Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет**

**при Правительстве Российской Федерации»**

**(Финуниверситет)**

**Департамент анализа данных и машинного обучения**

**Факультет** **информационных технологий и анализа больших данных**

**КУРСОВАЯ РАБОТА**

**по дисциплине** «Программная инженерия»

**на тему:** «Сравнительный анализ Agile методологий Scrum и XP на примере

конкретного проекта. Предметная область: ПО для финтесс-треккера»

Направление подготовки: 09.03.03 – Прикладная информатика

Профиль: «ИТ-сервисы и технологии обработки данных в экономике и финансах»

**Выполнила студентка** 3 курса,

группы ПИ21-7,

формы обучения очной

Филиппова Е. Р.

**Руководитель:**

Журавлева М. Г.

|  |  |  |
| --- | --- | --- |
| Дата поступления работы на кафедру: | Работа допущена  к защите:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | Работа защищена  с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) |
| \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2024 г. | \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2024 г. | \_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

Москва 2024

# **Оглавление**

Введение

1. Методология исследования
   1. Основные принципы Scrum
   2. Основные принципы Extreme Programming (XP)
   3. Преимущества и недостатки каждой из методологий
   4. Описание проекта ПО для финтесс-треккера и его особенностей
2. Сравнительный анализ Scrum и XP
   1. Применение Scrum в рамках проекта
   2. Применение XP в рамках проекта
   3. Сравнительный анализ результатов применения Scrum и XP

Заключение

Список использованных источников

Приложения

[Оглавление 1](#_Toc166185118)

[Введение 3](#_Toc166185119)

[1.Методология исследования 5](#_Toc166185120)

[1.1. Основные принципы Scrum 5](#_Toc166185121)

[1.2 Основные принципы Extreme Programming (XP) 9](#_Toc166185122)

[1.3 Преимущества и недостатки каждой из методологий 14](#_Toc166185123)

[1.4 Описание проекта ПО для финтесс-треккера и его особенностей 15](#_Toc166185124)

[2.Сравнительный анализ Scrum и XP 18](#_Toc166185125)

[2.1 Применение Scrum в рамках проекта 18](#_Toc166185126)

[2.1.1 Описание команды 18](#_Toc166185127)

[2.1.2 Персона пользователя (User Persona) 19](#_Toc166185128)

[2.1.3 Пользовательские истории (User Stories) 21](#_Toc166185129)

[2.1.4 Формирование Backlog продукта 22](#_Toc166185130)

[2.1.5 Планирование первого спринта (Sprint Planning) 25](#_Toc166185131)

# **Введение**

В современном мире программное обеспечение играет ключевую роль в нашей повседневной жизни, оказывая влияние на различные сферы деятельности от игровой индустрии до бизнеса. С развитием технологий разработчики сталкиваются с постоянной необходимостью создания и обновления программных продуктов, что требует применения эффективных методов управления проектами. В этом контексте Agile-методологии выступают важным инструментом, обеспечивая гибкий подход к разработке и способствуя быстрой адаптации к изменениям.

Выбор подходящей Agile-методологии для конкретного проекта играет решающую роль в успешной реализации задач. В рамках данного исследования мы обращаем внимание на процесс разработки программного обеспечения для фитнес-треккера, а именно на методологии Scrum и XP.

Наша **цель** заключается в изучении особенностей их применения в контексте данного проекта, а также в анализе их преимуществ и недостатков. Мы также рассматриваем их соответствие специфике требований и задач, стоящих перед разработчиками в данной области.

**Актуальность** нашего исследования обусловлена растущим интересом к Agile-методологиям в сфере разработки программного обеспечения и необходимостью выбора наиболее подходящего подхода к управлению проектами в конкретных условиях. Проекты, связанные с разработкой фитнес-треккеров, в силу своей специфики и динамичного характера, требуют особого внимания к процессу разработки, что делает результаты нашего исследования важными для разработчиков и менеджеров проектов.

**Научная проблема**, поставленная перед нами, заключается в определении оптимального подхода к управлению проектами в области разработки программного обеспечения для фитнес-треккера с учетом их специфики и требований рынка. Таким образом, **объектом** **исследования** является процесс разработки программного обеспечения для фитнес-треккера, а **предметом** - методологии Scrum и XP, их особенности и эффективность применения в данной области.

Наша **задача** состоит в том, чтобы определить оптимальный подход к управлению проектами в области фитнес-треккеров и предоставить читателю полное представление о Scrum и XP, помогая им выбрать подход к управлению проектом. Мы планируем изучить принципы Agile методологий, проанализировать литературу, описать особенности проекта ПО для финтесс-треккера, определить критерии сравнения и проанализировать использование двух гибких методологий на его основе.

Таким образом, наша работа охватывает все аспекты сравнения Scrum и XP и их применимости в контексте разработки программного обеспечения для фитнес-треккеров.

# **Методология исследования**

## **Основные принципы Scrum**

Scrum - популярная методология разработки программного обеспечения, которая обеспечивает гибкость и способность адаптироваться к меняющимся требованиям. Она была разработана в начале 1990-х годов Кеном Швабером и Джеффом Сазерлендом и с тех пор получила широкое распространение.

