
Техническое Задание (ТЗ) на создание "ИИ-Проджект Менеджера" (ИИ-ПМ)

Версия документа: 1.0

Дата: 2 июня 2025 г.

1. Общие сведения

- **1.1. Наименование проекта:** "ИИ-Проджект Менеджер (AI-PM) для управления разработкой корпоративной ИИ-платформы" (далее – Система или ИИ-ПМ).
 - **1.2. Заказчик:** Владелец малого бизнеса (далее – Пользователь).
 - **1.3. Исполнитель:** Пользователь / Привлеченные специалисты / Иные ИИ-инструменты под управлением Пользователя.
 - **1.4. Основание для разработки:** Стратегическая задача по переводу компании на качественно более высокий уровень с использованием ИИ-технологий, необходимость эффективного управления проектом создания основной ИИ-платформы.
 - **1.5. Сроки выполнения проекта (MVP):** [Указать желаемые или предполагаемые сроки, например, "4-8 недель"]
-

2. Назначение и цели создания Системы

- **2.1. Назначение Системы:**
ИИ-ПМ предназначен для оказания интеллектуальной поддержки Пользователю в управлении проектом разработки корпоративной ИИ-платформы. Система должна автоматизировать рутинные задачи проектного управления, предоставлять аналитическую поддержку и способствовать повышению эффективности и прозрачности процесса разработки.
- **2.2. Цели создания Системы (MVP):**
 - Снизить временные затраты Пользователя на задачи проектного менеджмента.
 - Обеспечить структурированное планирование и отслеживание задач проекта.
 - Улучшить качество и своевременность отчетности по проекту.
 - Создать централизованную базу знаний по проекту.
 - Обеспечить базовые механизмы для идентификации и отслеживания

рисков проекта.

3. Требования к Системе

3.1. Функциональные требования (MVP)

- 3.1.1. Управление задачами:
 - * 3.1.1.1. Содействие в декомпозиции высокоуровневых целей проекта на конкретные задачи и подзадачи (на основе входных данных от Пользователя и преднастроенных шаблонов).
 - * 3.1.1.2. Помощь в определении приоритетов задач (например, предложение методов приоритизации, таких как MoSCoW, матрица Эйзенхауэра, на основе критериев Пользователя).
 - * 3.1.1.3. Отслеживание статуса выполнения задач (на основе ручного ввода Пользователем или через будущую интеграцию с внешними системами).
 - * 3.1.1.4. Назначение ответственных за задачи (Пользователь).
 - * 3.1.1.5. Установка и отслеживание плановых и фактических сроков выполнения задач.
- 3.1.2. Планирование и контроль:
 - * 3.1.2.1. Генерация черновиков планов проекта и дорожных карт на основе введенных целей, задач и сроков.
 - * 3.1.2.2. Создание и обновление простых временных графиков (например, в текстовом или табличном формате, с возможностью экспорта данных для построения диаграмм Ганта).
 - * 3.1.2.3. Формирование напоминаний Пользователю о предстоящих сроках, ключевых вехах и просроченных задачах.
- 3.1.3. Отчетность и коммуникация:
 - * 3.1.3.1. Генерация периодических (например, ежедневных, еженедельных) отчетов о статусе проекта, включая выполненные задачи, задачи в работе, просроченные задачи и ключевые достижения.
 - * 3.1.3.2. Суммаризация текстовых документов (например, протоколов встреч, описаний требований), предоставленных Пользователем.
- 3.1.4. Управление рисками (базовое):
 - * 3.1.4.1. Содействие в идентификации потенциальных рисков проекта на основе анализа текущего плана, статуса задач и информации от Пользователя.
 - * 3.1.4.2. Ведение простого реестра рисков (наименование риска, описание, возможные последствия, статус – на основе ввода Пользователя).
- 3.1.5. Управление базой знаний проекта:
 - * 3.1.5.1. Хранение и структуризация ключевой информации по проекту (цели,

требования, принятые решения, проектные документы – через загрузку Пользователем или ввод текста).

* 3.1.5.2. Обеспечение поиска и извлечения информации из базы знаний по запросам Пользователя (RAG-подобная функциональность на основе загруженных документов).

3.2. Нефункциональные требования

- **3.2.1. Удобство использования (Usability):** * Интуитивно понятный интерфейс взаимодействия (на начальном этапе – преимущественно текстовый/чат). * Минимальное время на обучение Пользователя для работы с Системой. * Четкие и понятные ответы, подсказки и сообщения об ошибках.
- **3.2.2. Надежность:** * Стабильная работа основных функций Системы. * Корректное сохранение введенных данных.
- **3.2.3. Производительность:** * Время ответа на стандартные запросы Пользователя не должно превышать нескольких секунд (для взаимодействий с LLM).
- **3.2.4. Безопасность:** * Обеспечение конфиденциальности проектных данных, загружаемых в Систему. * При использовании внешних API (LLM, др. сервисы) – безопасная работа с ключами доступа.
- **3.2.5. Модифицируемость и Масштабируемость (для MVP – заложить основы):** * Архитектура должна позволять относительно простое добавление новых функций и интеграций в будущем. * Способность обрабатывать увеличивающийся объем проектных данных (в разумных пределах для MVP).

