Институт приборостроения, автоматизации и информационных технологий

Кафедра информационных систем и цифровых технологий

Дисциплина «Основы управления проектами»

Отчет к лабораторной работе № 7

«Разработка плана коммуникаций

проекта и плана управления конфигурациями»

Проект: мобильное приложение для уведомления о приеме лекарств и отслеживания метрик здоровья

Выполнил:

Василения Иван Валерьевич

Принял:

Олькина Елена Викторовна

Орёл, 2025г

1. План сбора информации

Источники информации и методы получения:

* Целевая аудитория (пенсионеры и их семьи):
  + Методы:
    - Анкетирование и фокус-группы для выявления потребностей.
    - Тестовые сессии с прототипом приложения для оценки удобства интерфейса.
  + Источники: Социальные центры, медицинские учреждения, онлайн-сообщества.
* Медицинские эксперты:
  + Методы:
    - Консультации для проверки корректности медицинских рекомендаций и метрик (уровень сахара, давление).
    - Верификация данных о лекарствах.
  + Источники: Врачи-терапевты, геронтологи, фармацевты.
* Технические специалисты:
  + Методы:
    - Сбор требований к совместимости с iOS/Android.
    - Анализ энергопотребления и офлайн-функционала.
  + Источники: Разработчики, тестировщики, архитекторы приложения.
* Рыночные данные:
  + Методы:
    - Анализ конкурентов (функционал, UX, цены).
    - Изучение трендов в мобильных приложениях для здоровья.
  + Источники: App Store, Google Play, отчеты аналитических агентств.
* Юридические аспекты:
  + Методы:
    - Консультации с юристами по GDPR и защите персональных данных.
    - Анализ требований к медицинским приложениям.
  + Источники: Юридические фирмы, законодательные базы.
* Инвесторы и партнеры:
  + Методы:
    - Сбор требований к ROI и маркетинговой стратегии.
    - Уточнение бюджетных ограничений.
  + Источники: Встречи с инвесторами, договоры с партнерами.

Потребители информации и способы доставки:

* Разработчики
  + - Ежедневные стендапы (Zoom/Teams).
  + - Jira/Confluence для управления задачами.
* Инвесторы
  + - Ежемесячные PDF-отчеты по email.
  + - Презентации через Zoom (каждые 2 месяца).
* Партнеры (маркетинг)
  + - Совещания в Teams.
  + - Общие документы в Google Drive.
* Пенсионеры (пользователи)
  + - Печатные брошюры (распространяются через соццентры).
  + - Видеоуроки на RuTube.
  + - SMS-оповещения.
* Семьи пенсионеров
  + - Email-рассылки.
  + - Push-уведомления в приложении.

Детальное описание документов:

1. Техническое задание (ТЗ)

* Формат: PDF/Docx (версии в Confluence/Jira).
* Содержание:
  + Функциональные требования (напоминания о лекарствах, мониторинг здоровья, аналитика).
  + Технические спецификации (совместимость с iOS/Android, офлайн-режим, интеграция с API).
  + Требования к безопасности (шифрование данных, аутентификация).
* Уровень детальности: Высокий.
* Определения:
  + "Офлайн-режим" — базовые функции (напоминания, просмотр истории) доступны без интернета.
  + "API интеграция" — подключение к медицинским базам данных и сервисам аналитики.

2. Руководство пользователя

* Формат: PDF/видео (с субтитрами).
* Содержание:
  + Пошаговая инструкция по установке и настройке приложения.
  + Раздел "Часто задаваемые вопросы" (FAQ).
  + Контакты поддержки (телефон, email).
* Уровень детальности: Низкий (простой язык, минимум технических терминов).
* Определения:
  + "Напоминание" — уведомление о приеме лекарств.
  + "Мониторинг здоровья" — ввод данных о давлении и сахаре.

3. Отчет о тестировании

* Формат: Excel/PDF (с графиками).
* Содержание:
  + Результаты тестирования функционала (баги, исправления).
  + Производительность: скорость работы, энергопотребление.
  + Рекомендации по доработке.
* Уровень детальности: Высокий (детали по каждому тест-кейсу).
* Определения:
  + "Критический баг" — ошибка, блокирующая работу приложения.
  + "Сценарий тестирования" — последовательность действий для проверки функции.

4. Политика конфиденциальности

* Формат: PDF (с электронной подписью).
* Содержание:
  + Типы собираемых данных
  + Условия хранения и передачи данных
  + Права пользователей
* Уровень детальности: Средний
* Определения:
  + "Персональные данные" — ФИО, возраст, контакты.