Основным принципом Scrum является разделение процесса разработки на небольшие автономные этапы, называемые итерациями. Во время каждой итерации команда сосредотачивается на выполнении определенного набора задач, которые способствуют достижению общей цели - созданию готового продукта. После завершения итерации команда анализирует результаты и получает отзывы от заказчиков или пользователей, чтобы оценить эффективность своей работы. Эти отзывы помогают команде определить области для улучшения и при необходимости адаптировать свой подход к меняющимся требованиям.

Инкрементальный подход в Scrum означает, что команда разрабатывает продукт поэтапно. Она не пытается создать готовый продукт сразу. Вместо этого сначала разрабатываются наиболее важные функции, а затем добавляются новые по мере продвижения. Таким образом, можно быстро получить работающий продукт и понять, что нужно улучшить.

Работа над проектом разбивается на короткие периоды времени, называемые спринтами. Каждый спринт обычно длится от одной до четырех недель. В конце каждого спринта команда представляет результаты своей работы, позволяя заказчику увидеть прогресс и внести необходимые коррективы, если что-то не совсем так.

Основные принципы Scrum включают несколько важных аспектов, которые помогают команде успешно выполнять проект. Например, команда должна быть самоорганизованной и иметь ясные цели, чтобы знать, что они должны достичь.

Вот краткое описание каждого из них:

1. Scrum определяет три основные роли: Product Owner, Scrum Master и Development Team. Каждая из трех ключевых ролей играет важную роль в обеспечении успешной реализации проекта.
   1. Product Owner (Владелец продукта): Он представляет интересы заказчика и пользователей продукта. Основная задача Владельца продукта - определить, что конкретно должен включать в себя продукт, и убедиться, что команда разработки понимает эти требования. Он отвечает за создание и обновление Product Backlog - списка задач и требований, необходимых для достижения целей проекта. Также определяет приоритеты в Product Backlog, учитывая бизнес-потребности и ожидания заказчика.
   2. Scrum Master (Мастер Scrum): Это специалист, который следит за правильным применением методологии Scrum в проекте. Он помогает команде разработки использовать Scrum наилучшим образом и устранять препятствия, которые могут возникнуть. Мастер Scrum проводит обучение и коучинг членов команды по методологии Scrum, помогает им применять принципы и практики Scrum.
   3. Development Team (Команда разработки): Это группа специалистов, ответственных за создание продукта. Команда разработки сама принимает решения о том, как выполнять работу и как достигать поставленных целей. У нее должны быть все необходимые навыки и компетенции для выполнения задач из Product Backlog. Они работают коллективно, чтобы создавать новую функциональность на протяжении каждого спринта.
2. В Scrum используются различные артефакты, которые играют важную роль в организации и управлении процессом разработки. Эти артефакты помогают команде разработки и заинтересованным сторонам понять текущее состояние проекта, определить приоритеты и принимать информированные решения.
   1. Бэклог продукта (Product Backlog): Это список всех задач, требований и идей, которые могут быть реализованы в рамках проекта. Product Backlog поддерживается и обновляется Владельцем продукта и содержит задачи, необходимые для достижения целей проекта. Задачи описываются в виде User Stories или других формализованных элементов и часто сопровождаются приоритетами и оценками сложности. Он служит источником задач для планирования спринтов и определения направления развития продукта.
   2. Бэклог спринта (Sprint Backlog): Это список задач, выбранных командой разработки для выполнения в рамках текущего спринта. Формируется на основе элементов из Product Backlog, выбранных для реализации в конкретном спринте. Sprint Backlog содержит подробное описание каждой задачи, сроки выполнения и ответственных исполнителей. Может подвергаться изменениям в процессе выполнения спринта.
   3. Инкремент (Increment): Это работающая версия продукта, которая достигается в конце каждого спринта. Инкремент должен быть полностью завершенным и работоспособным, содержать новую функциональность или улучшения, которые могут быть протестированы и использованы. Каждый новый Increment представляет собой шаг вперед в развитии продукта и демонстрирует прогресс команды разработки.
3. Существуют определенные события в Scrum, которые играют важную роль в управлении процессом разработки и обеспечивают команде возможность оценивать свой прогресс и адаптировать свою работу.
   1. Планирование спринта (Sprint Planning): Это событие происходит в начале каждого спринта и направлено на определение целей и плана работы на следующий период времени. На этом событии команда разработки совместно с Владельцем продукта обсуждает задачи, которые будут выполнены в следующем спринте, и устанавливает цели, которые они планируют достигнуть к его завершению.
   2. Ежедневный Scrum (Daily Scrum): Это короткое событие, которое проводится ежедневно в течение спринта. Каждый член команды отвечает на три вопроса: о проделанной работе, планах на день и препятствиях. Это позволяет команде быть в курсе текущего состояния проекта и выявлять любые проблемы или препятствия.
   3. Обзор спринта (Sprint Review): Это событие происходит в конце каждого спринта и направлено на представление выполненной работы заказчику или заинтересованным сторонам. На обзоре команда разработки демонстрирует продукт и обсуждает его с заказчиком, получая обратную связь для дальнейшего улучшения.
   4. Ретроспектива спринта (Sprint Retrospective): Это событие происходит после завершения каждого спринта и направлено на анализ работы команды и выявление возможностей для улучшения. Члены команды обсуждают, что прошло хорошо и что можно улучшить, чтобы повысить эффективность работы в следующем спринте.

