**Вариант 8**

1) Суть проекта:

Разработка и внедрение нового мобильного приложения для онлайн-бронирования и оплаты услуг в сфере туризма.

2) Основные ограничения:

- Бюджет: до 1,5 миллионов рублей.

- Сроки: не более года с момента начала разработки.

- Качество: стабильная работа и высокая производительность приложения, соответствие требованиям Google Play и App Store.

- Ресурсы: ограниченный штат сотрудников компании-исполнителя и ограниченное использование подрядчиков.

- Технологические ограничения: приложение должно быть совместимо с операционными системами Android и iOS.

3) Описание заказчика, его пожеланий:

Заказчик - туристическое агентство Х, которое предоставляет услуги по бронированию туров и отелей. Они хотят создать мобильное приложение, которое позволит клиентам удобно и быстро выбирать и оплачивать туры и отели, а также получать актуальную информацию о доступных предложениях и акциях.

Заказчик желает, чтобы приложение включало функции поиска и фильтрации туров и отелей, онлайн-оплаты, отправки подтверждений и электронных билетов на почту клиента, а также возможность оставлять отзывы о поездке.

4) Описание компании-исполнителя:

Компания-исполнитель является разработчиком мобильных приложений и имеет опыт работы в сфере туризма. У компании есть успешные проекты по разработке и поддержке мобильных приложений для туристических агентств и отелей. Она обладает необходимыми ресурсами и опытом для выполнения данного проекта.

5) Состав проектной команды:

- Руководитель проекта - 1 человек, опыт работы в управлении проектами в сфере разработки мобильных приложений - не менее 3 лет.

- Аналитик - 1 человек, опыт работы в анализе и документировании требований для мобильных приложений - не менее 2 лет.

- Разработчики - 2 человека, опыт работы в разработке мобильных приложений для Android и iOS - не менее 2 лет.

- Дизайнер - 1 человек, опыт работы в разработке пользовательских интерфейсов мобильных приложений - не менее 2 лет.

- Тестировщик - 1 человек, опыт работы в тестировании мобильных приложений - не менее 2 лет.

6) Особенности среды выполнения проекта:

- Необходимость интеграции с базой данных туров и отелей компании Х.

- Зависимость от сторонних API для получения актуальной информации о турах и отелях.

- Возможное влияние изменений требований со стороны заказчика на процесс разработки и внедрения приложения.

- Необходимость соблюдения регуляторных требований в области обработки и хранения персональных данных клиентов.

**1**) *Дополните легенду вашего проекта, если посчитаете необходимым.*

* Монетизация приложения: Определить, как туристическое агентство Х будет получать доход через приложение. Возможно, это будут комиссии с бронирований, платные подписки для получения эксклюзивных предложений, или рекламные интеграции с отелями и турагентами.
* Функция персонализации: Включить возможность персонализированных рекомендаций для пользователей на основе их предыдущих бронирований, интересов или данных о предпочтениях.
* Поддержка нескольких языков: Учитывая глобальный характер туристических услуг, приложение может поддерживать несколько языков для пользователей из разных регионов.

2) *К какой системе модели Кеневин относится ваш проект? Почему? Какие условия должны измениться, чтобы можно было причислить его к другому типу?*

Проект по разработке мобильного приложения для туризма можно отнести к **сложной системе** по модели Кеневин. Вот основные причины:

* **Множество взаимосвязанных элементов**: Приложение должно интегрироваться с базой данных туров и отелей, сторонними API для получения актуальной информации, системами онлайн-оплаты и электронной почтой для подтверждений. Все эти элементы должны работать синхронно, что усложняет проект.
* **Неопределенность и возможные изменения**: Требования могут изменяться по ходу проекта, особенно с учетом пожеланий заказчика или внешних факторов (например, изменения в туристической индустрии или законодательстве по защите данных).
* **Требуется экспериментация**: Не все решения можно предугадать с самого начала, так как нужно протестировать функциональность, убедиться в надежной интеграции сторонних сервисов и добиться высоких показателей производительности.

Условия для перехода в другую категорию:

* **Простая система**: Проект можно было бы отнести к простой системе, если бы все требования и функциональность были четко определены с самого начала, интеграция с сервисами была бы стандартной и предсказуемой, а также не предполагалось бы никаких изменений в процессе разработки.
* **Осложненная или хаотическая система**: Если бы проект включал новые, ранее неиспользуемые технологии или неопределенные факторы (например, инновационные решения для работы с данными клиентов), его можно было бы отнести к усложненной или даже хаотической системе. В таком случае потребовалось бы применять быстрые эксперименты и адаптацию для нахождения решений.

3) *Ориентируясь на сформулированную легенду, оцените проект по параметрам модели Agile Suitability Model. Результат нарисуйте на графике*.

Для оценки проекта по параметрам Agile Suitability Model, можно рассмотреть следующие аспекты:

 **Культура:**

* **Поддержка:** 7/10. Заказчик поддерживает гибкие подходы, но есть ограниченные возможности для изменений в критичных аспектах.
* **Доверие:** 8/10. Заказчик доверяет команде и открыт к обсуждению предложений, касающихся улучшения продукта.
* **Решения:** 7/10. Команда может самостоятельно принимать некоторые решения, но важные изменения требуют согласования с заказчиком.

