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 фитнес-трекера»

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

**Содержание**

[Введение 2](#_Toc166387626)

[1. Методология исследования 4](#_Toc166387627)

[1.1. Основные принципы Scrum 4](#_Toc166387628)

[1.2 Основные принципы Extreme Programming (XP) 8](#_Toc166387629)

[1.3 Преимущества и недостатки каждой из методологий 14](#_Toc166387630)

[1.4 Описание проекта ПО для фитнес-трекера и его особенностей 15](#_Toc166387631)

[2. Сравнительный анализ Scrum и XP 18](#_Toc166387632)

[2.1 Применение Scrum в рамках проекта 18](#_Toc166387633)

[2.1.1 Описание команды 18](#_Toc166387634)

[2.1.2 Персона пользователя (User Persona) 20](#_Toc166387635)

[2.1.3 Пользовательские истории (User Stories) 22](#_Toc166387636)

[2.1.4 Формирование Backlog продукта 24](#_Toc166387637)

[2.1.5 Планирование первого спринта и скрам-доска 30](#_Toc166387638)

[2.1.6 Диаграмма выгорания задач 40](#_Toc166387639)

[2.1.7 Выводы по применению Scrum в рамках проекта 44](#_Toc166387640)

[2.2 Применение XP в рамках проекта 45](#_Toc166387641)

[2.2.1 Описание команды 45](#_Toc166387642)

[2.2.2 Планирование релизов 46](#_Toc166387643)

[2.2.3 Планирование итераций 55](#_Toc166387644)

[2.2.4 Выводы по применению XP в рамках проекта 56](#_Toc166387645)

[2.3 Сравнительный анализ результатов применения Scrum и XP 57](#_Toc166387646)

[2.3.1 Преимущества применения Scrum 57](#_Toc166387647)

[2.3.2 Преимущества применения XP 58](#_Toc166387648)

[2.3.3 Недостатки и ограничения методологий 59](#_Toc166387649)

[2.3.4 Соответствие методологий специфике проекта 60](#_Toc166387650)

[Заключение 61](#_Toc166387651)

[Список использованных источников 62](#_Toc166387652)

# **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-трекера, 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-трекеров, 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-трекера с учетом их специфики и требований рынка. Таким образом, **объектом** **исследования** является процесс разработки программного обеспечения для фитнес-трекера, 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-трекеров 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 фитнес-трекера, 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-трекеров.

# **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 спринтами. Во время каждого спринта команда сосредотачивается на выполнении определенного набора задач, которые способствуют достижению общей цели - созданию готового продукта. После завершения этапа 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.
   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)[[1]](#footnote-1): 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)[[2]](#footnote-2): 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.
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)[[3]](#footnote-3): 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[[4]](#footnote-4) – 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 [[5]](#footnote-5)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**

Таблица 1 - Преимущества и недостатки методологий Scrum и XP

|  |  |  |
| --- | --- | --- |
| **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 | + | - |

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 фитнес-трекера 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.[[6]](#footnote-6)

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 фитнес 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[[7]](#footnote-7), 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[[8]](#footnote-8) 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. Роман, тестировщик:

* Написание и запуск автоматизированных тестов для проверки функциональности приложения;
* Проведение ручного тестирования новых функций и исправленных ошибок;
* Оценка качества и стабильности 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;

### **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.



Рисунок 1 – User persona 1

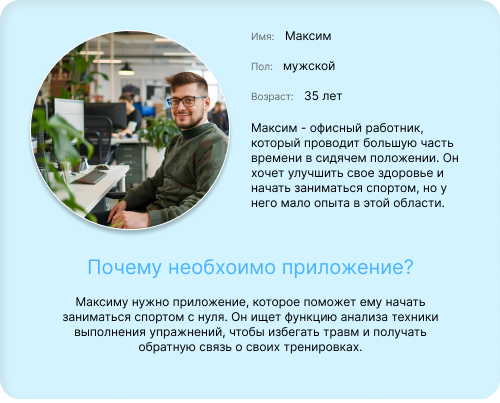


Рисунок 2 – User persona 2

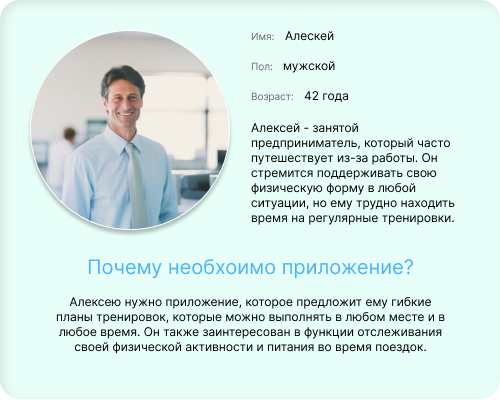


Рисунок 3 – User persona 3

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

User Story[[9]](#footnote-9), 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 итоге станут его пользователем или покупателем.