На основе вышеизложенного, Scrum становится мощным инструментом управления проектами, обеспечивая командам способность быстро реагировать на изменения в требованиях и на рынке. Его методология широко применяется в различных отраслях и проектах, что делает его одним из наиболее востребованных подходов к разработке программного обеспечения в настоящее время. В результате Scrum становится неотъемлемым инструментом для успешного управления проектами в современном бизнесе.

## **Основные принципы Extreme Programming (XP)**

Extreme Programming (XP) - это метод разработки программного обеспечения, который является частью гибкого подхода к работе. Главная цель - обеспечить высокое качество кода и более быструю доставку ценности продукта заказчикам. XP фокусируется на создании адаптируемой среды, что способствует ускорению процесса разработки. Давайте вместе рассмотрим эти принципы подробнее:

1. Обратная связь (Feedback) играет ключевую роль в методологии Extreme Programming (XP), где она рассматривается как один из основных элементов успешной разработки программного обеспечения. Обратная связь в XP включает в себя два основных аспекта: обратную связь от заказчика и внутреннюю обратную связь в команде разработки.
   1. XP активно поддерживает взаимодействие с заказчиком на протяжении всего процесса разработки. Регулярные сеансы обратной связи позволяют заказчику выражать свои требования, предпочтения и ожидания относительно продукта. Это помогает команде понять и уточнить требования, а также оперативно реагировать на изменения и корректировать планы разработки. Благодаря обратной связи от заказчика, команда может быть уверена, что создаваемый продукт соответствует его ожиданиям и требованиям.
   2. Внутренняя обратная связь в команде разработки включает обмен знаниями, опытом и мнениями между участниками команды. Это может включать обсуждение различных подходов к решению проблем, обмен идеями, а также предоставление конструктивной критики и обратной связи по работе других членов команды. Благодаря этому процессу команда может постоянно улучшать свою работу, обмениваясь опытом и находя наилучшие решения для достижения поставленных целей. Внутренняя обратная связь также помогает команде быстро реагировать на возникающие проблемы и адаптироваться к изменениям в проекте.
2. Принцип простоты в методологии Extreme Programming (XP) представляет собой стремление к созданию программного кода и архитектуры продукта, которые максимально просты и понятны. Цель этого принципа - упростить процесс разработки и поддержки программного обеспечения.
   1. Создание простого и понятного кода. В XP простота не только в реализации, но и в понимании. Код должен быть легким для чтения и понимания другими членами команды. Это облегчает обслуживание и развитие программы, так как разработчики могут быстро понять существующий код и вносить изменения.
   2. Избегание излишней сложности. Команда XP старается избегать излишней сложности в проекте. Сложные алгоритмы или структуры данных могут усложнить код и увеличить риск ошибок. Вместо этого предпочтение отдаётся простым и эффективным решениям.
   3. Упрощение поддержки и развития. Принцип простоты также способствует упрощению поддержки и развития продукта. Простой код упрощает поиск ошибок, добавление новых функций и изменения в существующем функционале. Это позволяет команде быстро реагировать на изменения требований заказчика и адаптироваться к изменяющимся условиям.
3. В методологии экстремального программирования итерации являются важнейшей частью, которая играет значительную роль в разработке программного обеспечения. Эти итерации основаны на поэтапном подходе, поэтому продукт создается и выпускается поэтапно. Это позволяет команде быстро получать обратную связь от клиентов и легко приспосабливаться к изменениям в их потребностях.

Каждая итерация в XP представляет собой своего рода цикл разработки, который включает несколько этапов:

* 1. Анализ: Здесь команда определяет, какие функции или изменения будут включены в текущую итерацию и проводит анализ требований.
  2. Проектирование: Определяется архитектура и дизайн новых функций, а также планируются необходимые изменения в существующем коде.
  3. Реализация: Программисты приступают к написанию кода для воплощения запланированных изменений и функциональности.
  4. Тестирование: После завершения разработки проводится тестирование, включая как автоматизированные, так и ручные тесты, для проверки корректности работы новой функциональности и выявления возможных ошибок.