3.3. Требования к интерфейсу

- **3.3.1. Основной интерфейс:** Конверсионный (диалоговый) интерфейс на естественном языке (русский) через взаимодействие с LLM.
- **3.3.2. Ввод данных:** Возможность ввода текста, загрузки файлов (например, .txt, .md, .pdf, .docx) для пополнения базы знаний и анализа.
- **3.3.3. Вывод данных:** Текстовые ответы, отчеты, списки, структурированные данные (например, JSON для возможного дальнейшего использования).

3.4. Требования к интеграции (MVP)

- **3.4.1.** Взаимодействие с API выбранной LLM – Google Gemini.
- **3.4.2. (Опционально для MVP, желательно для удобства):** Интеграция с облачным хранилищем (Google Drive, Dropbox и т.п) для доступа к проектным файлам.
- **3.4.3. (Перспектива):** Интеграция с системами управления проектами (Jira,

Asana, Trello, Notion), календарями.

3.5. Требования к данным

- **3.5.1. Входные данные:** Текстовые описания задач, целей, рисков, статусов; файлы документов; команды и запросы Пользователя.
- **3.5.2. Хранимые данные:** Структурированная информация о проекте (задачи, сроки, статусы, риски), контент базы знаний, логи взаимодействий (для анализа и улучшения).
- **3.5.3. Резервное копирование:** Пользователь самостоятельно определяет стратегию резервного копирования данных, не хранящихся в облачных сервисах LLM или интегрированных системах.

4. Состав и содержание работ по созданию Системы (MVP)

- **4.1. Этап 1: Анализ и проектирование (срок: [например, 1 неделя]) ***
Уточнение и финализация требований к MVP ИИ-ПМ. * Выбор конкретной LLM и других ключевых технологий/сервисов. * Разработка детальных сценариев использования (use cases). * Проектирование архитектуры ИИ-ПМ (включая структуру промптов, механизм работы с базой знаний).
- **4.2. Этап 2: Разработка MVP (срок: [например, 2-5 недель]) ***
Настройка окружения и подключение к API LLM. * Разработка системы промптов (system prompts, user prompts) для реализации функциональных требований. * Реализация механизма управления базой знаний (загрузка, базовый поиск/RAG). * Создание интерфейса взаимодействия (настройка чат-клиента или использование готовых платформ). * Реализация логики для основных функций: управление задачами, планирование, отчетность, управление рисками.
- **4.3. Этап 3: Тестирование и отладка (срок: [например, 1 неделя]) ***
Внутреннее тестирование разработчиком (Пользователем). * Исправление выявленных ошибок и недочетов.
- **4.4. Этап 4: Ввод в эксплуатацию и документирование (срок: [например, 1 неделя]) ***
Подготовка краткого руководства пользователя по работе с ИИ-ПМ. * Развертывание ИИ-ПМ для использования Пользователем.

5. Порядок контроля и приемки Системы

- **5.1. Виды испытаний:** Функциональное тестирование Пользователем.
- **5.2. Критерии приемки MVP:** * ИИ-ПМ выполняет все заявленные в разделе 3.1 функциональные требования. * Интерфейс взаимодействия соответствует

требованиям раздела 3.3. * Система стабильно работает при выполнении типовых сценариев. * Ответы ИИ-ПМ релевантны запросам и задачам Пользователя (с учетом ограничений текущих LLM).

- **5.3. Процедура приемки:** Пользователь проводит тестирование по заранее согласованным сценариям. По итогам тестирования составляется протокол с перечнем замечаний (при наличии). После устранения критических замечаний Система считается принятой.

6. Требования к документированию

- **6.1. Руководство пользователя ИИ-ПМ (MVP):** * Описание назначения и возможностей Системы. * Инструкции по началу работы и основным операциям. * Рекомендации по формулированию запросов (промптов) для получения наилучших результатов. * Описание известных ограничений Системы.
- **6.2. (Внутреннее) Описание архитектуры и ключевых промптов:** Для возможности дальнейшего развития и модификации Пользователем.

7. Перспективы развития Системы (после MVP)

- **7.1.** Углубленная интеграция с внешними системами управления проектами, календарями, облачными хранилищами.
 - **7.2.** Расширение аналитических возможностей (например, предиктивный анализ сроков, выявление узких мест).
 - **7.3.** Повышение степени автоматизации (например, автоматическое обновление статусов задач из интегрированных систем).
 - **7.4.** Разработка более продвинутого пользовательского интерфейса (возможно, графического дашборда).
 - **7.5.** Внедрение механизмов машинного обучения для персонализации и улучшения рекомендаций ИИ-ПМ.
-