В основной части User Story обычно указываются:

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

Сделать обязательно:

1. Как пользователь, я хочу иметь возможность отслеживать свои ежедневные физические активности, чтобы следить за своими тренировками и достигать своих целей фитнеса.
2. Как пользователь, я хочу иметь функцию анализа выполнения упражнений, чтобы получать обратную связь о моей технике и избегать возможных травм во время тренировок.
3. Как пользователь, я хочу иметь возможность делать фотографии еды, чтобы получать информацию о калорийности и рекомендации по улучшению питания.
4. Как пользователь, я хочу получать персонализированные планы тренировок, основанные на моих целях и физической подготовке, чтобы эффективно тренироваться и достигать результатов.
5. Как пользователь, я хочу иметь возможность видеть статистику своих тренировок и прогресса в достижении целей, чтобы мотивироваться и отслеживать свой прогресс.
6. Как пользователь, я хочу иметь возможность сохранять свои любимые тренировки и программы, чтобы мог быстро находить их и использовать повторно.
7. Как пользователь, я хочу иметь доступ к библиотеке упражнений с инструкциями и видеороликами, чтобы повторять упражнения.

Хорошо бы сделать:

1. Как пользователь, я хочу иметь функцию интеграции с социальными сетями, чтобы делиться своими достижениями с друзьями и получать поддержку от сообщества.
2. Как пользователь, я хочу иметь функцию напоминаний о тренировках и достижении целей, чтобы следить за своей активностью.
3. Как пользователь, я хочу иметь возможность создавать и управлять групповыми тренировками, чтобы тренироваться с друзьями онлайн.

Можно будет сделать:

1. Как пользователь, я хочу иметь доступ к информации о фитнес-мероприятиях и тренировках в моем районе, чтобы их посещать.
2. Как пользователь, я хочу иметь функцию отслеживания моего пульса и других биометрических показателей, чтобы следить за своим здоровьем.
3. Как пользователь, я хочу иметь доступ к тренировкам от профессиональных тренеров, чтобы посещать их онлайн занятия.



Рисунок 4 – User Stories Map

### **2.1.4 Формирование Backlog продукта**

Backlog продукта формируется на основе User Stories. Приведу в пример переход от User Story к задаче.

User story: Как пользователь, я хочу иметь доступ к библиотеке упражнений с инструкциями и видеороликами, чтобы повторять упражнения.

Задача: Разработка библиотеки упражнений.

Подзадачи:

1. Провести анализ требований к базе данных упражнений;
2. Проектирование базы данных;
3. Создание таблиц и связей;
4. Наполнение базы данных;
5. Создание API для доступа к базе данных;
6. Создание пользовательского опыта (UX);
7. Разработка пользовательского интерфейса (UI);
8. Проведение тестирования прототипов дизайна;
9. Верстка адаптивных макетов приложения;
10. Вывести в раздел “Упражнения” все записи из базы данных;
11. Обеспечить возможность просмотра каждой записи;
12. Реализовать поиск и фильтрацию по упражнениям.

