1. **Тема для разработки веб-сайта**

Приложение для поиска местных фермеров и рынков.

1. **Предметная область**

*Заказчик:*

* Союз фермеров Ленинградской области и Санкт-Петербурга.

*Сроки:*

* Ввод в эксплуатацию первичной версии продукта в неполном функционале в течение 1 года с момента начала разработки.
* В течение 3 лет с момента ввода в эксплуатацию планируются регулярные обновления на основе отзывов заказчика и пользователей, а также внедрение роботизированной доставки.

*Основные функции:*

* Каталог фермеров и рынков
* Отзывы
* Ближайшие ярмарки
* Система уведомлений
* Формирование заказов
* Сообщество
* Связь пользователей с фермерами
* Робото-доставка
* Добавление товаров в избранно

*Уровни доступа к сайту:*

* Фермеры: регистрация, создание профиля, добавление продукции, управление заказами и взаимодействие с покупателями.
* Покупатели: поиск фермеров и рынков, просмотр предложений, оформление заказов, оставление отзывов.
* Администраторы: управление контентом, модерация пользователей, анализ данных и поддержка пользователей.
* Логисты: координация доставки, управление маршрутами и взаимодействие с фермерами и покупателями.
* Робототехники: внедрение роботизированной доставки.

*Особенности клиента:*

* Структура: физические лица, ИП и малые/средние фермерские хозяйства.
* Местоположение: приложение ориентировано на городские и пригородные районы, где есть спрос на местные продукты.
* Прочие особенности: учитывать предпочтения пользователей в отношении органических продуктов, устойчивого развития и поддержки местных производителей.

*Сложность функционала ПО:*

* Базовые функции: регистрация и авторизация пользователей, поиск фермеров и рынков, просмотр продукции.
* Продвинутые функции: интеграция с платёжными системами, система отзывов, возможность подписки на фермеров, функция уведомлений о новых поступлениях.

*Требования к надёжности:*

* Надёжность системы: минимум 99% времени безотказной работы, быстрый отклик (не более 2 секунд на запрос), регулярное резервное копирование данных и защита личной информации пользователей.
* Первоначальное количество личных кабинетов: 100 000. С возможностью дальнейшего бесперебойного расширения.
* Нагрузка: активно одновременно 5.000 пользователей, до 800 запросов в секунду
* Безопасность: защита от несанкционированного доступа и утечек персональных и платежных данных.

*Технологический стек:*

* Frontend: React или Vue.js для веб-приложения; React Native или Flutter для мобильного приложения.
* Backend: Node.js с Express или Python с Django для серверной части.
* База данных: MongoDB или PostgreSQL для хранения информации о пользователях, продуктах и рынках.
* API: использование картографических API (например, Google Maps) для отображения местоположений.
* Роботизированная система доставки: на основе опыта Яндекс Доставки в Москве и Иннополисе.

*Монетизация:*

* Финансирование заказчика.
* Реклама от местных производителей.
* Премиум-подписка для фермеров с дополнительными функциями (например, продвинутые аналитические инструменты).
* Партнёрство с магазинами здорового питания или ресторанами.

*Сильные и слабые стороны предметной области:*

*Сильные стороны:*

1. Прямое соединение фермеров с покупателями помогает развивать местную экономику, минуя посредников.
2. Короткие цепочки поставок сокращают транспортные расходы и снижают углеродный след.
3. Покупатели получают продукты напрямую от производителей, что обеспечивает их качество и свежесть.
4. Приложение может обучать пользователей особенностям сезонных продуктов и органического земледелия.
5. Платформа объединяет фермеров и покупателей, укрепляя локальные связи.
6. Фильтры по категориям продуктов упрощают выбор для покупателей.