 **Команда:**

* **Размер команды:** 1/10 (небольшая команда).
* **Опыт:** 8/10. В команде достаточно опытных специалистов для выполнения задач.
* **Доступ:** 5/10. Ежедневный контакт с заказчиком возможен, но решения могут задерживаться из-за необходимости согласования.

 **Проект:**

* **Изменения:** 7/10. Возможны умеренные изменения в процессе разработки.
* **Критичность:** 5/10. Ошибки могут привести к задержкам и финансовым потерям, но не катастрофическим последствиям.
* **Поставка:** 6/10. Продукт можно разделить на несколько итераций, с постепенным добавлением функциональности и возможностью получения обратной связи.

4) *На основе предыдущих ответов определите, какая модель ЖЦ подошла бы лучше всего в вашей ситуации. Возможны ли альтернативы?*

Для проекта по разработке мобильного приложения для туризма, лучше всего подойдет **инкрементная модель жизненного цикла**. Вот почему:

* **Ограниченные сроки и бюджет**: Инкрементная модель позволяет разделить проект на несколько частей и реализовывать каждую поочередно. Это удобно в условиях жестких временных и финансовых ограничений, так как функциональность приложения может постепенно расширяться, и каждая версия будет добавлять новые возможности.
* **Изменения требований**: Проект зависит от возможных изменений со стороны заказчика. Инкрементная модель хорошо подходит для адаптации к таким изменениям — вы можете гибко добавлять новые функции или модифицировать существующие.
* **Интеграция с базами данных и сторонними API**: Процесс интеграции может быть сложным и требует тестирования на каждом этапе. Инкрементный подход позволяет вводить интеграции поэтапно, что снижает риск критических ошибок.

Возможные альтернативы:

**Agile**: Если изменения требований со стороны заказчика будут частыми, Agile может быть предпочтительным, обеспечивая более тесное взаимодействие и быструю адаптацию. Но это потребует высокой степени вовлеченности заказчика.

**V-Model**: Если требования точно определены, и необходимо тщательное тестирование на каждом этапе (особенно из-за требований к обработке персональных данных), V-Model может помочь, но это менее гибкий подход, чем инкрементная модель.

**Подход к декомпозиции**

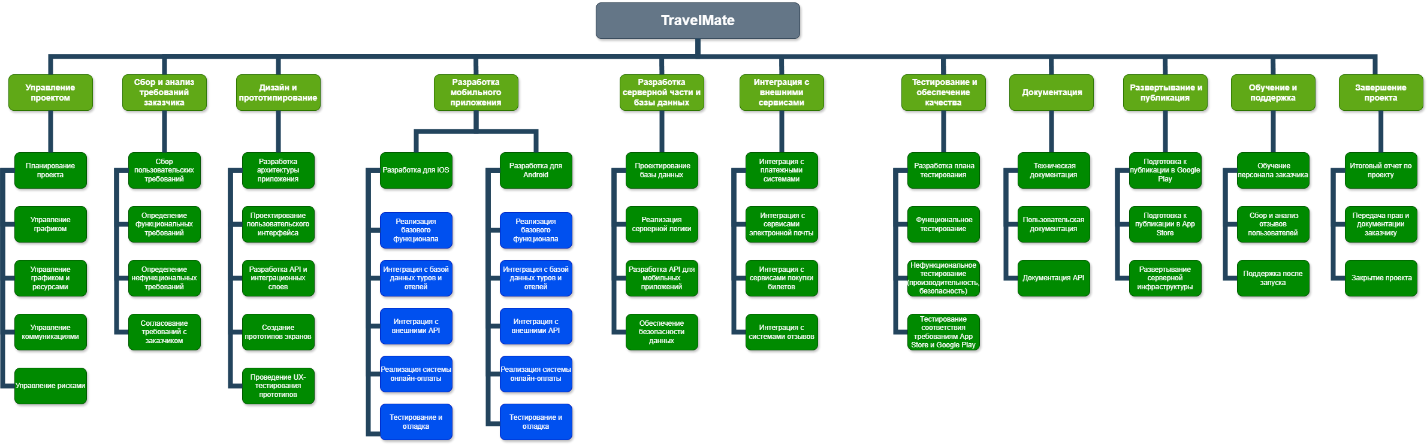
Для данного проекта выбран **продуктовый подход к декомпозиции**. Этот подход предполагает разбивку проекта на основные компоненты конечного продукта — мобильного приложения для онлайн-бронирования и оплаты туристических услуг. Такой метод позволяет сосредоточиться на создании конкретных функциональных модулей приложения, обеспечивая целостность и качество разработки каждой части.

