**План управления программным проектом (SPMP) для менеджера бюджета**

Утверждаю

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Дата\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25.02.25 Михайлов Иван: Создание первой версии документации

**1. Введение**

**1.1 Обзор проекта**

Этот проект организован для разработки программного приложения "Менеджер бюджета" — инструмента, который поможет пользователям отслеживать свои доходы и расходы. Проект выполняется командой из трех студентов в рамках учебного курса по разработке программного обеспечения. Приложение будет разработано с использованием языка программирования Python, библиотеки Flet для создания пользовательского интерфейса и базы данных SQLite3 для хранения данных. Разработка будет проходить в три этапа (итерации), чтобы обеспечить постепенное улучшение функциональности и качества продукта.

**1.2 Результирующие артефакты проекта**

Следующие материалы должны быть поставлены в указанные сроки.

**Версия 1 (прототип)** с документацией — вторая неделя марта 2025 года

**Версия 2** с документацией — третья неделя апреля 2025 года.

**Версия 3 (финальная)** с документацией — третья неделя мая 2025 года.

Документация для каждой версии включает: SPMP (План управления проектом, данный документ), SRS (Спецификация требований к программному обеспечению), SDD (Проектная документация программного обеспечения), STP (План тестирования программного обеспечения), исходный код, исполняемый файл приложения, руководство пользователя.

**1.3 Развитие SPMP**

Этот документ будет поддерживаться и обновляться лидером проекта (назначение роли будет указано в разделе 2). Лидер проекта отвечает за актуальность SPMP, внося изменения по мере необходимости, не реже одного раза в неделю. Документ будет находиться под управлением конфигурации (например, с использованием Git), чтобы отслеживать все изменения. SPMP в основном следует рекомендациям стандарта IEEE 1058.1-1987, адаптированным под учебный проект.

**1.4 Ссылочные материалы**

Основное руководство: Software Engineering: an Objected-Oriented Perspective E. Braude, Wiley, 2000

Руководство по Python: официальная документация Python (python.org)

Документация Flet: официальный сайт библиотеки (flet.dev)

Документация SQLite3: официальная документация (sqlite.org)

**1.5. Аббревиатуры**

SPMP — Softivare Project Management Plan. План управления программным  
проектом (данный документ)  
SRS \_ Software Requiremcnts Specification. Спецификация требований к про-  
граммному обеспечению,  
SDD — Softwvare Design Document. Проектная документация программного  
обеспечения.

STD – документация по тестированию программного обеспечения.

**Flet** — Библиотека для создания интерфейсов на Python.

**SQLite3** — Легковесная система управления базами данных.

**Git** — Система управления версиями.

1. **Организация проекта**

#### 2.1. Модель процесса

Разработка проекта "Менеджер расходов" будет осуществляться с использованием итеративной модели разработки, включающей три итерации. Каждая итерация охватывает определенные фазы: анализ требований, проектирование, реализацию и тестирование. Такой подход позволяет постепенно улучшать приложение, адаптируясь к новым требованиям и отзывам преподавателя.

* **Итерация 1 (прототип)**: Завершена к середине февраля 2025 года. Включает настройку репозитория (SCMP), базовую работу с SQLite3 (добавление и удаление транзакций), создание простого интерфейса с использованием Flet и выпуск тестовой версии.
* **Итерация 2**: Запланирована на март–апрель 2025 года. Включает завершение SQAP (план тестирования), улучшение интерфейса, внедрение визуализации данных (графики) и тестирование работы с базой данных.
* **Итерация 3**: Запланирована на апрель–май 2025 года. Включает финальное тестирование, доработку UX/UI, документирование структуры базы данных и версионности кода, а также подготовку к сопровождению приложения.

Детали задач и расписание будут уточняться по мере продвижения проекта и обратной связи от преподавателя.

#### 2.2. Организационная структура

Проект "Менеджер расходов" выполняется небольшой командой из трех студентов в рамках учебного курса. Команда организована как группа равных участников с распределением ролей для выполнения задач. Назначенные роли включают:

* **Лидер команды** — отвечает за общее управление проектом, ведение SPMP и координацию работы команды. Эту роль выполняет Иван Михайлов.
* **Ответственный за конфигурацию и базу данных** — отвечает за настройку репозитория, управление версиями кода и работу с SQLite3.
* **Ответственный за интерфейс и тестирование** — отвечает за разработку и улучшение интерфейса с использованием Flet, а также за тестирование приложения.