Таблица 2 – Backlog продукта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Задача | Важность | Предварительная оценка[[10]](#footnote-10) | Критерий приемки | Примечания |
| 1 | Проведение анализа рынка фитнес-приложений | 9 | 2 | Просмотр отчета или презентации с результатами анализа | Определение конкурентных преимуществ и возможностей для уникального предложения |
| 2 | Исследование потребностей целевой аудитории и их предпочтений в функционале приложений | 9 | 2 | Проведение опроса целевой аудитории или анализ отзывов о существующих продуктах | Выявление основных потребностей и предпочтений пользователей для создания ценного функционала |
| 3 | Создание пользовательского опыта (UX) | 10 | 8 | Демонстрация дизайн-макетов. Возможность взаимодействовать с элементами интерфейса и легко находить необходимую информацию и функционал. | Создание удобного и интуитивно понятного пользовательского опыта для максимального комфорта пользователей |
| 4 | Разработка пользовательского интерфейса (UI) | 9 | 4 | Представление макетов интерфейса. Возможность переходить в любой из разделов приложения, взаимодействовать с красиво оформленными элементами интерфейса, которые легко воспринимаются. | Создание привлекательного и удобного визуального оформления приложения |
| 5 | Проведение тестирования прототипов дизайна | 8 | 2 | Проведение пользовательских тестов интерфейсов, сбор обратной связи об опыте использования. | Выявление проблем и улучшение пользовательского опыта и интерфейса на основе обратной связи от пользователей |
| 6 | Верстка адаптивных макетов приложения | 8 | 8 | Приложение автоматически адаптируется к различным разрешениям экранов, обеспечивая удобное отображение и использование. | Обеспечение корректного отображения приложения на различных устройствах и экранах |
| 7 | Разработка функции для съемки фотографий еды и анализа калорийности | 9 | 8 | Пользователь переходит в раздел "Питание" и выбирает опцию "Анализ продуктов".  После съемки фотографии, приложение проводит анализ и отображает результаты о калорийности и составе продукта. | Обеспечение возможности съемки и анализа фотографий еды с использованием ИИ |
| 8 | Интеграция с базой данных пищевых продуктов | 9 | 2 | Пользователь переходит в раздел "Питание".  Здесь доступен поиск и просмотр информации о калорийности и питательных веществах продуктов. | Обеспечение доступа к актуальным данным о пищевой ценности продуктов |
| 9 | Разработка библиотеки упражнений | 8 | 2 | Пользователь открывает раздел "Тренировки" в главном меню.  Здесь доступен список инструкций и видеороликов по различным упражнениям для занятий фитнесом. | Обеспечение доступа к богатой базе данных с упражнениями и инструкциями для пользователей |
| 10 | Разработка функции анализа техники выполнения упражнений | 9 | 8 | В разделе “Тренировки”, после выбора определенного упражнения, пользователь может нажать кнопку “Начать тренировку” и предоставить доступ к камере. После начала выполнения упражнения, камера смартфона анализирует технику, а приложение дает рекомендации. | Обеспечение возможности анализа техники выполнения упражнений с помощью камеры смартфона |
| 11 | Создание алгоритмов для распознавания различных аспектов техники выполнения упражнений | 9 | 8 | Тестирование различных алгоритмов распознавания и анализа для выявления наилучшего и его последующего улучшения. | Обеспечение точного распознавания различных аспектов техники выполнения упражнений |
| 12 | Разработка функционала для отображения статистики тренировок и прогресса пользователей | 8 | 4 | Пользователь переходит в раздел "Прогресс" в главном меню.  Здесь доступны данные о выполненных тренировках, достижениях и прогрессе пользователя. | Обеспечение доступа к статистике тренировок и прогресса пользователей на основе их активности |
| 13 | Разработка функционала для отслеживания физической активности | 8 | 4 | Пользователь открывает раздел "Активность" в главном меню.  После активации функции отслеживания, приложение автоматически собирает данные о физической активности. | Обеспечение точного отслеживания физической активности пользователей |
| 14 | Интеграция с API устройств для сбора данных о физической активности | 8 | 4 | Пользователь переходит в настройки приложения и выбирает раздел "Настройки устройств".  Здесь осуществляется подключение к устройствам (например, фитнес-трекерам), и приложение начинает собирать данные. | Обеспечение корректной интеграции с API устройств для сбора данных о физической активности |
| 15 | Создание алгоритмов для генерации персонализированных тренировочных программ | 9 | 8 | После указания целей и характеристик в разделе “Профиль”, приложение создает персонализированную программу и отображает ее в разделе “Тренировки”. | Обеспечение создания эффективных и персонализированных тренировочных программ на основе данных пользователя |
| 16 | Возможность установки целей и отслеживания их выполнения | 8 | 2 | Пользователь переходит в раздел "Цели" в Профиле. Здесь он устанавливает цели (например, сжигание определенного количества калорий) и отслеживает их выполнение. | Обеспечение возможности установки целей и отслеживания прогресса их выполнения пользователем |
| 17 | Создание функции для сохранения любимых тренировок и программ | 7 | 1 | В разделе “Тренировки” пользователь может нажать на иконку сердца, приложение добавит выбранный элемент в раздел “Избранное”. | Обеспечение возможности сохранения и последующего доступа к любимым тренировкам и программам |
| 18 | Разработка функционала для интеграции с социальными сетями | 7 | 4 | Пользователь переходит в раздел "Общение" в главном меню.  Здесь осуществляется подключение к аккаунтам в социальных сетях и возможность обмена информацией. | Обеспечение корректной интеграции с популярными социальными сетями |
| 19 | Создание функции для установки напоминаний о тренировках и достижении целей | 8 | 2 | Пользователь переходит в раздел "Настройки" ->  "Уведомления".  Здесь он устанавливает временные интервалы и вид уведомлений, которые хочет получать. | Обеспечение возможности установки и получения уведомлений о тренировках и достижении целей |
| 20 | Разработка функционала для управления групповыми тренировками | 8 | 8 | Пользователь переходит в раздел "Тренировки"  -> "Тренировки с друзьями".  Здесь он создает новую групповую тренировку, выбирает участников и отправляет приглашения. После этого появляется возможность провести тренировку онлайн в видео формате. | Обеспечение возможности создания и управления групповыми тренировками с приглашением друзей |
| 21 | Интеграция с базой данных фитнес-мероприятий и тренировок | 7 | 2 | Пользователь открывает раздел "Мероприятия".  Здесь доступна информация о предстоящих событиях, которые соответствуют интересам пользователя. | Обеспечение доступа к актуальным данным о мероприятиях и тренировках в районе пользователя |
| 22 | Разработка функции для отслеживания биометрических показателей | 9 | 4 | Пользователь переходит в раздел "Профиль" -> "Здоровье".  Здесь он активирует функцию отслеживания пульса и других показателей с помощью ранее подключенных устройств в настройках приложения. | Обеспечение возможности отслеживания биометрических показателей здоровья пользователей с помощью устройств трекинга |
| 23 | Создание возможности доступа к тренировкам от профессиональных тренеров (онлайн или в записи) | 8 | 4 | Пользователь переходит в раздел "Тренировки"  -> "Онлайн занятия с тренером".  Здесь доступен выбор тренировок и программ от профессиональных тренеров для онлайн-занятий. | Обеспечение  доступа к высококачественным тренировкам от профессиональных тренеров |
| 24 | Разработка модели монетизации приложения | 9 | 4 | Представление модели монетизации. Пользователь переходит в раздел "Профиль" -> "Подписки", где может ознакомиться с доступными опциями подписок, рекламы или покупки дополнительного контента. | Выбор эффективной модели монетизации для максимизации дохода приложения |
| 25 | Создание стратегии продвижения и привлечения пользователей | 10 | 8 | Представление стратегии | Разработка эффективной стратегии маркетинга и привлечения пользователей для максимизации дохода приложения |
| 26 | Внедрение платежных систем | 9 | 4 | Пользователь переходит в раздел "Профиль" -> "Подписки", выбирает интересующую подписку, подходящий способ оплаты и осуществляет покупку. | Обеспечение безопасных и удобных платежных транзакций для пользователей |