Обоснование выбранного подхода к декомпозиции

**Продуктовый подход** к декомпозиции выбран, потому что он:

* **Сосредоточен на конечном продукте**: Позволяет команде четко видеть, какие компоненты приложения должны быть разработаны.
* **Улучшает управление проектом**: Облегчает планирование, распределение задач и контроль выполнения работ.
* **Облегчает коммуникацию с заказчиком**: Позволяет наглядно демонстрировать прогресс по отдельным функциональным модулям.
* **Гибок к изменениям**: Позволяет вносить корректировки в отдельные части продукта без существенного влияния на весь проект.

3) Иерархическая структура работ (WBS)



Словарь ИСР

**1. Управление проектом**

* **Планирование проекта**: Разработка детального плана, определение целей, задач, сроков и ресурсов проекта.
* **Управление графиком и ресурсами**: Контроль выполнения работ согласно графику, распределение задач между членами команды.
* **Управление бюджетом**: Контроль затрат, обеспечение соответствия расходов установленному бюджету.
* **Управление коммуникациями**: Организация эффективного взаимодействия между командой и заказчиком, проведение совещаний, отчетность.
* **Управление рисками**: Идентификация рисков, оценка вероятности и влияния, разработка планов реагирования.

**2. Анализ требований**

* **Сбор и анализ требований заказчика**: Проведение встреч с заказчиком для определения потребностей и ожиданий.
* **Определение функциональных требований**: Формирование списка функций, которые должно выполнять приложение.
* **Определение нефункциональных требований**: Определение требований к производительности, безопасности, совместимости и др.
* **Согласование требований с заказчиком**: Утверждение всех требований и получение одобрения на дальнейшую разработку.

**3. Дизайн и прототипирование**

* **Разработка архитектуры приложения**: Определение технической структуры приложения, выбор технологий и платформ.
* **Проектирование пользовательского интерфейса (UI/UX)**: Создание дизайна экранов, навигации и взаимодействия с пользователем.
* **Создание прототипов экранов**: Разработка интерактивных макетов для визуализации работы приложения.
* **Проведение UX-тестирования прототипов**: Оценка удобства использования, сбор обратной связи от потенциальных пользователей.

**4. Разработка мобильного приложения**

**4.1 Разработка версии для Android**

* **Реализация базового функционала**: Создание основных функций приложения, включая поиск и фильтрацию туров и отелей.
* **Интеграция с базой данных туров и отелей**: Обеспечение доступа к актуальной информации о предложениях.
* **Интеграция с внешними API**: Подключение к сторонним сервисам для получения дополнительной информации.
* **Реализация системы онлайн-оплаты**: Внедрение безопасных платежных методов.
* **Тестирование и отладка**: Проверка работоспособности и устранение ошибок.

**4.2 Разработка версии для iOS**

* **Реализация базового функционала** **платформы iOS.**
* **Интеграция с базой данных туров и отелей**
* **Интеграция с внешними API**
* **Реализация системы онлайн-оплаты**
* **Тестирование и отладка**

**5. Разработка серверной части и базы данных**

* **Проектирование базы данных**: Создание структуры базы данных для хранения информации о турах, отелях и пользователях.
* **Реализация серверной логики**: Разработка серверных приложений для обработки запросов от мобильных клиентов.
* **Разработка API для мобильных приложений**: Создание интерфейсов для взаимодействия мобильных приложений с сервером.
* **Обеспечение безопасности данных**: Внедрение мер защиты персональных данных и финансовой информации.

**6. Интеграция с внешними сервисами**

* **Интеграция с платежными системами**: Подключение к платежным шлюзам для обеспечения онлайн-оплаты.
* **Интеграция с сервисами электронной почты**: Настройка автоматической отправки подтверждений и билетов клиентам.
* **Интеграция с системами отзывов**: Внедрение функциональности для оставления и просмотра отзывов о поездках.

**7. Тестирование и обеспечение качества**

* **Разработка плана тестирования**: Определение видов тестирования, необходимых для обеспечения качества.
* **Функциональное тестирование**: Проверка работы всех функций приложения в соответствии с требованиями.
* **Нефункциональное тестирование**: Тестирование производительности, безопасности и совместимости приложения.
* **Тестирование соответствия требованиям App Store и Google Play**: Проверка приложения на соответствие политикам публикации.
* **Пользовательское приемочное тестирование**: Проведение тестирования с участием реальных пользователей для подтверждения готовности приложения.

**8. Документация**

* **Техническая документация**: Создание подробной документации для разработчиков и технического персонала.
* **Пользовательская документация**: Подготовка инструкций и руководств для пользователей приложения.
* **Документация API**: Описание методов и интерфейсов для взаимодействия с серверной частью.

**9. Развертывание и публикация**

* **Подготовка к публикации в Google Play**: Создание учетной записи разработчика, подготовка материалов и загрузка приложения.
* **Подготовка к публикации в App Store**: Аналогично пункту 9.1, но для платформы iOS.
* **Развертывание серверной инфраструктуры**: Настройка серверов и сервисов, необходимых для работы приложения.

**10. Обучение и поддержка**

* **Обучение персонала заказчика**: Проведение тренингов по использованию и администрированию приложения.
* **Поддержка после запуска**: Обеспечение технической поддержки, устранение возникающих проблем.
* **Сбор и анализ отзывов пользователей**: Мониторинг обратной связи для улучшения приложения.

