201.00 Р
Обоснование бизнес идеи	0	0	0	0	51	0	0	0	0	51	\$ 1,147.50	104,720.85 Р
Определение требований к продукту	0	0	0	0	39	0	0	0	0	39	\$ 877.50	80,080.65 Р
Разработка Use Case	0	0	0	0	18	0	96	0	0	114	\$ 3,045.00	277,886.70 Р
Планирование разработки	0	0	0	15	24	0	0	0	0	39	\$ 877.50	80,080.65 Р
Описание основных определений	0	0	0	0	6	0	0	0	0	6	\$ 135.00	12,320.10 Р
Проектирование												
Проектирование backend/frontend	0	39	0	6	0	3	0	0	0	48	\$ 2,730.00	249,139.80 Р
Проектирование схемы БД	24	0	0	3	0	3	0	0	0	30	\$ 1,567.50	143,050.05 Р
Проектирование интерфейсов	0	0	0	3	0	3	24	0	0	30	\$ 787.50	71,867.25 Р
Проектирование API	0	48	0	0	0	0	0	0	0	48	\$ 3,120.00	284,731.20 Р
Разработка												
Разработка API	0	0	60	0	0	0	0	0	0	60	\$ 1,500.00	136,890.00 Р
Разработка backend	0	0	126	0	0	0	0	0	0	126	\$ 3,150.00	287,469.00 Р
Разработка UI	0	0	156	0	0	0	0	0	0	156	\$ 3,900.00	355,914.00 Р

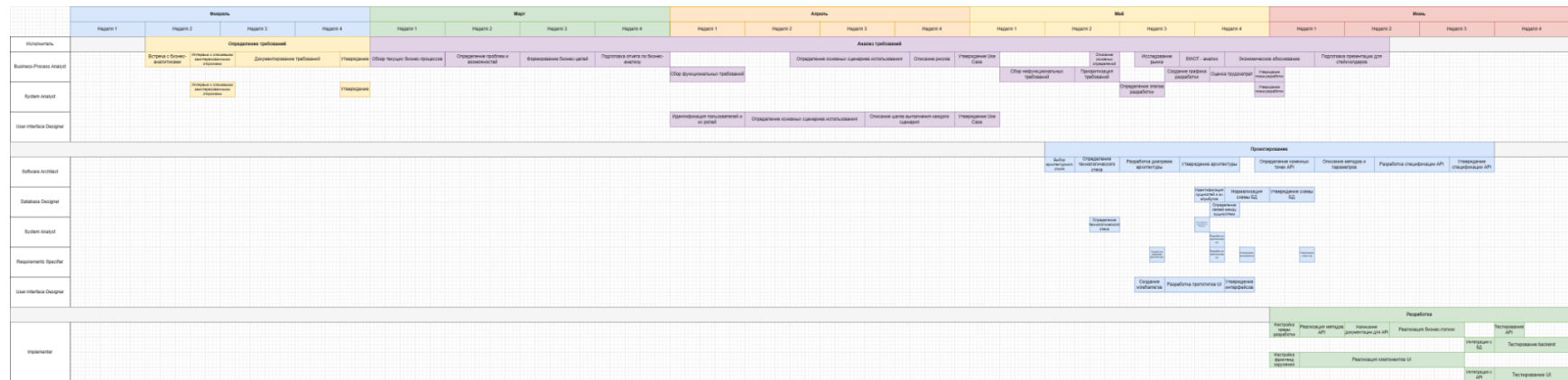
Тестирование												
Написание unit-тестов	0	0	36	0	0	0	0	24	0	60	\$ 1,320.00	120,463.20 ₽
Написание интеграционных тестов	0	0	36	0	0	0	0	24	0	60	\$ 1,320.00	120,463.20 ₽
Тестирование UC	0	0	18	0	0	0	0	30	0	48	\$ 975.00	88,978.50 ₽
Внедрение												
Развертывание на Helios	0	0	0	0	0	0	0	0	30	30	\$ 675.00	61,600.50 ₽
Создание документации по использованию	0	0	0	0	30	0	0	0	0	30	\$ 675.00	61,600.50 ₽
Общая стоимость:											\$ 30,502.50	2,783,658.15 ₽