### **2.1.5** **Планирование первого спринта и скрам-доска**

В ходе совещания по планированию первого спринта будет определена цель предстоящего спринта и выбор определенного количества задач из Product Backlog, 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.

В ходе собрания по планированию первого спринта были приняты следующие решения:

Команда определила целью первого спринта реализацию разработки основного функционала приложения, который включает в себя возможность анализа фотографий еды и интеграцию с базой данных пищевых продуктов, а также базовый пользовательский интерфейс для взаимодействия с этими функциями.

Команда спрогнозировала, что за первый спринт возможно выполнить первые 8 задач из верхней части бэклога, учитывая имеющиеся ресурсы и временные ограничения.

Критерии завершенности для каждой задачи соответствуют столбцу “Критерий приемки”, только после их выполнения задача будет считаться завершенной.

Прогнозируемая производительность = 2 + 2 + 8 + 4 + 2 + 8 + 8 + 2 = 36

Таким образом, в результате совещания, каждый член команды понимает свои обязанности и ответственность за выполнение конкретных задач в рамках первого спринта. Было достигнуто согласие относительно цели и плана действий на первый спринт, что обеспечивает ясное понимание у всех участников команды и помогает в реализации проекта в соответствии с установленными целями и сроками.

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

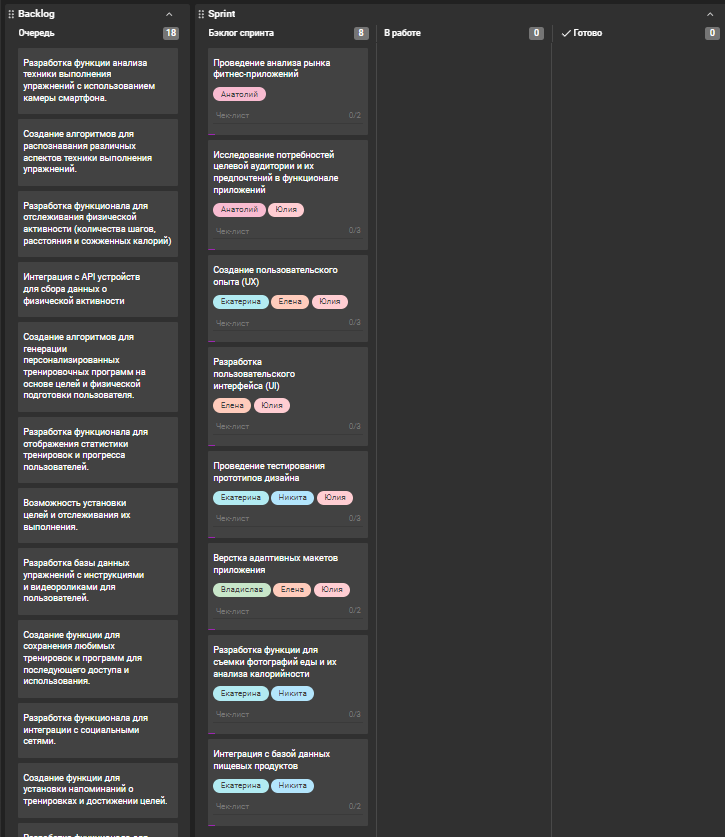


Рисунок 5 – Скрам-доска проекта

Далее представлены подробные описания карточек с задачами:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 6 – Описание карточки “Проведение анализа рынка фитнес-приложений”

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 7 – Описание карточки “Исследование потребностей целевой аудитории и их предпочтений в функционале приложений”

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 8 – Описание карточки “Создание пользовательского опыта (UX)”

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 9 – Описание карточки “Разработка пользовательского интерфейса (UI)”

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

Автоматически созданное описание

Рисунок 10 – Описание карточки “Проведение тестирования прототипов дизайна”

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 11 – Описание карточки “Верстка адаптивных макетов приложения”