**11. Завершение проекта**

* **Итоговый отчет по проекту**: Подготовка итоговой документации о выполненной работе.
* **Передача прав и документации заказчику**: Официальная передача всех материалов и прав на приложение.
* **Закрытие проекта**: Окончательное завершение всех работ и формальностей.

3. Риски

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Риск** | **Вероятность** | **Влияние** | **Стратегия реагирования** |
| 1 | **Изменение требований заказчика в процессе разработки** | Высокая | Среднее | **Смягчение**: Регулярные встречи с заказчиком, гибкое планирование, использование Agile-методологий |
| 2 | **Задержки в интеграции с внешними API** | Средняя | Высокое | **Смягчение**: Раннее начало интеграции, разработка резервных планов, тесное взаимодействие с поставщиками API. |
| 3 | **Несоответствие требованиям App Store и Google Play** | Средняя | Высокое | **Смягчение**: Изучение требований магазинов заранее, внутреннее тестирование на соответствие, корректировка приложения. |
| 4 | **Проблемы с безопасностью данных пользователей**. | Низкая | Высокое | **Смягчение**: Внедрение лучших практик безопасности, регулярные аудиты, обучение команды. |
| 5 | **Превышение бюджета проекта** | Средняя | Высокое | **Смягчение**: Тщательное бюджетирование, регулярный мониторинг затрат, приоритизация задач. |
| 6 | **Срыв сроков проекта** | Средняя | Высокое | **Смягчение**: Реалистичное планирование, контроль выполнения задач, возможность перераспределения ресурсов. |
| 7 | **Недостаточный опыт команды в новых технологиях** | Низкая | Среднее | **Смягчение**: Обучение сотрудников, привлечение экспертов, использование проверенных технологий. |
| 8 | **Проблемы с производительностью приложения** | Средняя | Среднее | **Смягчение**: Оптимизация кода, проведение нагрузочного тестирования, использование эффективных инструментов. |
| 9 | **Изменения в законодательстве по персональным данным** | Низкая | Высокое | **Принятие**: Мониторинг законодательства, готовность к оперативным изменениям, консультации с юристами. |
| 10 | **Сбои в работе платежных систем** | Средняя | Высокое | **Смягчение**: Интеграция с несколькими платежными системами, регулярное тестирование, наличие офлайн-режима оплаты. |
| 11 | **Низкая удовлетворенность пользователей приложением** | Средняя | Средняя | **Смягчение**: Проведение UX-тестирования, сбор обратной связи, регулярные обновления. |
| 12 | **Уход ключевых сотрудников из команды** | Низкая | Высокое | **Смягчение**: Создание комфортных условий работы, мотивация персонала, документирование процессов. |
| 13 | **Сбои в серверной инфраструктуре** | Низкая | Высокое | **Смягчение**: Использование надежных хостингов, настройка резервирования и мониторинга. |
| 14 | **Превышение расходов на использование сторонних API** | Средняя | Среднее | **Смягчение**: Оптимизация использования API, переговоры с поставщиками, поиск альтернатив. |
| 15 | **Появление сильных конкурентов на рынке** | Средняя | Среднее | **Смягчение**: Мониторинг рынка, внедрение уникальных функций, улучшение пользовательского опыта. |

Пояснения к стратегиям реагирования

**Смягчение**: Активные действия для снижения вероятности или влияния риска.

**Принятие**: Признание риска без активных действий, готовность реагировать при его реализации.

**Стратегии реагирования на риски**

**1. Изменение требований заказчика в процессе разработки**

**Вероятность**: Высокая

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Регулярные встречи с заказчиком**: Организовать еженедельные или двухнедельные встречи для обсуждения прогресса, получения обратной связи и своевременного выявления новых требований.

**Использование Agile-методологий**: Применить гибкие методологии разработки (например, Scrum или Kanban), позволяющие быстро адаптироваться к изменениям и эффективно управлять приоритетами задач.

**Документирование изменений**: Все изменения фиксировать в реестре изменений с указанием их влияния на сроки и бюджет. Это позволит прозрачнее управлять ожиданиями и ресурсами.

**Приоритизация функциональности**: Совместно с заказчиком определить критически важные функции и концентрироваться на их реализации в первую очередь.

**2. Задержки в интеграции с внешними API**

**Вероятность**: Средняя

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Раннее начало интеграции**: Планировать интеграцию с внешними API на ранних этапах проекта, чтобы выявить возможные проблемы заранее.

**Тесное взаимодействие с поставщиками API**: Установить контакт с технической поддержкой поставщиков API для оперативного решения возникающих вопросов.

**Разработка резервных планов**: Иметь альтернативные варианты или запасные API на случай недоступности или нестабильности основных сервисов.

**Мониторинг обновлений API**: Регулярно отслеживать изменения в документации и версиях API, чтобы своевременно адаптироваться к ним.

**3. Несоответствие требованиям App Store и Google Play**

**Вероятность**: Средняя

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Изучение требований и руководств**: Детально ознакомиться с последними требованиями и рекомендациями для разработчиков от Apple и Google.