## Декомпозированные задачи:

1. Определение требований
  - a. Сбор требований
    - i. Встреча с бизнес-аналитиками
    - ii. Интервью с ключевыми заинтересованными сторонами
    - iii. Документирование требований
    - iv. Утверждение требований
2. Анализ требований
  - a. Анализ бизнес-идеи
    - i. Обзор текущих бизнес-процессов
    - ii. Определение проблем и возможностей
    - iii. Формирование бизнес-целей
    - iv. Подготовка отчета по бизнес-анализу
  - b. Разработка Use Case
    - i. Идентификация пользователей и их ролей
    - ii. Определение основных сценариев использования
    - iii. Описание шагов выполнения каждого сценария
    - iv. Утверждение Use Case
  - c. Обоснование бизнес идеи
    - i. Исследование рынка
    - ii. SWOT - анализ
    - iii. Экономическое обоснование
    - iv. Подготовка презентации для стейкхолдеров
  - d. Определение требований к продукту
    - i. Сбор функциональных требований
    - ii. Сбор нефункциональных требований
    - iii. Приоритизация требований
  - e. Описание основных определений
  - f. Планирование разработки
    - i. Определение этапов разработки
    - ii. Создание графика разработки
    - iii. Оценка трудозатрат
    - iv. Утверждение плана разработки
3. Проектирование
  - a. Проектирование backend/frontend
    - i. Выбор архитектурного стиля
    - ii. Определение технологического стека
    - iii. Разработка диаграмм архитектуры
    - iv. Утверждение архитектуры
  - b. Проектирование схемы БД
    - i. Идентификация сущностей и их атрибутов
    - ii. Определение связей между сущностями
    - iii. Нормализация схемы БД
    - iv. Утверждение схемы БД
  - c. Проектирование интерфейсов
    - i. Создание wireframe'ов
    - ii. Разработка прототипов UI
    - iii. Утверждение интерфейсов
  - d. Проектирование API
    - i. Определение конечных точек API

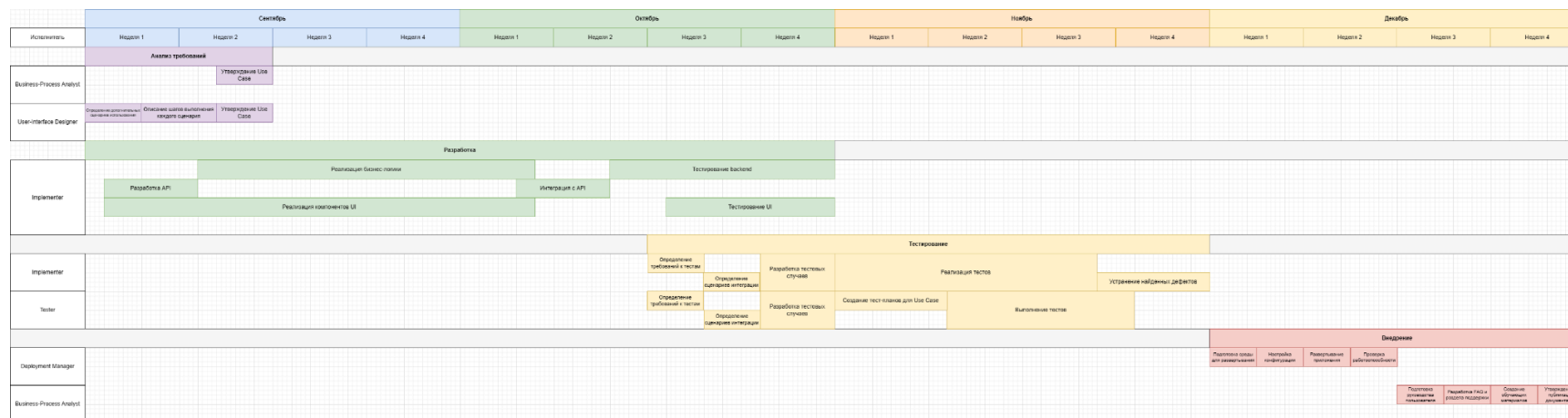
- ii. Описание методов и параметров
  - iii. Разработка спецификации API
  - iv. Утверждение спецификации API
- 4. Разработка
  - a. Разработка API
    - i. Настройка среды разработки
    - ii. Реализация методов API
    - iii. Написание документации для API
    - iv. Тестирование API
  - b. Разработка backend
    - i. Реализация бизнес-логики
    - ii. Интеграция с БД
    - iii. Тестирование backend
  - c. Разработка UI
    - i. Настройка фронтенд окружения
    - ii. Реализация компонентов UI
    - iii. Интеграция с API
    - iv. Тестирование UI
- 5. Тестирование
  - a. Написание unit-тестов
    - i. Определение требований к тестам
    - ii. Разработка тестовых случаев
    - iii. Реализация тестов
    - iv. Выполнение тестов
  - b. Написание интеграционных тестов
    - i. Определение сценариев интеграции
    - ii. Разработка тестовых случаев
    - iii. Реализация тестов
    - iv. Выполнение тестов
  - c. Тестирование Use Case
    - i. Создание тест-планов для Use Case
    - ii. Выполнение тестов
    - iii. Устранение найденных дефектов
- 6. Внедрение
  - a. Развертывание на Helios
    - i. Подготовка среды для развертывания
    - ii. Настройка конфигурации
    - iii. Развертывание приложения
    - iv. Проверка работоспособности
  - b. Создание документации по использованию
    - i. Подготовка руководства пользователя
    - ii. Разработка FAQ и раздела поддержки
    - iii. Создание обучающих материалов
    - iv. Утверждение и публикация документации