1. В методе экстремального программирования (XP) идея небольших и частых выпусков - это способ работы, который побуждает команду выпускать программный продукт часто и небольшими порциями. Этот метод снижает риск, повышает качество и повышает ценность продукта для потребителя по мере его разработки.
   1. В XP предпочтение отдается частым выпускам продукта, чтобы заказчик мог получать обратную связь по ходу проекта. Чем чаще команда выпускает новые версии продукта, тем быстрее можно понять, что работает хорошо, а что требует улучшений. Это позволяет команде гибко адаптировать планы и изменять приоритеты в соответствии с реальными потребностями заказчика.
   2. Предпочтение маленьким релизам в XP помогает снизить риски разработки. Небольшие изменения проще контролировать и управлять, что уменьшает вероятность серьезных ошибок или непредвиденных проблем. Более того, маленькие релизы позволяют быстрее исправлять обнаруженные ошибки и улучшать функционал на основе обратной связи.
   3. Частые и небольшие релизы также увеличивают ценность продукта на каждом этапе разработки. Вместо ожидания завершения всего проекта для предоставления заказчику функционала, команда может постепенно внедрять новые возможности и улучшения, что позволяет заказчику быстрее начать использовать и получать пользу от продукта.
2. В Extreme Programming (XP) особое внимание уделяется проверке продукта перед его выпуском. Это важно для того, чтобы убедиться, что программа работает правильно и соответствует требованиям заказчика. Есть два основных способа проверки:
   1. Ручное тестирование: Люди из команды тестируют программу, проверяя, как она работает. Они могут делать разные виды тестов, чтобы найти ошибки и улучшить программу. Например, они могут проверить, что все кнопки на экране работают правильно или что программа работает быстро.
   2. Автоматическое тестирование: Это когда специальные программы тестируют программу автоматически. Они проверяют, что все работает правильно, и находят ошибки. Это помогает быстрее найти и исправить проблемы.
3. Парное программирование - это ключевая практика, предложенная методологией экстремального программирования (XP). В этой практике два программиста работают вместе за одним компьютером, обсуждая, планируя и записывая код. Такой подход помогает создавать более качественные и эффективные программы.
   1. Когда программисты работают в паре, они делятся знаниями и опытом друг с другом. Это помогает каждому из них стать лучше в своей работе и лучше понять проект.
   2. Также парное программирование помогает улучшить качество кода. Вместе программисты могут находить ошибки и улучшать структуру кода, делая его более понятным и надежным.
   3. Благодаря постоянному обсуждению кода, вероятность обнаружения ошибок увеличивается. Это позволяет сделать программу более стабильной и надежной.
   4. Все участники команды в парном программировании принимают участие в принятии решений. Это позволяет избежать ошибок и лучше понимать проект.
4. Владение кодом всей командой является ключевой частью XP (Extreme Programming), что означает, что каждый член команды может вносить изменения в код и улучшать его. Таким образом, каждый имеет право голоса в отношении того, что делает код и как он работает, что помогает распределить знания и ответственность между членами команды. Это также означает, что код, скорее всего, будет качественным и стабильным.
   1. Один из плюсов такого подхода - это то, что все члены команды могут знать о проекте как можно больше. Все имеют доступ к коду и могут его менять, что улучшает общение в команде, делает обмен опытом проще и позволяет быстрее реагировать на изменения и проблемы.
   2. Кроме того, такой подход помогает улучшить качество кода. Поскольку каждый член команды может предлагать свои идеи и исправления, ошибки легче находить и устранять, а код становится более понятным и читабельным.
   3. Еще одно преимущество - это устойчивость процесса разработки. Если кто-то из команды временно отсутствует или уходит из проекта, другие могут легко взять на себя управление над его кодом и продолжить работу. Это делает команду менее зависимой от отдельных членов и обеспечивает непрерывность работы.
   4. Коллективное владение кодом также способствует повышению мотивации и ответственности. Каждый знает, что его вклад важен, и чувствует себя более вовлеченным и ответственным за результат.

Итак, принципы Extreme Programming обеспечивают эффективную работу команды разработки, способствуют достижению высокого качества продукта и ускоряют процесс доставки ценности продукта заказчику.

## **Преимущества и недостатки каждой из методологий**

|  |  |  |
| --- | --- | --- |
| **Критерии** | **Scrum** | **XP** |
| Гибкость | + | - |
| Прозрачность | + | + |
| Сотрудничество | + | + |
| Итеративный подход | + | + |
| Частые релизы | + | + |
| Парное программирование | - | + |
| Тестирование | - | + |
| Рефакторинг | - | + |
| Сложность внедрения | - | - |
| Зависимость от Scrum-мастера | - | + |
| Зависимость от навыков разработчиков | + | - |
| Управление продуктом | + | - |
| Временные рамки | + | - |

Таблица 1. Преимущества и недостатки методологий Scrum и XP

Исходя из приведенных преимуществ и недостатков, можно сделать вывод, что обе методологии имеют свои преимущества и недостатки, и выбор между ними зависит от конкретных потребностей и условий проекта.

Scrum может быть более подходящим для проектов, требующих гибкости и адаптации к изменениям, а XP — для проектов, где важно качество кода и ускорение процесса разработки.

К тому же, важно отметить, что обе методологии требуют изменения культуры и процессов в организации, а также обучения и поддержки со стороны руководства. Поэтому критерий “сложность внедрения” был отмечен недостатком для каждой из них.

## **Описание проекта ПО для финтесс-треккера и его особенностей**

В первом квартале 2024 года в России зафиксирован рост интереса к онлайн-фитнесу, таковы выводы аналитиков компании "Билайн", основанные на анализе обезличенных данных интернет-трафика своих абонентов. В Москве спрос на онлайн-фитнес вырос на 30% по сравнению с предыдущим годом, после чего следуют Санкт-Петербург, Казань, Уфа, Краснодар, Екатеринбург, Ставрополь, Новосибирск, Нижний Новгород и Кемерово. Среди наиболее популярных приложений выделяются приложения для домашних тренировок, бега и сервисы для занятий йогой.

Интересно, что почти 77% пользователей таких фитнес-приложений - женщины, и их число увеличилось на 5% в 2024 году. Основная возрастная группа пользователей - люди от 35 до 40 лет, их активность наблюдается в будние дни, в основном с понедельника по среду.