**Промежуточные проверки**: Проводить внутренние аудиты приложения на соответствие требованиям магазинов на этапах разработки.

**Тестирование на соответствие**: Использовать автоматизированные инструменты для проверки приложения на соответствие политикам публикации.

**Быстрая реакция на замечания**: В случае отклонения приложения оперативно вносить необходимые исправления и повторно подавать его на рассмотрение.

**4. Проблемы с безопасностью данных пользователей**

**Вероятность**: Низкая

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Внедрение передовых практик безопасности**: Использовать SSL/TLS для шифрования данных, безопасное хранение паролей (например, хеширование с солью), внедрять двухфакторную аутентификацию.

**Проведение регулярных аудитов**: Заказывать внешние аудиты безопасности и проводить внутренние проверки кода на наличие уязвимостей.

**Обучение команды**: Проводить тренинги по безопасной разработке и обработке персональных данных.

**Разработка плана реагирования на инциденты**: Создать протокол действий на случай утечки данных или взлома, включая уведомление пользователей и соответствующих органов.

**5. Превышение бюджета проекта**

**Вероятность**: Средняя

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Детальное планирование бюджета**: Разработать подробный бюджет с разбивкой по задачам и этапам, включая резерв на непредвиденные расходы.

**Регулярный контроль затрат**: Ежемесячно сравнивать фактические расходы с запланированными и анализировать отклонения.

**Приоритетность задач**: Фокусироваться на наиболее важных функциях, которые приносят максимальную ценность заказчику.

**Оптимизация ресурсов**: Рассмотреть возможность использования более экономичных инструментов или технологий, оптимизировать процессы разработки.

**6. Срыв сроков проекта**

**Вероятность**: Средняя

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Реалистичное планирование**: При составлении графика учитывать возможные риски и дополнительные буферные времена.

**Ежедневные встречи команды**: Использовать ежедневные стендапы для обсуждения прогресса и препятствий.

**Раннее выявление отклонений**: Использовать инструменты управления проектами для мониторинга выполнения задач.

**Перераспределение ресурсов**: При необходимости перенаправлять усилия команды на критические задачи, возможно привлечение дополнительных ресурсов.

**7. Недостаточный опыт команды в новых технологиях**

**Вероятность**: Низкая

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Инвестирование в обучение**: Организовать курсы, вебинары или самостоятельное изучение необходимых технологий.

**Наставничество**: Назначить более опытных сотрудников в качестве менторов для менее опытных.

**Привлечение внешних специалистов**: Если необходимо, временно привлечь экспертов для выполнения сложных задач или проведения обучения команды.

**8. Проблемы с производительностью приложения**

**Вероятность**: Средняя

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Оптимизация архитектуры**: С самого начала проектировать приложение с учетом производительности, выбирать эффективные алгоритмы и структуры данных.

**Профилирование и мониторинг**: Использовать инструменты для измерения производительности и выявления узких мест.

**Кэширование данных**: Внедрить кэширование часто используемых данных для снижения нагрузки на серверы и ускорения работы приложения.

**Тестирование на реальных устройствах**: Проводить тесты на различных моделях устройств для оценки производительности в реальных условиях.

**9. Изменения в законодательстве по персональным данным**

**Вероятность**: Низкая

**Влияние**: Высокое

**Стратегия реагирования**: **Принятие**

**План действий:**

**Мониторинг изменений**: Регулярно отслеживать новости и публикации о законодательных изменениях в сфере защиты данных.

**Консультации с юристами**: Наладить сотрудничество с юридическими специалистами, чтобы своевременно получать рекомендации.

**Гибкость системы**: Проектировать приложение таким образом, чтобы изменения в обработке данных могли быть внедрены с минимальными усилиями.

**10. Сбои в работе платежных систем**

**Вероятность**: Средняя

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Интеграция с несколькими платежными системами**: Обеспечить альтернативные способы оплаты для пользователей.

**Автоматическое переключение**: В случае недоступности одного платежного шлюза автоматически предлагать другой.

**Оповещение пользователей**: Информировать пользователей о проблемах с оплатой и предлагать отложить платеж или выбрать другой метод.

**Тестирование отказоустойчивости**: Проводить регулярные проверки на устойчивость системы к сбоям платежных сервисов.

**11. Низкая удовлетворенность пользователей приложением**

**Вероятность**: Средняя

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Анализ пользовательского опыта**: Регулярно собирать и анализировать отзывы пользователей, проводить опросы.

**Итеративное улучшение**: На основе обратной связи вносить изменения и улучшения в приложение.

**Внедрение системы поддержки**: Предоставить пользователям возможность легко связаться с поддержкой для решения проблем.

**Обновления и исправления**: Регулярно выпускать обновления с новыми функциями и улучшениями.

**12. Уход ключевых сотрудников из команды**

**Вероятность**: Низкая

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Создание благоприятной атмосферы**: Поощрять командную работу, обеспечивать комфортные условия труда и возможности профессионального роста.

**Диверсификация навыков**: Обучать нескольких сотрудников ключевым навыкам, чтобы уменьшить зависимость от одного человека.