*Слабые стороны:*

1. В некоторых районах может быть мало фермеров, что снижает разнообразие предложения.
2. Технические проблемы: в сельской местности возможны проблемы с интернет-соединением, затрудняющие использование приложения.
3. Отсутствие единых правил у фермеров (например, в упаковке или доставке) может вызывать неудобства у покупателей.
4. Конкуренция с крупными ритейлерами: крупные магазины могут предлагать более низкие цены и стабильный ассортимент круглый год.
5. Проблемы с логистикой: организация транспортировки в отдаленные районы требует дополнительных ресурсов и затрат.
6. Сезонность: ассортимент сильно зависит от времени года, что ограничивает выбор в межсезонье
7. **Роли команды разработчиков (14 человек)**

* Ведущий разработчик
* Младший разработчик
* Стажёр-разработчик
* Архитектор
* Системный администратор
* Бизнес-аналитик
* Технический писатель
* IS Control – контроль ИБ
* Релиз-менеджер
* Менеджер проекта
* Ведущий тестировщик
* Младший тестировщик
* Инженер
* Робототехник

1. **Смотреть пункт 2**
2. **Модели жизненного цикла**

1. Водопадная модель

Описание: линейный подход, где каждый этап (анализ, проектирование, реализация, тестирование, внедрение) выполняется последовательно.

Плюсы:

Позволяет четко планировать этапы разработки стандартных функций, таких как каталог фермеров или система отзывов. Обеспечивает полную документацию на каждом этапе, что важно для дальнейшего сопровождения.

Минусы:

Не дает возможности оперативно вносить изменения в требования, что критично для нашего динамичного проекта. Приводит к позднему тестированию функционала, когда ошибки в геолокации или платежах обнаруживаются только на финальных этапах. Затягивает выход MVP на длительный срок.

2. Гибкая модель разработки

Описание: итеративный подход, который предполагает частые изменения и вовлечение пользователей на всех этапах разработки.

Плюсы:

Позволяет быстро выпустить рабочий MVP с базовым функционалом поиска фермеров. Дает возможность постепенно добавлять новые функции по мере необходимости. Обеспечивает постоянную обратную связь с реальными пользователями - фермерами и покупателями. Позволяет гибко адаптировать разработку под будущие сложные интеграции (роботизированная доставка).

Минусы:

Требует высокой дисциплины и слаженной работы всей команды разработчиков. Создает сложности с точной оценкой сроков реализации отдельных функций. Может привести к чрезмерному увлечению доработками второстепенных элементов интерфейса.

3. Модель V-образного цикла

Описание: модель, которая сочетает в себе элементы водопадной модели и тестирования, где каждый этап проектирования соответствует этапу тестирования.

Плюсы:

Обеспечивает надежное тестирование критически важных систем (платежи) на ранних этапах. Позволяет четко документировать все процессы разработки.

Минусы:

Не справляется с частыми изменениями требований, характерными для нашего проекта. Замедляет процесс добавления новых функций по запросам пользователей. Усложняет интеграцию с внешними сервисами.

**Обоснование выбора модели**

Выбор: Kanban

Обоснование:

* Гибкость под меняющиеся требования

Kanban позволяет мгновенно адаптироваться к сезонным изменениям спроса и новым пожеланиям фермеров без жестких временных рамок.

* Быстрый вывод MVP

Методология дает возможность выпускать готовые модули по мере их завершения, что ускоряет получение первой рабочей версии приложения.

* Прозрачность для всех участников

Визуальная Kanban-доска делает процесс разработки понятным для всех членов команды, включая нетехнических специалистов.

* Эффективная работа с разнородными задачами

Позволяет параллельно вести разработку интерфейсов, интеграцию платежей и логистических алгоритмов.

* Оптимизация ресурсов стартапа

Минимизирует бюрократические затраты и позволяет быстро корректировать приоритеты в условиях ограниченного бюджета.

* Поддержка распределенных команд

Единая система управления задачами эффективно синхронизирует работу всех специалистов, включая удаленных сотрудников.

* Простое масштабирование

Методология естественным образом адаптируется под рост проекта и добавление новых функциональных направлений.

1. **Смотреть пункт 5**
2. **Виды деятельности и этапы**

*ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ*

[6.3.1.3](https://6.3.1.3/).1.3 По мере необходимости и при согласии всех заинтересованных сторон требования проекта могут быть изменены на этом этапе для достижения критериев завершения.

6.2.4.3.3.4 Гарантировать обеспечение обратной связи от результатов выполненных оценок к штатному персоналу.