**План ввода в действие выбранных видов коммуникаций**

Этапы реализации:

1. Подготовительный этап (1-2 недели):
   * Настройка инструментов:
     + Создание аккаунтов в Jira/Confluence для управления задачами.
     + Настройка Google Workspace для совместной работы с документами.
     + Регистрация в Mailchimp/SendPulse для рассылок.
   * Обучение команды:
     + Проведение тренингов для разработчиков по работе с Jira.
     + Инструктаж маркетологов по использованию Mailchimp и Google Analytics.
   * Подготовка материалов:
     + Разработка шаблонов финансовых отчетов, руководств пользователя и политики конфиденциальности.
     + Создание видеоуроков для пенсионеров (с субтитрами).
2. Пилотный запуск (2 недели):
   * Тестирование коммуникаций:
     + Проведение пробных стендапов с разработчиками через Zoom.
     + Рассылка тестовых SMS-оповещений и пуш-уведомлений.
     + Распространение печатных брошюр в одном социальном центре для оценки удобства.
   * Сбор обратной связи:
     + Анкетирование участников пилота (пенсионеры, семьи, разработчики).
     + Анализ открываемости и кликабельности рассылок.
3. Полномасштабный запуск (4 недели):
   * Внедрение каналов:
     + Рассылка руководств пользователя через соццентры и RuTube.
     + Запуск чата поддержки в приложении и горячей линии.
   * Интеграция инструментов:
     + Настройка автоматических уведомлений в Firebase (push) и Twilio (SMS).

**Методы обновления и совершенствования плана коммуникаций**

1. Регулярный сбор и анализ обратной связи

* Методы:
  + Опросы пользователей
  + Аналитика данны
* Действия:
  + Корректировка каналов доставки
  + Обновление инструкций и видеоуроков на основе частых вопросов.

2. Адаптация к технологическим изменениям

* Методы:
  + Оценка эффективности текущих платформ и поиск альтернатив.
  + Внедрение автоматизации
  + Интеграция новых каналов
* Действия:
  + Тестирование новых инструментов
  + Обновление технической документации при изменении процессов.

**План управления конфигурацией и версиями проекта**

**Введение**  
План управления конфигурацией и версиями проекта «Мобильное приложение для пенсионеров» разработан для обеспечения контроля над изменениями в программном коде, документации и процессах.

**Конфигурационное управление программным продуктом**

1. Инструменты управления версиями и конфигурацией:
   * Git (GitHub):
   * Jira + Confluence:
   * Docker:

2. Хранение конфигураций:

* + Конфигурационные файлы (настройки БД, API-ключи):
    1. Шаблоны хранятся в Git с суффиксом .template (например, config.yaml.template).
    2. Реальные значения подставляются через переменные окружения или Secrets (GitHub Secrets).
  + Окружения:
    1. dev — локальные машины разработчиков.
    2. staging — тестовый сервер, идентичный продакшену.
    3. production — рабочий сервер.

**Программа конфигурационного управления**

Политики и процедуры

1. Управление изменениями:
   * Подача запроса на изменение:
     + Все изменения (код, документация, настройки) регистрируются в Jira как задачи с типом «Improvement» или «Bug».
     + Запрос должен включать описание, обоснование и оценку влияния на другие компоненты.
   * Утверждение изменений:
     + Изменения в код утверждаются через Pull Request (PR) в GitHub.
     + Для критических изменений (например, обновление ядра приложения) требуется согласование с архитектором проекта.
2. Резервное копирование:
   * Хранение резервных копий не менее 30 дней.

Критерии приемки изменений

1. Для кода:
   * Успешное прохождение CI/CD (сборка + тесты).
2. Для документации:
   * Соответствие шаблонам в Confluence.
   * Согласование с техническим писателем.
3. Для релизов:
   * Тестирование на staging не менее 48 часов.
   * Подписание чек-листа релиза

**Этапы управления конфигурацией**

1. Идентификация конфигурационных единиц

* Что включает:
  + Код приложения
  + Документация: Техническое задание, руководства пользователя, политика конфиденциальности.
  + Настройки: Конфигурационные файлы (API-ключи, настройки БД), Docker-образы.

2. Контроль изменений

* Процедура:
  1. Создание задачи в Jira: Описание изменения, его цели и влияния.
  2. Разработка в изолированной ветке: Например, feature/add-medication-reminder.
  3. Код-ревью: Проверка PR минимум двумя разработчиками.
  4. Интеграция в develop: После успешного CI/CD и тестирования.
  5. Релиз: Мердж develop в main с тегом версии (например, v1.4.0).

3. Инструменты:

* Jira для отслеживания статусов задач.
* GitHub Projects для визуализации этапов.

4. Релиз и развертывание

* Процесс:
  1. Тестирование:
     + Проверка всех функций и интеграций.
     + Мониторинг производительности.
  2. Публикация:
     + Загрузка APK/IPA в App Store и Google Play.
     + Обновление Docker-образов на продакшен-серверах.
  3. Уведомление:
     + Рассылка релиз-нот через email и push-сообщения.
     + Обновление Changelog в GitHub Releases.

5. Откат изменений

* Сценарии:
  + Критический баг: Возврат к предыдущей версии
  + Нестабильность продакшена: Автоматический откат через CI/CD
* Документирование:
  + Причина отката фиксируется в Jira.
  + Информация включается в отчёт о релизе.

**Обучение и ресурсы**

1. Для разработчиков:
   * Git и GitHub:
     + Базовый курс: ветвление, мердж, разрешение конфликтов.
   * CI/CD (GitHub Actions):
     + Настройка пайплайнов, интеграция тестов, деплой на staging/production.
   * Формат:
     + Онлайн-курсы
2. Для тестировщиков:
   * Автоматизированное тестирование:
     + Написание скриптов на Python/JavaScript для интеграционных тестов.
   * Работа с Jira:
     + Создание тест-кейсов, привязка к задачам, отчеты о багах.
   * Формат:
     + Онлайн-курсы