Роли для Дмитрия Марфина и Семена Сергачева будут уточнены в разделе 2.4 и могут корректироваться в процессе работы. Все участники команды участвуют в инспектировании работы друг друга. Инспектирование проводится либо совместно на еженедельных встречах, либо индивидуально, если времени на совместное обсуждение недостаточно.

#### 2.3. Организационные рамки и взаимосвязи

Команда проекта взаимодействует со следующими сторонами:

* **Преподаватель** — основное контактное лицо, предоставляет требования, оценивает результаты и дает обратную связь.
* **Учебная группа** — возможные пользователи прототипа для тестирования и получения отзывов (при необходимости).

Команда работает автономно, но может обращаться к преподавателю за консультациями по техническим или организационным вопросам.

#### 2.4. Ответственность за проект

Ответственность участников проекта представлена в таблице ниже. Каждый участник отвечает за создание соответствующих документов или частей проекта вовремя, а лидер команды координирует процесс и определяет, кто выполняет конкретные задачи. Документы поддерживаются в актуальном состоянии ответственными лицами.

| **Участник** | **Лидер команды** | **Ответственный за конфигурацию и БД** | **Ответственный за интерфейс и тестирование** |
| --- | --- | --- | --- |
| **Имя** | Иван Михайлов | Дмитрий Марфин | Семен Сергачев |
| **Отвечает за связь** | Преподаватель | — | Учебная группа (при необходимости) |
| **Отвечает за документ/задачу** | SPMP, объединение SQLite и FLet | Анализ задач, SCMP, работа с SQLite | SQAP, STP, интерфейс Flet |

**Таблица 2.1. Ответственность участников проекта "Менеджер расходов"**

1. **Управляющий процесс**

#### 3.1. Цели и приоритеты

Основная цель проекта "Менеджер расходов" — разработать функциональное приложение для отслеживания доходов и расходов, которое будет принято преподавателем в качестве итоговой работы учебного курса. Приоритеты проекта выстроены следующим образом:

1. **Достижение заданного уровня качества**: Приложение должно быть стабильным, удобным и соответствовать требованиям преподавателя (например, работа с базой данных, визуализация данных, минимальный интерфейс).
2. **Соблюдение сроков**: Завершение проекта к третьей неделе мая 2025 года для своевременной сдачи.
3. **Выполнение требований**: Реализация ключевых функций (добавление/удаление транзакций, визуализация данных, улучшенный интерфейс).
4. **Повторное использование кода**: Создание модульной структуры, которую можно будет использовать в будущих учебных проектах (например, работа с SQLite3 или интерфейс на Flet).

Привлекательный и полностью отполированный продукт не является главным приоритетом, так как это учебный проект, но удобство использования (UX/UI) будет доработано в третьей итерации для положительной оценки.

#### 3.2. Допущения, зависимости и ограничения

Проект основан на следующих предположениях и условиях:

* Команда из трех студентов имеет базовые навыки программирования на Python и сможет освоить SQLite3 и Flet в процессе работы.
* Преподаватель предоставит обратную связь после каждой итерации, что может повлиять на требования и приоритеты.
* Ограничение по времени: проект должен быть завершен к маю 2025 года без возможности продления сроков.
* Технические зависимости: доступ к компьютерам с установленными Python, SQLite3 и Flet, а также стабильный интернет для работы с GitHub.
* Отсутствие внешних факторов (например, сбоев оборудования или потери данных), которые могут задержать разработку.

Если преподаватель изменит требования (например, потребует добавить экспорт данных в Excel), это может потребовать перераспределения задач между итерациями.

#### 3.3. Управление рисками

Управление рисками проводится для минимизации проблем, которые могут повлиять на выполнение проекта. Риски обсуждаются на еженедельных встречах команды с использованием мозгового штурма. Ниже приведены основные риски, выявленные на текущем этапе:

**Риск №1: Проблемы с визуализацией данных**

**Описание**: Никто в команде не имеет опыта создания графиков (например, с использованием библиотек вроде Matplotlib или Plotly в связке с Flet). Неясно, насколько сложно интегрировать графики в интерфейс и будет ли это работать стабильно.