Эти данные имеют важное значение для бизнеса. Они свидетельствуют не только о растущем интересе россиян к фитнесу, но и о стремлении к здоровому образу жизни в целом. Кроме того, наблюдается увеличение интереса к полезным продуктам, приложениям для правильного питания, спортивным товарам и услугам частных тренеров.

Исходя из приведенной статистики, видно, что рост спроса на фитнесс приложения не угасает, а только растёт, что обуславливает актуальность темы данной курсовой работы. Но, тем не менее, очевидно, что рынок приложений для поддержания здорового образа жизни довольно широк и разнообразен, поэтому стоит выходить на него как можно быстрее и с новыми революционными функциями, которые упростят жизнь многим людям и сэкономят их время и ресурсы.

Именно поэтому, было принято решение разрабатывать абсолютно инновационное приложение “SimpleFit”, направленное на максимально эффективное отслеживание физической активности и поддержку здорового образа жизни пользователей.

Помимо всем известных функций, оно будет содержать ряд совершенно новых возможностей, которые упростят жизнь каждого и привьют любовь к спорту:  
1. **Анализ техники выполнения упражнений:** Уникальная функция "SimpleFit" предоставляет возможность анализировать технику выполнения упражнений с помощью камеры смартфона. Затем искусственный интеллект анализирует видеозапись и предоставляет пользователю обратную связь о его технике. Искусственный интеллект может распознавать различные аспекты техники выполнения упражнений, такие как положение тела, амплитуда движений, скорость и т. д. Это несомненно поможет пользователям правильно выполнять упражнения, избегая травм и максимально эффективно проводить тренировки в любое удобное время.

2. **Анализ питания.** Приложение использует искусственный интеллект для анализа фотографий еды. Пользователь делает фото еды, загружает его в приложение, и искусственный интеллект анализирует его, определяя тип еды, её калорийность и содержание питательных веществ. Искусственный интеллект может распознавать различные продукты питания, такие как фрукты, овощи, мясо, молочные продукты и т. д. Он также может определять количество калорий, белков, жиров и углеводов в каждом продукте. Таким образом, функция анализа питания позволяет пользователям контролировать свой рацион и следить за количеством потребляемых калорий и питательных веществ. Это может помочь им достичь своих целей в области здорового образа жизни и сэкономить время на расчетах.

3. **Умное планирование тренировок:** Приложение предлагает персонализированные планы тренировок, основанные на целях и физической подготовке каждого пользователя. Это позволяет создавать оптимальные тренировочные программы, соответствующие особенностям каждого и достигать видимых результатов в кротчайшие сроки без вреда для организма.

Разработка “SimpleFit” будет осуществляться по методологии Agile, позволяющей команде оперативнее реагировать на изменения рынка и достигать лучших результатов. Также планируется внедрить итеративный подход, который предполагает постепенное добавление нового и улучшение существующего функционала. Таким образом, каждые 2-4 недели будут выпускаться новые завершенные функции, что позволит быстрее выйти на рынок и собрать аудиторию.

Ключевыми особенностями разработки станут обеспечение безопасности пользовательских данных, возможность масштабирования приложения для будущего расширения, разработка интуитивного пользовательского интерфейса и интеграция с другими сервисами, например, социальными сетями.

Немаловажным моментом будет обратная связь с пользователями, что поможет команде адаптировать приложение к их нуждам и предпочтениям. Такой подход позволит создать продукт, который будет по-настоящему полезным и удобным, а значит, успешным на рынке.

# **Сравнительный анализ Scrum и XP**

## **2.1 Применение Scrum в рамках проекта**

### **2.1.1 Описание команды**

Команда включает в себя следующих участников:

1. Анатолий, Product Owner:

* Определение стратегии продукта и его целей;
* Управление Product Backlog'ом;
* Приоритизация задач в соответствии с требованиями бизнеса и клиентов.

1. Юлия, Scrum Master:

* Обеспечение правильного применения методологии Scrum в команде;
* Устранение препятствий и обеспечение эффективного взаимодействия между членами команды;
* Организация и проведение ежедневных scrum-совещаний и других мероприятий Scrum.

1. Никита, Backend разработчик:

* Разработка и поддержка серверной части приложения;
* Обеспечение функциональности и безопасности бэкэнда;
* Взаимодействие с другими членами команды для интеграции бэкэнда с фронтендом и другими компонентами приложения.

4. Екатерина, Frontend разработчик:

* Разработка и поддержка пользовательского интерфейса приложения;
* Оптимизация интерфейса для обеспечения лучшего пользовательского опыта;
* Интеграция фронтенда с бэкэндом и другими компонентами приложения.

1. Владислав, Верстальщик:

* Верстка пользовательского интерфейса на основе дизайн-макетов;
* Обеспечение адаптивности интерфейса для совместимости с различными устройствами.

6. Елена, Дизайнер:

* Создание дизайн-макетов пользовательского интерфейса и элементов приложения;
* Разработка концепции дизайна с учетом требований бренда и ожиданий пользователей;
* Поддержка разработчиков и верстальщиков в реализации дизайна.

### **2.1.2 Персона пользователя (User Persona)**

Персона пользователя – это описание биографии человека, составленное на основе исследования рынка и опыта для описания соответствующих характеристик, потребностей и целей людей, которые будут использовать продукт. Используя эти биографии, команда разработчиков может представить портрет целевой аудитории приложения и понять, какие проблемы оно решит для них.