**План преемственности**: Иметь план на случай ухода сотрудника, включая потенциальных кандидатов на замену.

**Регулярное общение**: Поддерживать открытый диалог с сотрудниками, чтобы вовремя выявлять возможные проблемы или неудовлетворенность.

**13. Сбои в серверной инфраструктуре**

**Вероятность**: Низкая

**Влияние**: Высокое

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Использование облачных сервисов**: Разместить серверную часть на надежных облачных платформах с гарантированным уровнем доступности.

**Резервирование и отказоустойчивость**: Настроить системы автоматического переключения на резервные серверы в случае сбоя основных.

**Мониторинг систем**: Внедрить инструменты для круглосуточного мониторинга работы серверов и быстрого обнаружения проблем.

**План восстановления после сбоев**: Разработать и протестировать план действий на случай критических сбоев.

**14. Превышение расходов на использование сторонних API**

**Вероятность**: Средняя

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Оптимизация использования API**: Минимизировать количество запросов, используя кэширование и эффективные алгоритмы.

**Анализ тарифных планов**: Изучить различные тарифы поставщиков API и выбрать наиболее подходящий.

**Переговоры с поставщиками**: Обсудить возможность получения скидок или индивидуальных условий.

**Поиск альтернатив**: Рассмотреть другие API с более выгодными условиями или возможности собственной реализации необходимых функций.

**15. Появление сильных конкурентов на рынке**

**Вероятность**: Средняя

**Влияние**: Среднее

**Стратегия реагирования**: **Смягчение**

**План действий:**

**Мониторинг рынка**: Регулярно отслеживать действия конкурентов, анализировать их предложения и стратегии.

**Дифференциация продукта**: Разрабатывать уникальные функции и сервисы, которые выделят приложение на фоне конкурентов.

**Улучшение качества сервиса**: Фокусироваться на высоком уровне обслуживания и поддержке клиентов.

**Маркетинговые кампании**: Проводить активные маркетинговые мероприятия, использовать социальные сети и другие каналы продвижения.

**Партнерства**: Заключать соглашения с другими компаниями для расширения возможностей приложения и привлечения новых пользователей.

Общие рекомендации по управлению рисками

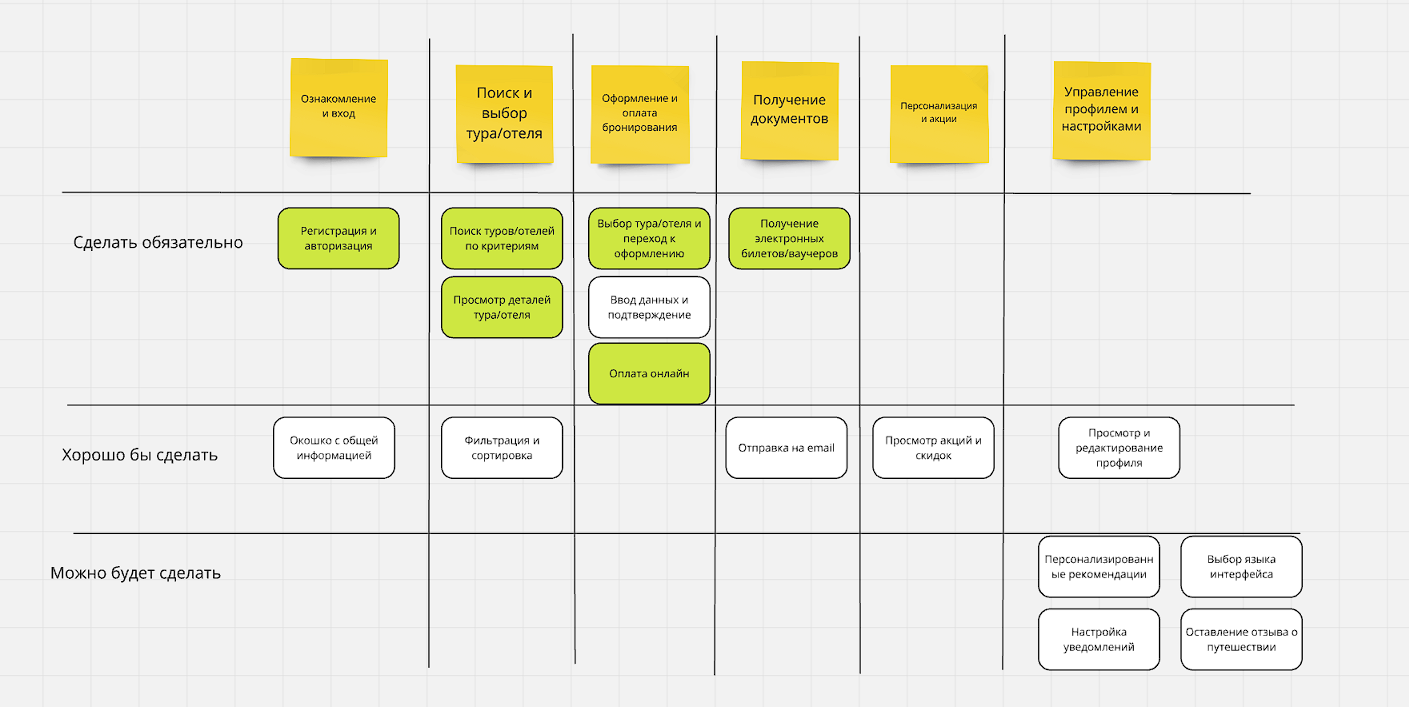
**Постоянный мониторинг рисков**: Регулярно обновлять реестр рисков, отслеживать изменения в их вероятности и влиянии.