**Последствия**: Задержка второй итерации или снижение качества визуализации.

**Меры**: Изучить документацию заранее, протестировать простые графики до начала итерации 2.

**Риск №2: Сложности с улучшением интерфейса** (вместо мобильной адаптации)

**Описание**: Улучшение интерфейса (добавление страниц и удобство UX) может занять больше времени, чем ожидается, из-за ограниченного опыта работы с Flet.

**Последствия**: Интерфейс может остаться недостаточно удобным, что повлияет на оценку.

**Меры**: Выделить время в итерации 2 для изучения возможностей Flet и тестирования новых страниц.

**Риск №3: Ошибки в работе с базой данных**

**Описание**: Неправильная структура SQLite3 или ошибки в запросах могут привести к потере данных или сбоям приложения.

**Последствия**: Задержка тестирования и исправлений в третьей итерации.

**Меры**: Провести тестирование базы данных в итерации 2 и документировать структуру в итерации 3.

#### 3.4. Механизмы мониторинга и контроля

Для контроля прогресса и координации действий команда будет:

* Проводить еженедельные встречи по понедельникам с 18:00 до 19:00 (время можно скорректировать по вашему расписанию). На этих встречах обсуждаются выполненные задачи, планы на следующую неделю и выявленные риски. Если повестка отсутствует, встреча отменяется.
* Резервировать время по пятницам с 17:00 до 18:00 для дополнительных обсуждений, если потребуется. Лидер команды (Иван Михайлов) уведомляет участников о необходимости встречи не позднее четверга 16:00.
* Использовать GitHub Issues для отслеживания задач и проблем. Каждый участник фиксирует свои задачи и прогресс в репозитории.

Лидер команды готовит краткий отчет о статусе проекта перед каждой встречей (например, что сделано, что в плане). Отчеты по качеству (тестирование) и конфигурации (GitHub) готовят ответственные участники (Дмитрий и Семен, соответственно).

#### 3.5. План расстановки кадров

Распределение ролей приведено в таблице ниже. Каждый участник выполняет свои задачи и участвует в проверке работы других членов команды.

**Таблица 3.1. Расстановка участников проекта "Менеджер расходов"**

| **Имя** | **Лидер команды** | **Ответственный за конфигурацию и БД** | **Ответственный за интерфейс и тестирование** |
| --- | --- | --- | --- |
| Иван Михайлов | ✓ |  |  |
| Дмитрий Марфин |  | ✓ |  |
| Семен Сергачев |  |  | ✓ |

**Таблица 3.1. Расстановка участников проекта "Менеджер расходов"**

**4. Технический процесс**

#### 4.1. Методы, инструменты и технологии

В проекте "Менеджер расходов" используются следующие технологии и подходы:

* **Язык программирования**: Python — основной язык разработки благодаря его простоте и широкой поддержке библиотек.
* **Библиотека интерфейса**: Flet — для создания кроссплатформенного графического интерфейса (десктоп и веб).
* **База данных**: SQLite3 — легковесная система управления базами данных для хранения транзакций.
* **Управление версиями**: Git и GitHub — для хранения кода и совместной работы. Репозиторий создан в первой итерации и содержит структуру проекта.
* **Инструменты визуализации**: Для графиков в итерации 2 планируется использовать библиотеку Matplotlib или Plotly (конкретный выбор будет сделан после тестирования).
* **Методология**: Объектно-ориентированный подход для модульности кода, чтобы упростить поддержку и повторное использование.

Документирование кода ведется с использованием комментариев в стиле Python docstrings там, где это необходимо (подробности в SRS).

#### 4.2. Документация программного обеспечения

Документация проекта включает:

* **SPMP**: Данный план управления проектом.
* **SRS**: Спецификация требований, описывающая функциональность приложения (например, добавление транзакций, отображение графиков).
* **SDD**: Проектная документация, включающая структуру базы данных и архитектуру приложения.
* **STP**: План тестирования, описывающий подходы к проверке функциональности (см. SQAP).

Детали документирования качества изложены в SQAP (будет завершен в итерации 2). Все документы хранятся в папке docs репозитория на GitHub и обновляются ответственными участниками.