Рис.1 User persona 1

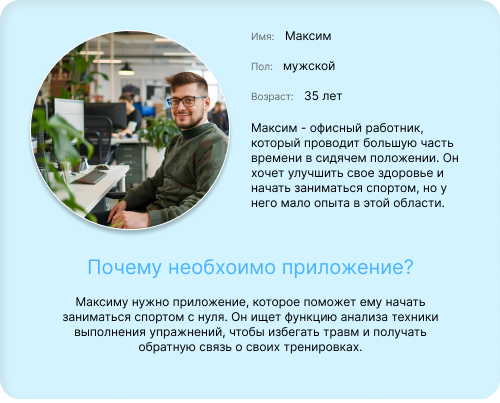


Рис.2 User persona 2

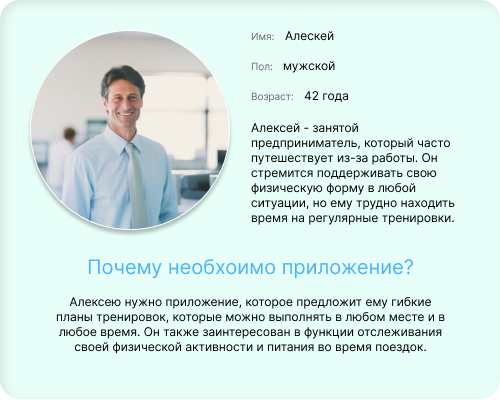


Рис.3 User persona 3

### **2.1.3 Пользовательские истории (User Stories)**

User Story, или пользовательская история, представляет собой способ увидеть функционал продукта глазами его конечного пользователя. Основная идея User Story - представить информацию кратко и наглядно, избегая технических деталей и лишних подробностей, а вместо этого сосредоточиться на потребностях и целях пользователей.

Один из ключевых этапов при разработке успешного продукта - изучение целевой аудитории, тех людей, которые в конечном итоге станут его пользователем или покупателем.

В основной части User Story обычно указываются:

* + - категория пользователя;
    - желаемое действие;
    - ожидаемый результат после выполнения действия.