**Вовлечение команды**: Поощрять открытое обсуждение рисков внутри команды, привлекать сотрудников к разработке стратегий реагирования.

**Коммуникация с заказчиком**: Поддерживать прозрачные отношения с заказчиком, своевременно информировать о потенциальных рисках и совместно искать решения.

**Гибкость и адаптивность**: Быть готовыми вносить изменения в планы и процессы в ответ на новые риски или изменения в существующих.

**1, 2) Бэклог и приоритизация**



User Story:

(1.1) Как пользователь, **я хочу увидеть краткое описание функционала** приложения, чтобы понять его преимущества.  
(1.2) Как пользователь, **я хочу зарегистрироваться (email/пароль**) для создания личного кабинета.  
 (1.2) Как пользователь, **я хочу авторизоваться с помощью email/пароля**, чтобы получить доступ к моим данным о бронированиях.  
(1.2) Как пользователь, **я хочу восстановить пароль**, чтобы при необходимости получить доступ к аккаунту.

(2.1) Как пользователь, **я хочу искать туры и отели по дате, стране, цене**, чтобы найти подходящий вариант.  
(2.2) Как пользователь, **я хочу фильтровать и сортировать результаты** (по цене, рейтингу), чтобы быстрее найти оптимальное предложение.  
(2.3) Как пользователь, **я хочу просмотреть детальную информацию о туре/отеле (описание, фото, отзывы)**, чтобы принять решение о бронировании.

(3.1) Как пользователь, **я хочу выбрать конкретный тур/отель из результатов поиска и перейти к оформлению**, чтобы забронировать его.  
(3.2) Как пользователь**, я хочу ввести данные о себе (имя, контакты, паспортные данные) и подтвердить бронирование**.  
(3.3) Как пользователь, **я хочу оплатить тур/отель через приложение (интеграция с платежной системой)**, чтобы сразу завершить покупку.

(4.1) Как пользователь, **я хочу получить электронный билет/ваучер после оплаты**, чтобы иметь подтверждение бронирования.  
(4.2) Как пользователь**, я хочу получить билеты на email**, чтобы они были всегда под рукой.

(5.1) Как пользователь**, я хочу видеть актуальные акции и скидки**, чтобы выбрать более выгодное предложение.  
(5.2) Как пользователь, **я хочу получать персонализированные рекомендации** на основе моих предыдущих бронирований и интересов, чтобы экономить время на поиске.

(6.1) Как пользователь**, я хочу изменить личные данные** (имя, контактный email), чтобы поддерживать актуальность информации.  
(6.2) Как пользователь, **я хочу настраивать уведомления** (о скидках, статусе бронирования), чтобы получать актуальные сведения в удобной форме.  
(6.3) Как пользователь, **я хочу оставлять отзывы о прошлых поездках**, чтобы поделиться опытом с другими пользователями.

**3. Оценка элементов бэклога:**

Метод парных сравнений:

1. **Выбор опорной истории:**  
   Сначала нужно выбрать историю, которую команда считает условным «эталоном» по сложности. Обычно это история средней сложности, которая будет служить точкой отсчёта.

* **Метод парных сравнений:**  
  После того как есть опорная история, каждую следующую историю сравнивают с ней по критерию сложности:
  + Сложнее или легче, чем регистрация/авторизация?
  + Насколько сложнее или легче?  
    Оценивают не в точных часах, а в относительных величинах. Если история чуть проще, можно дать ей 2 SP. Если кажется такой же, то тоже 3 SP. Если заметно сложнее, может быть 5 или даже 8 SP.  
    Команда при этом учитывает:
  + Объём работы (количество экранов, форм, необходимость валидации данных)
  + Неопределённость (насколько понятны требования, есть ли открытые вопросы)
  + Интеграции (с третьими системами, API, платёжными шлюзами)
  + Риски и потенциальные сложности

1. **Пример применения:**
   * **Регистрация/Авторизация:**  
     Команда обсуждает: нужно сделать экран регистрации, ввод email/пароля, валидация, запрос к бэкенду, получение токена. Всё достаточно стандартно.

***Решают оценить в 3 SP.***

* + **Просмотр деталей тура/отеля:**  
    Пользователь выбирает тур и видит детальную информацию: описание, фото, отзывы. Данные уже приходят с backend, требуется отобразить их на экране. По сложности это кажется чуть проще, чем поиск (нет логики фильтрации, просто отображение деталей), но примерно на уровне регистрации, может даже чуть проще.