#### 4.3. Функции сопровождения проекта

На время разработки (февраль–май 2025 года) сопровождение проекта ограничивается поддержкой команды:

* **Техническая поддержка**: Каждый участник отвечает за свои инструменты.
* **После сдачи**: Планируется подготовка краткого плана сопровождения (в итерации 3), чтобы преподаватель или другие студенты могли использовать приложение. Это включает документацию структуры базы данных и инструкцию по запуску.

Дополнительный персонал для сопровождения не привлекается, так как это учебный проект.

**5. Распределение работ, план-график и бюджет**

#### 5.1. Распределение работ

Распределение работ между итерациями и участниками команды будут показано в таблице ниже. Так как детальное проектирование еще не завершено, конкретные задачи могут уточняться по мере продвижения проекта.

#### 5.2. Зависимости

Выполнение задач в проекте "Менеджер расходов" зависит от следующих факторов и связей между этапами:

* **Итерация 1**:
  + Настройка репозитория (GitHub) должна быть завершена до начала работы с кодом и базой данных.
  + Работа с SQLite3 (добавление/удаление транзакций) зависит от определения структуры базы данных.
  + Базовый интерфейс Flet зависит от завершения минимальной функциональности работы с БД, чтобы отображать данные.
* **Итерация 2**:
  + Завершение SQAP (план тестирования) зависит от анализа результатов первой итерации и обратной связи преподавателя.
  + Улучшение интерфейса (добавление страниц) зависит от стабильности базового интерфейса из итерации 1.
  + Визуализация данных (графики) требует готовности базы данных с тестовыми транзакциями для отображения.
  + Тестирование работы с БД зависит от внедрения всех изменений в SQLite3 из итерации 2.
* **Итерация 3**:
  + Финальное тестирование зависит от завершения всех функций (интерфейс, графики, БД) в итерации 2.
  + Доработка UX/UI зависит от результатов тестирования и отзывов преподавателя.
  + Документирование структуры БД и версионности кода зависит от финальной структуры базы данных и истории коммитов в GitHub.
  + Подготовка к сопровождению зависит от завершения всей документации и кода.

**Внешние зависимости:**

* Доступ к компьютерам с установленным ПО (Python, SQLite3, Flet).
* Стабильный интернет для работы с GitHub.
* Обратная связь преподавателя после каждой итерации, которая может повлиять на приоритеты задач.

Если какая-либо зависимость не выполняется (например, задержка с настройкой БД или проблемы с Flet), это может сместить сроки последующих задач.

#### 5.3. Аппаратные ресурсы

Для разработки используются личные компьютеры участников команды со следующими минимальными характеристиками:

* Процессор: Intel Core i3 или эквивалент.
* Оперативная память: 8 ГБ.
* Дисковое пространство: 10 ГБ свободного места.
* Операционная система: Windows 10/11, macOS или Linux.
* Установленное ПО: Python 3.10+, SQLite3, Flet, Git.
* Доступ в интернет для работы с GitHub.

Дополнительное оборудование не требуется, так как проект ориентирован на десктопное использование.

#### 5.4. Выделение бюджета и ресурсов

Поскольку это учебный проект, реальный бюджет отсутствует. Затраты ограничены временем участников и использованием бесплатных инструментов (Python, SQLite3, Flet, GitHub). Оценка усилий проведена на основе:

1. **Неформальной оценки сверху вниз**: Опыт команды из предыдущих учебных заданий (примерно 10–15 человеко-недель).
2. **Функционального размера**: Оценка ключевых функций (работа с БД, интерфейс, графики) и экстраполяция на весь проект (около 12–16 человеко-недель).

| **Метод** | **Минимум (человеко-недели)** | **Максимум (человеко-недели)** | **Комментарий** |
| --- | --- | --- | --- |
| Неформальная оценка | 10 | 15 | На основе опыта команды |
| Функциональный размер | 12 | 16 | Учет основных функций |
| Наиболее консервативная оценка | 12 | 16 | Максимум минимумов и максимумов |