*ПРОЕКТИРОВАНИЕ*

[7.1.3.3](https://7.1.3.3/).1 Проектирование архитектуры программных средств Для каждого программного элемента (или элемента конфигурации, если он определен) данный вид деятельности состоит из решения следующих задач:

[7.1.3.3](https://7.1.3.3/).1.1 Исполнитель должен преобразовать требования к программным составным частям в архи­тектуру, которая описывает верхний уровень его структуры и идентифицирует программные компоненты. Необходимо гарантировать, что все требования к программным составным частям распределяются по про­граммным компонентам и в дальнейшем уточняются для облегчения детального проектирования. Архитек­туру программной составной части необходимо документировать.

[7.1.4.3](https://7.1.4.3/).1.2 Исполнитель должен разработать и документально оформить детальный проект для вне­шних интерфейсов к программным составным частям, между программными компонентами и между про­граммными блоками. Необходимо, чтобы детальный проект для интерфейсов позволял проводить кодиро­вание без потребности в получении дополнительной информации. [7.1.4.3](https://7.1.4.3/).1.3 Исполнитель должен разработать и документально оформить детальный проект базы данных.

*РЕАЛИЗАЦИЯ*

[7.1.5.3](https://7.1.5.3/) Виды деятельности и задачи При реализации проекта необходимо выполнять следующие виды деятельности и задачи в соответствии с принятыми в организации политиками и процедурами в отношении процесса конструирования программных средств.

*ТЕСТИРОВАНИЕ*

[7.1.5.3](https://7.1.5.3/).1.2 Исполнитель должен тестировать каждый программный блок и базу данных, гарантируя, что они удовлетворяют требованиям. Результаты тестирования должны быть документально оформлены.

*ВНЕДРЕНИЕ*

[6.3.1.3](https://6.3.1.3/).2.1 Планы должны включать в себя, по крайней мере: а) графики работ для своевременного завершения задач; b) оценку усилий; с) ресурсы, необходимые для выполнения задач; d) распределение задач; е) распределение обязанностей; f) количественное определение рисков, связанных с задачами или самим процессом; g) мероприятия по гарантии качества для применения в пределах всего проекта; h) затраты, связанные с выполнением процесса; і) обеспечение окружающей среды и инфраструктуры; j) определение и сопровождение модели жизненного цикла, состоящей из стадий, используя конкретные модели жизненного цикла для проектов организации.

[6.3.1.3](https://6.3.1.3/).3 Активизация проекта Данный вид деятельности состоит из решения следующих задач:

[6.3.1.3](https://6.3.1.3/).3.1 Менеджер должен получить полномочия на проект.

[6.3.1.3](https://6.3.1.3/).3.2 Менеджер должен представить заявки на необходимые ресурсы для выполнения проекта.

[6.3.1.3](https://6.3.1.3/).3.3 Менеджер должен инициировать выполнение планов проекта для удовлетворения совокупности целей и критериев осуществления управления проектом.

*ЭКСПЛУАТАЦИЯ*

[6.4.9.3](https://6.4.9.3/).1.2 Оператор должен определять процедуры для получения, регистрации, решения, просле­живания проблем и обеспечения обратной связи. Всякий раз, когда возникают проблемы, они должны быть зарегистрированы и введены в процесс решения проблем программных средств

[6.2.5.3](https://6.2.5.3/).1.6 Организация должна проводить мониторинг состояния совершенствования продукции и услуг в области качества.

[6.4.9.3](https://6.4.9.3/).1.2 Оператор должен определять процедуры для получения, регистрации, решения, просле­живания проблем и обеспечения обратной связи. Всякий раз, когда возникают проблемы, они должны быть зарегистрированы и введены в процесс решения проблем программных средств.

[6.4.9.3](https://6.4.9.3/).4.1 Оператор должен обеспечивать содействие и консультации пользователей по их просьбе. Эти заявки и последующие действия должны быть зарегистрированы и проконтролированы.

[6.4.9.3](https://6.4.9.3/).4.2 Оператор должен направлять заявки пользователей (по мере необходимости) для выпол­нения в процессе сопровождения программных средств (см. 6.4.10). Все решения должны контролироваться для заключения об их результативности.

[6.4.9.3](https://6.4.9.3/).5.1 Оператор должен направлять возникшие проблемы в процесс решения проблем в про­граммных средствах для их устранения.