***Решают оценить в 3 SP.***

* + **Оставление отзывов:**  
    Форма, отправка на сервер, возможно модерация. Похоже на редактирование профиля или регистрацию

***Решают оценить в 3 SP.***

* + **Отправка билетов на email:** Это похожая логика на получение электронных билетов, но уже сделали часть этого функционала. Вероятно 3 SP, как и получение билетов.

***Решают оценить в 3 SP.***

* + **Просмотр акций и скидок:** Простой запрос на сервер для получения списка акций, отображение на экране. Похоже по сложности на просмотр деталей (3 SP) или даже проще.

***Решают оценить в 3 SP.***

* + **Редактирование профиля:**  
    Форма, возможность изменить данные, отправить на сервер. Похоже на логику регистрации, но данных может быть чуть больше. Однако требований жестких нет.

***Решают оценить в 3 SP.***

* + **Получение электронных билетов:**  
    После успешной оплаты нужно получить PDF или ссылку на билет, выслать на email. В целом, это запрос к API + отправка email. Сложнее ли это, чем регистрация? Примерно так же или чуть проще, т.к. логика понятна.

***Решают оценить в 3 SP.***

* + **Поиск туров/отелей:**  
    Нужно реализовать форму поиска, фильтр по параметрам (дата, страна, цена), сделать запрос на сервер, получить результаты и отобразить их. Есть интеграция с внешней БД и API, немного сложнее, чем регистрация, но не чрезмерно.

***Решаю оценить в 5 SP.***

* + **Фильтрация и сортировка:**  
    Нужно настроить дополнительную логику на клиенте (а возможно и на сервере). Сравнивая с поиском (5 SP), фильтрация добавляет новую логику, но не настолько сложна, чтобы быть 8 SP.

***Решают оценить в 5 SP.***

* + **Настройка уведомлений:** Нужно добавить экраны, где пользователь выбирает типы уведомлений, сохранить настройки на сервере или локально.

***Решаю оценить в 5 SP.***

* + **Дополнительные языки интерфейса:** Нужно обеспечить локализацию, перевод всех текстов. Это может быть более сложно, чем простые истории, но не настолько, как оплата или рекомендации.

***Решаю оценить в 5 SP.***

* + **Расширенные фильтры:** Уже были фильтры и сортировка (5 SP). Расширенные фильтры (например, сложные условия) могут быть на том же уровне или чуть сложнее. Если требуется много логики, можно дать 5 SP, так как это не настолько сложно, чтобы прыгать к 8.

***Решаю оценить в 5 SP.***

* + **Оформление и оплата бронирования:**  
    Здесь нужно не только собрать данные пользователя, но и интегрироваться с платёжным сервисом, обработать сценарии успеха/ошибок, возможно, добавить защиту данных. Это кажется значительно сложнее, чем регистрация (которая была 3 SP). Интеграция с платёжной системой и сложная логика валидации данных — это большой риск.

***Решают оценить в 8 SP.***

* + **Персонализированные рекомендации:** Сложная логика, возможно нужно анализировать историю бронирований, соединяться с рекомендационным сервисом. Неопределённость высока. Это больше, чем поиск.

***Решают оценить в 8 SP.***

**4. Нулевая итерация:**

* Размер итерации: 2 недели, 5 дней в неделю, по 8 часов = 80 часов на человека.
* Команда разработчиков: 2 человека.  
  Итого: 2 разработчика × 80 часов = 160 человеко-часов. Но часть времени пойдет на настройку окружения, CI/CD, координацию, дизайн. Допустим, на разработку функционала отведём примерно половину — 80 человеко-часов чисто под разработку функционала.  
  Предположим, что за первую итерацию команда сможет выполнить 8–10 SP (с учётом настройки инфраструктуры).

**Цель нулевой итерации:**  
Предоставить работающий «каркас» приложения, в котором можно:

* Зарегистрироваться/авторизоваться
* Выполнить базовый поиск туров/отелей
* Просмотреть детали выбранного варианта

Таким образом, пользователь получит базовый функционал, демонстрирующий основную ценность — поиск предложений.

**Выбор историй для нулевого спринта:**

* Регистрация/Авторизация (3 SP)
* Поиск туров/отелей (5 SP)
* Просмотр деталей тура/отеля (3 SP)

Общий объем: 3 + 5 + 3 = 11 SP.

11 SP — чуть выше предполагаемых 8–10, но можно скорректировать, например, взять только поиск (5 SP) и регистрацию/авторизацию (3 SP), что даст 8 SP, или постараться выполнить все три истории, учитывая, что часть времени уйдет на инфраструктуру.

**Итоговый бэклог нулевой итерации:**

* Регистрация/Авторизация: 3 SP
* Поиск туров/отелей: 5 SP

Итого: 8 SP. Если команда справится с этим раньше, можно попытаться добавить просмотр деталей тура/отеля (ещё 3 SP).

**Цель нулевой итерации:**  
Имеется работоспособный прототип, позволяющий пользователю зарегистрироваться, войти в систему и выполнить базовый поиск туров/отелей. Это создаёт основу для дальнейшего расширения функционала в следующих итерациях.