Изображение выглядит как текст, Самоклеющийся листок, Шрифт, дизайн

Автоматически созданное описание

Рис.4 User Stories Map

### **2.1.4 Формирование Backlog продукта**

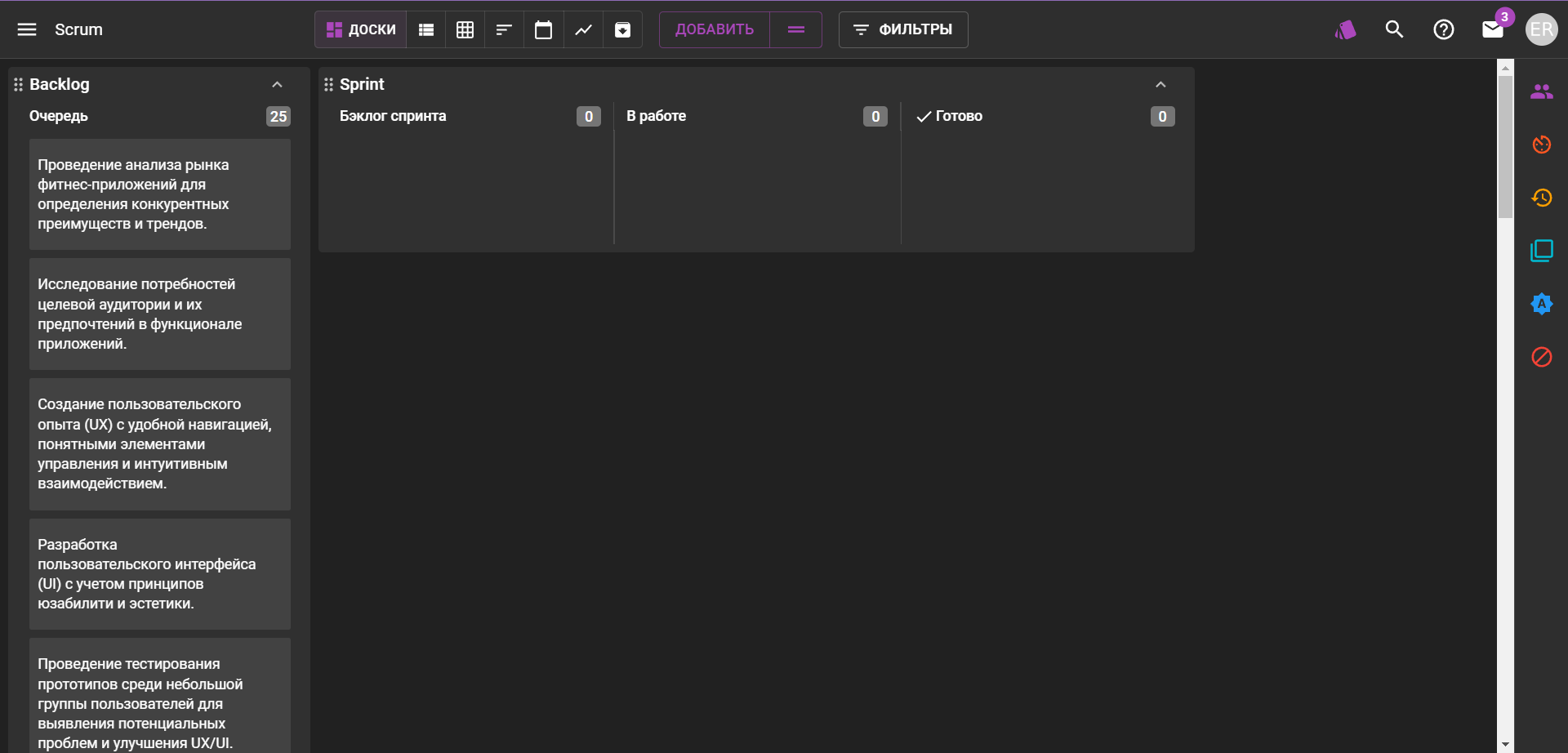


Рис.5 Backlog продукта

Созданы 26 карточек, из которых формируется Backlog приложения (расположены по приоритезации задач):

1. Проведение анализа рынка фитнес-приложений для определения конкурентных преимуществ и трендов.
2. Исследование потребностей целевой аудитории и их предпочтений в функционале приложений.
3. Создание пользовательского опыта (UX) с удобной навигацией, понятными элементами управления и интуитивным взаимодействием.
4. Разработка пользовательского интерфейса (UI) с учетом принципов юзабилити и эстетики.
5. Проведение тестирования прототипов среди небольшой группы пользователей для выявления потенциальных проблем и улучшения UX/UI.
6. Верстка адаптивных макетов приложения.
7. Разработка функции для съемки фотографий еды и анализа их с помощью искусственного интеллекта.
8. Интеграция с базой данных пищевых продуктов для получения информации о калориях и питательных веществах.
9. Разработка базы данных упражнений с инструкциями и видеороликами для пользователей.
10. Разработка функции анализа техники выполнения упражнений с использованием камеры смартфона.
11. Создание алгоритмов для распознавания различных аспектов техники выполнения упражнений.
12. Разработка функционала для отображения статистики тренировок и прогресса пользователей.
13. Разработка функционала для отслеживания физической активности (количества шагов, расстояния и сожженных калорий)
14. Интеграция с API устройств для сбора данных о физической активности
15. Создание алгоритмов для генерации персонализированных тренировочных программ на основе целей и физической подготовки пользователя.
16. Возможность установки целей и отслеживания их выполнения.
17. Создание функции для сохранения любимых тренировок и программ для последующего доступа и использования.
18. Разработка функционала для интеграции с социальными сетями.
19. Создание функции для установки напоминаний о тренировках и достижении целей.
20. Разработка функционала для создания и управления групповыми тренировками, возможность приглашения друзей.
21. Интеграция с базой данных фитнес-мероприятий и тренировок для предоставления пользователю информации о событиях в его районе.
22. Разработка функции для отслеживания пульса и других биометрических показателей с помощью устройств трекинга здоровья.
23. Создание возможности доступа к тренировкам от профессиональных тренеров через приложение.
24. Разработка модели монетизации приложения, включая премиум-подписки, рекламные блоки или покупку дополнительного контента.
25. Создание стратегии продвижения и привлечения пользователей для максимизации доходов.
26. Внедрение платежных систем для реализации премиум-функционала и монетизации приложения.

### **2.1.5 Планирование первого спринта (Sprint Planning)**

В данной курсовой работе мы будем проводить планирование только для первого спринта, так как методология Scrum подразумевает поэтапное планирование. Наша цель - выбрать задачи из Product Backlog, которые будут реализованы в первом спринте.

Для начала мы оценим задачи из Product Backlog, чтобы определить их сложность. Затем проведем обсуждение с командой, чтобы определить, какие задачи мы можем взять в первый спринт, учитывая доступные ресурсы и сроки. Ключевыми критериями выбора будут значимость задачи для пользователей и их ожидания, а также техническая и операционная возможность их выполнения в рамках одного спринта.

Ниже представлено планирование первого спринта и распределение обязанностей между участниками команды:

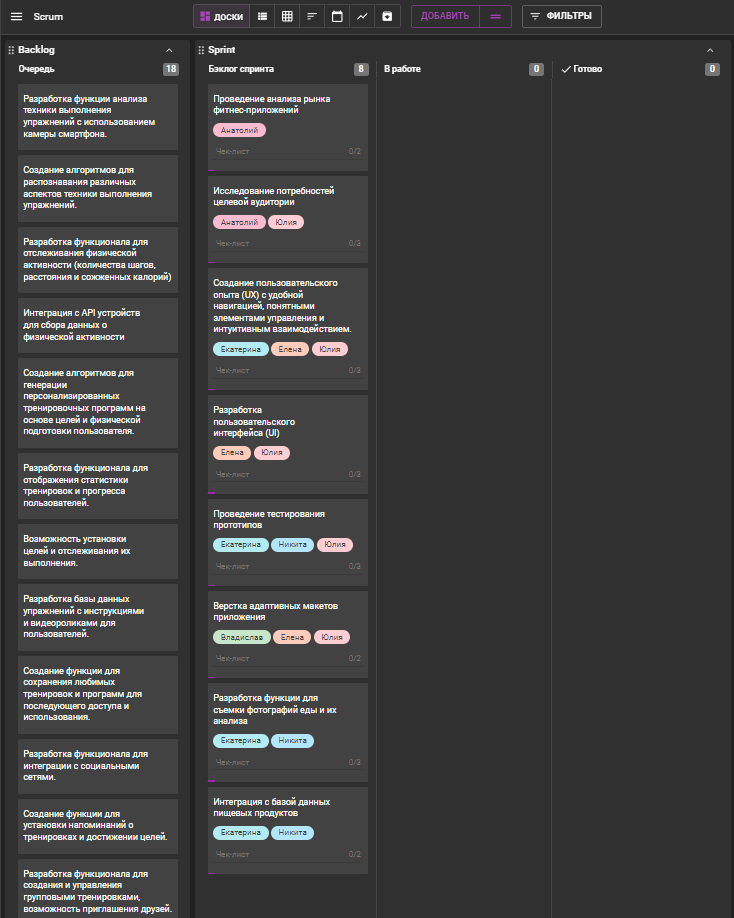


Рис.6 Первый Sprint

Далее представлены подробные описания карточек с задачами:

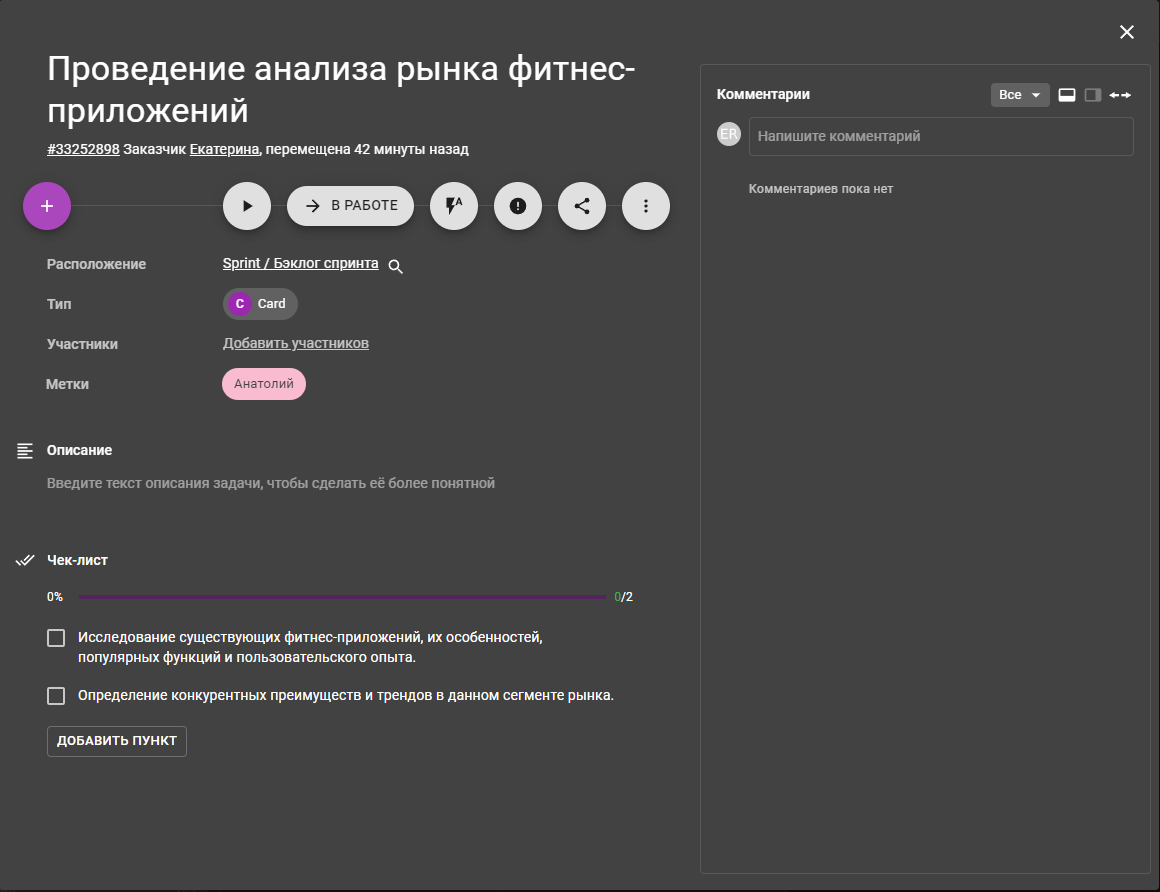


Рис.7 Описание карточки “Проведение анализа рынка фитнес-приложений”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.8 Описание карточки “Исследование потребностей целевой аудитории”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.9 Описание карточки “Создание пользовательского опыта (UX)”

Изображение выглядит как текст, снимок экрана, программное обеспечение, дизайн

Автоматически созданное описание

Рис.10 Описание карточки “Разработка пользовательского интерфейса (UI)”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.11 Описание карточки “Проведение тестирования прототипов”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.12 Описание карточки “Верстка адаптивных макетов приложения”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.13 Описание карточки “Разработка функции для съемки фотографий еды и их анализа”

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рис.14 Описание карточки “Интеграция с базой данных пищевых продуктов”