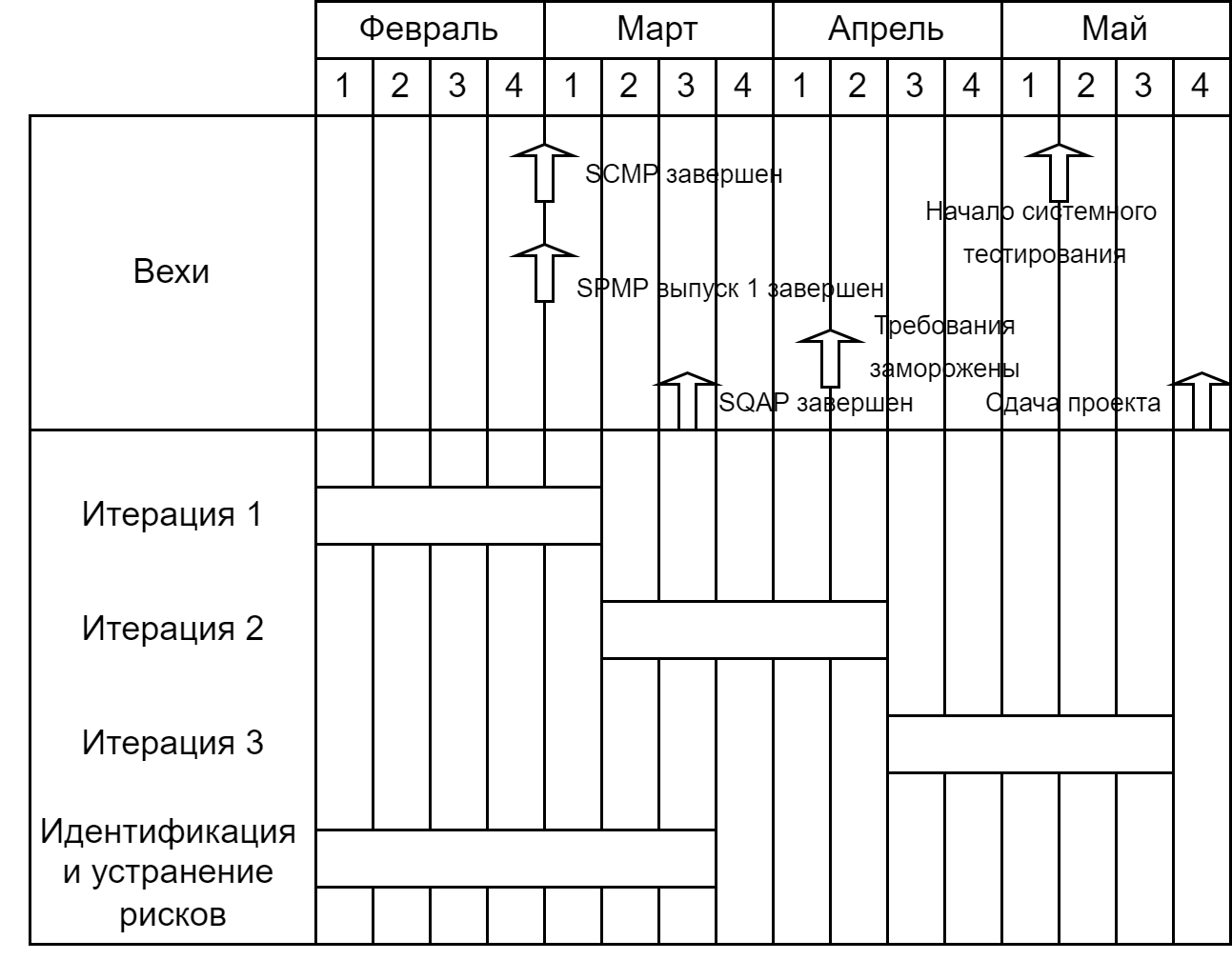
**Таблица 5.2. Оценка усилий**

Принята наиболее консервативная оценка (12–16 человеко-недель), так как требования могут уточняться преподавателем. Уточненные оценки будут представлены:

* После итерации 1 (февраль 2025).
* После итерации 2 (апрель 2025).
* Перед сдачей (май 2025).

#### 5.5. План-график

График мероприятий по контролю качества см. в SQAP (итерация 2).



**6. Дополнения**

**6.1 Указатель**

Должен быть представлен.

**6.2 Приложения**

Должны быть представлены.