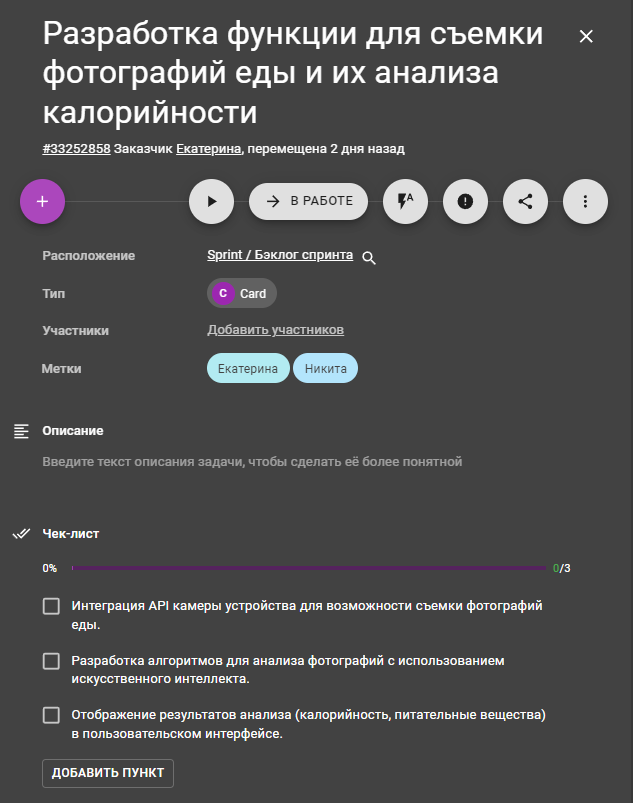


Рисунок 12 – Описание карточки “Разработка функции для съемки фотографий еды и их анализа калорийности”

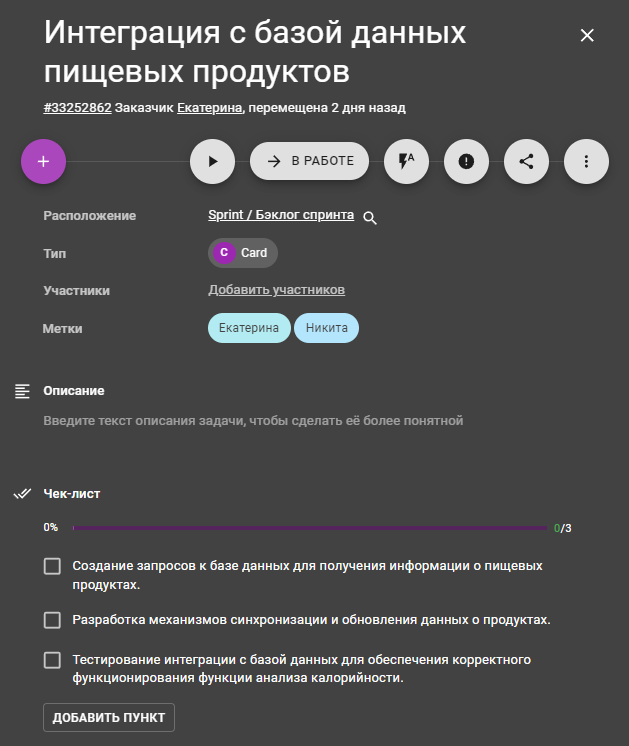


Рисунок 13 – Описание карточки “Интеграция с базой данных пищевых продуктов”

### **2.1.6 Диаграмма выгорания задач**

Для построения диаграммы выгорания[[11]](#footnote-11), все задачи из Backlog были выписаны на отдельный лист книги Exel. Голубым цветом выделены задачи, входящие в первый спринт.

Изображение выглядит как текст, электроника, снимок экрана, число

Автоматически созданное описание

Рисунок 14 – Лист со списком задач в Microsoft Excel

Далее был рассчитан объем работ, задана длительность спринта (в днях) и установлена дата отчета (равна дате последнего выполнения задачи).

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 15 – Основные параметры для расчетов

После этого, я произвела необходимые для графика расчеты. По дням было найдена сумма сделанных работ (по оценкам), сколько всего сделано работ к определенному дню и объем оставшейся работы.

Изображение выглядит как текст, число, линия, Шрифт

Автоматически созданное описание

Рисунок 16 – Формула для расчета столбца “Сделано работы”

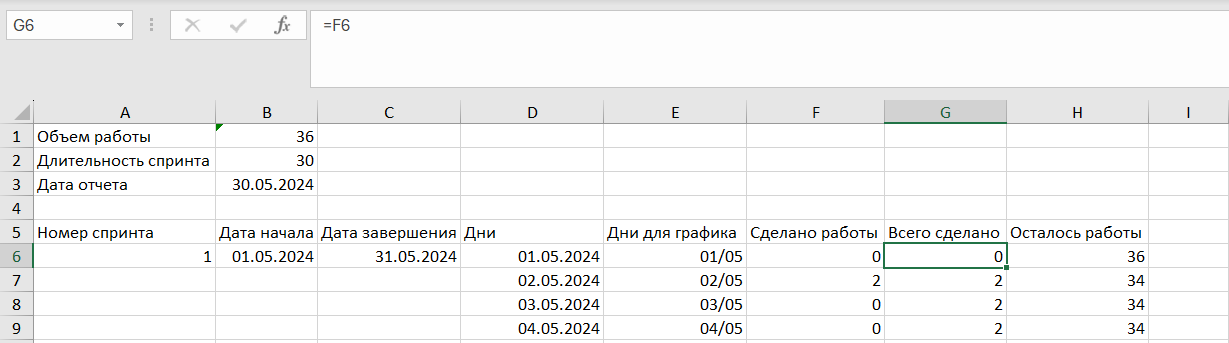


Рисунок 17 – Формула для расчета первой строки столбца “Всего сделано”