## Диаграмма Ганта 1 (Весна)



Полная версия: SDP.drawio

### Диаграмма Ганта 2 (Осень)



Полная версия:  SDP.drawio

#### 4.2.5 Ресурсы проекта

##### 4.2.5.1 План по сотрудникам

*[Приведите описание сотрудников, необходимых для разработки проекта - их навыки, требуемый опыт, количество. Укажите, в каких фазах и итерациях они понадобятся.]*

Сейчас у нас в команде есть

- Backend-developer / software-architect/database designer / deployment-manager / tester
- Frontend-developer / user-interface designer / tester
- System-analytic / business-process analytic / requirements Specifier / tester

Этого нам будет достаточно для реализации базовой версии продукта (МВП)

##### 4.2.5.2 План поиска сотрудников

*[Опишите, каким образом будут найдены необходимые проекту сотруднику (если их ещё нет в команде)]*

Ищем людей, которые пишут на Java и традиционной ориентации. Так же готовы писать фронт на js. Чтобы мы могли реализовать проект. Далее при расширении штата будем брать на работу проводя 2 этапа собеседования. Алгоритмический и технический. Соответственно при расширении штата нужно будет расширять бюджет.

#### 4.2.6 Бюджет

*[Определите бюджет проекта]*

Сумма \$	Объект	Описание
30502.50	Зарплата	<a href="#">Рассчитана</a> в таблице на основе трудозатрат + обоснована на диаграмме Ганта
0	Оборудование	У нас есть свое оборудование для работы, а сервера для развертывания приложения предоставляет заказчик

#### 4.3 Мониторинг и контроль проекта

##### 4.3.1 План управления требованиями

*[Определите, каким образом будет контролироваться процесс работы с требованиями проекта.]*

Каждый этап анализа требований при достижении готовности обсуждается с заказчиком во время встреч. Реализуемые УС согласуются с заказчиком и утверждаются в документе УС.

Во время проектирования мы должны предоставить заказчику схемы базы данных, спроектированное арі и схему продукта в виде схемы микросервисов.

#### 4.3.2 План управления расписанием

*[Определите, каким образом будут контролироваться сроки выполнения работ проекта.]*

Мягкие рамки каждой из фаз определены в данном документе, жёсткие рамки определены сроком готовности доверенных лиц стейкхолдеров (преподавателями) принимать разработанную ИСУВИ. Управление расписание через тіго с помощью канбан доски. Выбор в сторону тіго был осуществлен вследствие того, что они требуют большое количество ресурсов.

#### 4.3.3 План управления бюджетом

*[Определите, каким образом будет контролироваться бюджет проекта.]*

Бюджет контролирует project manager, зарплаты распределяются на основе опыта сотрудников.

#### 4.3.4 План управления качеством

*[Определите, каким образом будет контролироваться качество проекта.]*

Этап планирование:

- проектирование схемы базы данных: Разработчик должен обосновать почему схема имеет определенную структуру.
- Проектирование арі: Разработчик должен согласовать арі с фронтенд разработчиком и защитить его.
- Реализация внутри продукта: Архитектор должен спроектировать систему и защитить её перед командой

Этап разработка:

- Проведение ревью кода для задач
- Тестирование задачи, после ее реализации

Этап тестирование:

- Написание unit тестов
- Написание интеграционных тестов
- Проверка УС полностью

#### 4.3.5 План отчетности

*[Определите, как часто будут создаваться отчеты по прогрессу разработки проекта, а также каково будет их содержание.]*



Документы формата docx, оформленные согласно шаблонам RUP, период отчетности - сокращённый спринт (1 неделя), за исключением июля-августа 2024.

#### 4.4 План управления рисками

*[Определите, каким образом будут контролироваться риски проекта.]*

Для управления рисками проекта разработан план, включающий стратегии смягчения и действия при наступлении рисков. Основные риски включают несогласованность требований, текучку кадров, несанкционированный доступ и временные катаклизмы, а также специфические для предметной области. Для каждого риска определены меры по снижению вероятности его наступления и планы действий на случай его реализации, такие как улучшение коммуникаций, усиление безопасности и разработка резервных планов.

Документ:  RL

#### 4.5 План завершения проекта

*[Опишите, каким образом будет определяться завершение разработки проекта, какие цели должны быть достигнуты для этого, что должно быть сделано после завершения разработки.]*

Развернутая на сервере Helios разработанная ИСУВИ, реализована вся функциональность (можно выполнить каждый из UseCase-ов), отсутствуют критические баги, препятствующие этому.