*ПОДДЕРЖКА*

[6.2.5.3](https://6.2.5.3/).2 Корректирующие действия менеджмента качества Данный вид деятельности состоит из решения следующих задач:

[6.2.5.3](https://6.2.5.3/).2.1 Организация должна предпринимать корректирующие действия, если конечные цели менеджмента качества не достигнуты.

[6.2.5.3](https://6.2.5.3/).2.2 Организация должна выполнять корректирующие действия и сообщать о результатах в пределах организации.

[6.3.2.3](https://6.3.2.3/).2 Управление проектом Данный вид деятельности состоит из решения следующих задач:

[6.3.2.3](https://6.3.2.3/).2.1 Менеджер должен исследовать, проанализировать и принять решения по проблемам, обнаруженным при выполнении проекта.

[6.3.2.3](https://6.3.2.3/).2.2 Менеджер должен представлять отчеты в согласованные сроки о развитии проекта, показывая соблюдение планов и решения в случае остановки в развитии проекта.

ЗАВЕРШЕНИЕ ПРОЕКТА

[7.2.3.2](https://7.2.3.2/) В результате успешного осуществления процесса гарантии качества программных средств: а) разрабатывается стратегия обеспечения гарантии качества; b) создается и поддерживается свидетельство гарантии качества; с) идентифицируются и регистрируются проблемы и (или) несоответствия с требованиями; d) верифицируется соблюдение продукцией, процессами и действиями соответствующих стандартов, процедур и требований.

УЛУЧШЕНИЕ ПРОЦЕССОВ

[6.3.2.3](https://6.3.2.3/).3 Оценка проекта Данный вид деятельности состоит из решения следующих задач:

[6.3.2.3](https://6.3.2.3/).4 Завершение проекта Данный вид деятельности состоит из решения следующих задач:

[6.3.2.3](https://6.3.2.3/).4.1 Когда все программные продукты, действия и задачи завершаются, менеджер должен определить, закончен ли проект, принимая во внимание критерии, указанные в контракте или установленные как часть процедуры организации.

[6.3.2.3](https://6.3.2.3/).4.2 Эти результаты и отчеты должны быть архивированы в соответствующей среде, как определено в контракте.

1. **Смотреть пункт 7**
2. **Построенная модель жизненного цикла**
3. *Подготовительный этап*

Сбор требований: определение основных потребностей пользователей и функций (поиск фермеров, отображение информации о продуктах и т.д.).

Создание бэклога: формирование списка задач (user stories), которые должны быть выполнены в ходе разработки.

1. *Формирование команды*

Смотреть пункт 3.

1. *Планирование спринта*

Определение длительности спринта: выбрать продолжительность спринта (2-4 недели).

Планирование спринта: провести встречу по планированию спринта, чтобы выбрать задачи из бэклога, которые будут выполнены в текущем спринте.

1. *Разработка*

Итеративная разработка: разработка функционала по частям, каждая из которых должна быть завершена к концу спринта.

1. *Тестирование*

Тестирование в процессе разработки: проводить тестирование новых функций параллельно с их разработкой, чтобы выявлять и исправлять ошибки на ранних стадиях.

Интеграционное тестирование: убеждаться, что новые функции корректно работают вместе с уже существующими.

1. *Демонстрация*

Демонстрация результатов: в конце спринта провести демонстрацию (sprint review) для заинтересованных сторон, показать, что было сделано.

Сбор обратной связи: получить отзывы от пользователей и заинтересованных сторон о новой функциональности.

1. *Ретроспектива*

Проведение ретроспективы: после демонстрации провести встречу для обсуждения того, что прошло хорошо, что можно улучшить и какие действия необходимо предпринять для улучшения процессов в следующем спринте.

1. *Обновление бэклога*

Анализ и обновление бэклога: на основе полученной обратной связи обновить бэклог, добавив новые задачи и приоритизировав их.

1. *Повторение процесса*

Начало нового спринта: повторять шаги 3-8, добавляя новые функции и улучшая приложение на основе обратной связи.

1. *Завершение проекта*

Оценка успешности проекта: провести финальный обзор проекта, оценив выполнение целей и удовлетворенность пользователей.

Подготовка итогового отчета: создать документ с результатами работы над проектом.

Закрытие проекта: формально завершить проект, включая архивирование документации и материалов.