Изображение выглядит как текст, число, Шрифт, линия

Автоматически созданное описание

Рисунок 18 – Формула для расчета столбца “Всего сделано”

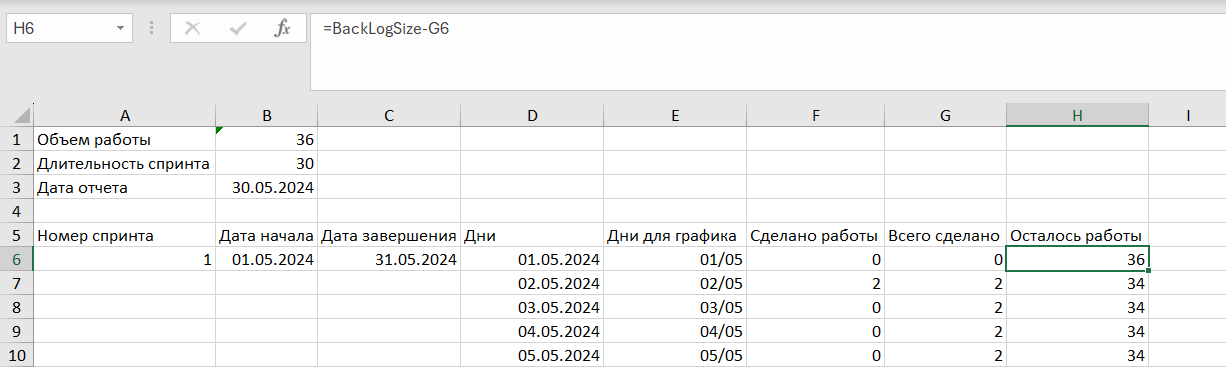


Рисунок 19 – Формула для расчета столбца “Осталось работы”

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 20 – Итоговый результат расчетов

Таким образом, у нас получилась итоговая диаграмма задач по прошествии первого спринта:

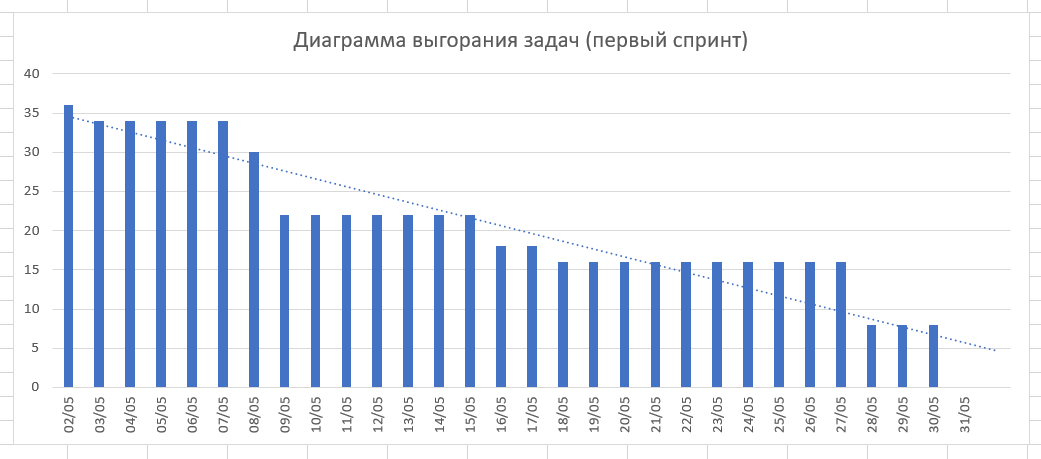


Рисунок 21 – Диаграмма выгорания задач

График отображает прогресс выполнения задач и позволяет следить за производительностью команды и сроками. На нем также видно линию тренда, опускающуюся, по мере выполнения задач, что свидетельствует о постепенном уменьшении объема оставшихся работ и приближении к завершению спринта.

Таким образом, на 30/05 все поставленные задачи на данный спринт были выполнены. Это означает, что смело можно проводить Sprint Review и Sprint Retrospective.

### **2.1.7 Выводы по применению Scrum в рамках проекта**

Применение методологии Scrum в рамках проекта ПО для фитнес-трекера стало положительным опытом в разработке.

Формирование четко определенной и разносторонней команды с определенными ролями и обязанностями помогло обеспечить ясное распределение ответственности и способствовало эффективной командной работе.

Создание детального описания целевой аудитории приложения помогло лучше понять ее потребности, предпочтения и проблемы, что послужило основой для формирования пользовательских историй, которые позволили сосредоточиться на реальных потребностях пользователей и определить функциональность, которая приоритетна для реализации.

Создание подробного Backlog продукта на основе пользовательских историй обеспечило ясное представление о задачах, которые требуется выполнить для достижения целей проекта.

Планирование первого спринта позволило команде четко определить цели и задачи на конкретный период времени. Использование скрам-доски обеспечило прозрачность в ходе работы и помогло команде отслеживать прогресс и эффективно распределять задачи.

Создание диаграммы выгорания задач помогло визуализировать прогресс работы команды, отслеживать выполнение задач и своевременно реагировать на любые проблемы или задержки.

В целом, применение методологии Scrum способствовало организации работы команды, повышению эффективности процессов разработки и обеспечило достижение поставленных целей проекта.

## **2.2 Применение XP в рамках проекта**

### **2.2.1 Описание команды**

Команда включает в себя следующих участников:

1. Анатолий, Аналитик:
   * + Анализ требований пользователей и бизнес-задач;
     + Помощь в определении приоритетов и составлении пользовательских историй;
     + Взаимодействие с заказчиком для уточнения требований и обратной связи.
2. Никита и Екатерина, Fullstack разработчики:

* Разработка и поддержка серверной части приложения;
* Обеспечение функциональности и безопасности бэкэнда;
* Разработка и поддержка пользовательского интерфейса приложения;
* Оптимизация интерфейса для обеспечения лучшего пользовательского опыта;

1. Роман, тестировщик:

* Написание и запуск автоматизированных тестов для проверки функциональности приложения;
* Проведение ручного тестирования новых функций и исправленных ошибок.
* Оценка качества и стабильности приложения.

1. Владислав, Верстальщик:

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

1. Елена, Дизайнер:

* Создание дизайн-макетов пользовательского интерфейса и элементов приложения;

### **2.2.2 Планирование релизов**

Работа над проектом по созданию программного обеспечения для фитнес-трекера, выполняемая в соответствии с методологией XP, начинается с этапа планирования. Этот процесс включает в себя два основных этапа: планирование релизов и планирование итераций.

Проект будет состоять из восьми основных релизов, каждый из которых запланирован на двухнедельный период. После завершения последнего основного релиза предусмотрены два технических релиза, предназначенных для устранения ошибок и доработки имеющегося функционала. Эти технические релизы также будут осуществляться каждые две недели. В процессе планирования релизов команда взаимодействует с заказчиком, который определяет, какие функции должны быть включены в каждый релиз.

После совещания всей команды разработки с заказчиком, были выделены следующие приоритетные задачи для первого релиза[[12]](#footnote-12):

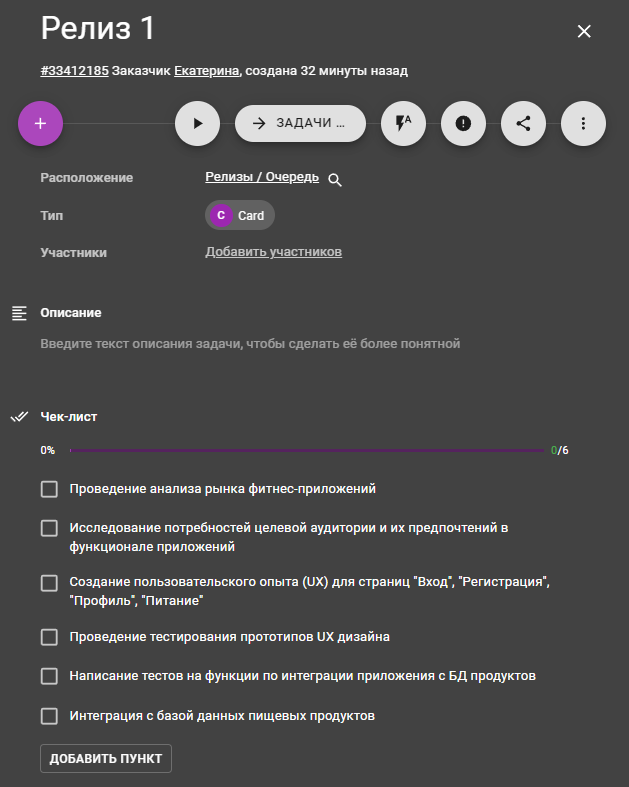


Рисунок 22 – Список задач на первый релиз

Таким образом, была создана доска задач с подробным описанием каждой из них:

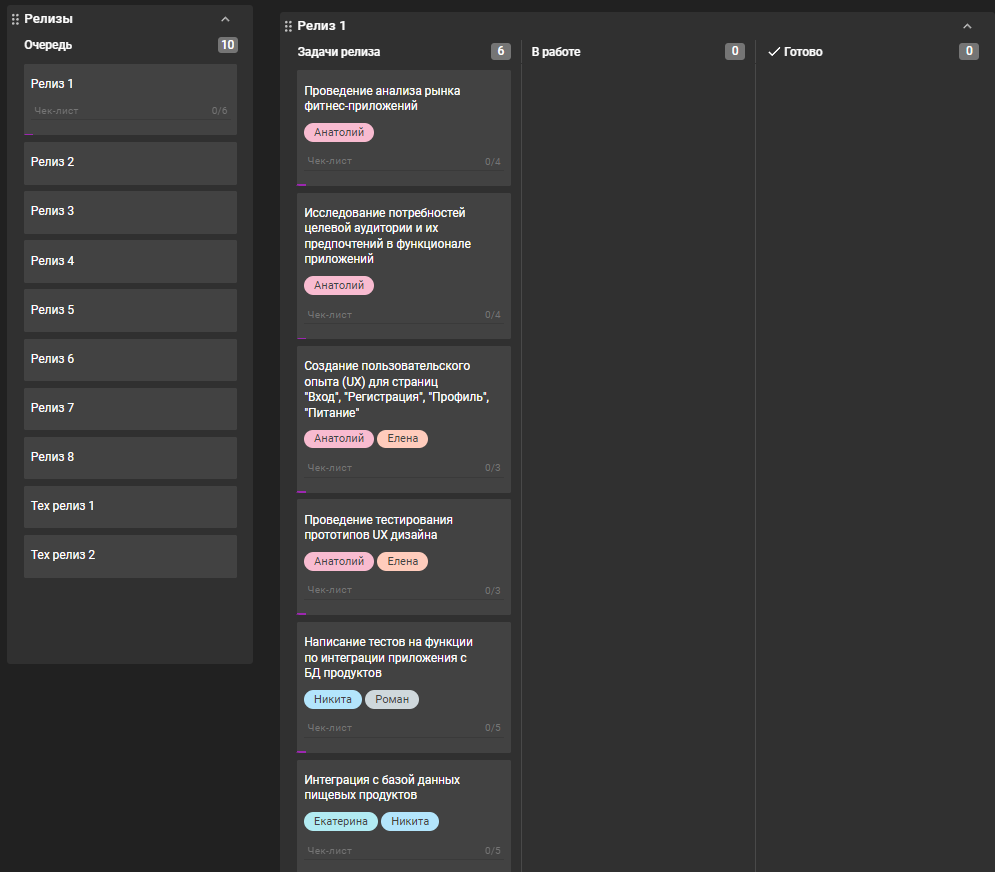


Рисунок 23 – Доска проекта по методологии XP

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 24 – Описание карточки “Проведение анализа рынка фитнес-приложений”

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 25 – Описание карточки “Исследование потребностей целевой аудитории и их предпочтений в функционале приложений ”

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 26 – Описание карточки “ Создание пользовательского опыта (UX) для страниц "Вход", "Регистрация", "Профиль", "Питание"”

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 27 – Описание карточки “ Проведение тестирования прототипов UX дизайна”

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 28 – Описание карточки “Написание тестов на функции по интеграции приложения с БД продуктов”

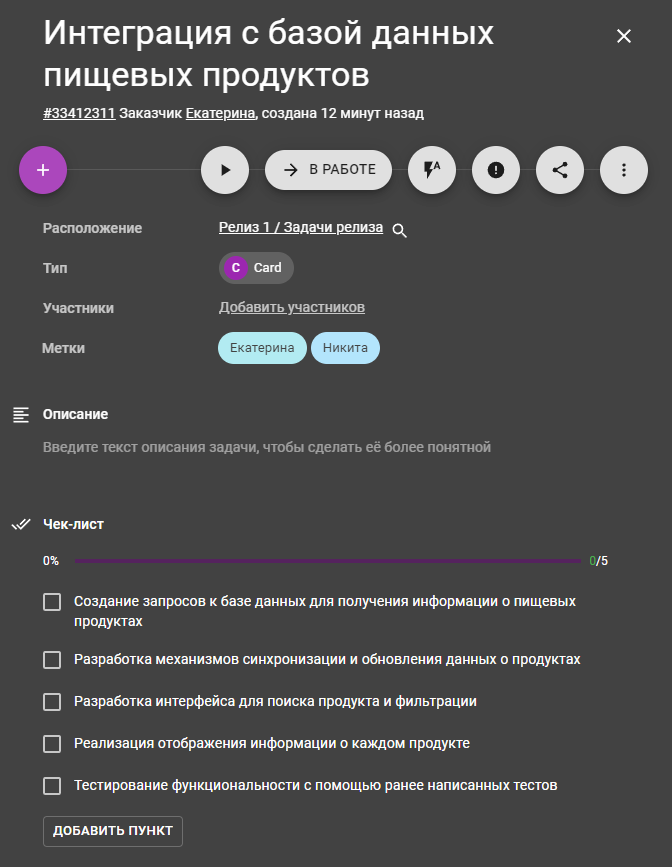


Рисунок 29 – Описание карточки “Интеграция с базой данных пищевых продуктов”

### **2.2.3 Планирование итераций**

Планирование итераций[[13]](#footnote-13) проходит каждую неделю и представляет собой регулярное событие, на котором команда определяет задачи, которые будут выполнены в ближайшей неделе. Важным аспектом этого процесса является возможность заказчика вносить правки в задачи на этом этапе. Такой подход обеспечивает гибкость процесса разработки и позволяет эффективно реагировать на изменения в требованиях или приоритетах.

На совещании было решено разделить задачи первого релиза на две итерации с учетом возможности их выполнения за каждую неделю. Распределение задач выглядит следующим образом (на доску добавлены метки):

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

Автоматически созданное описание

Рисунок 30 – Распределение задач первого релиза на две итерации

### **2.2.4 Выводы по применению XP в рамках проекта**

Применение экстремального программирования (XP) в рамках проекта принесло несколько значительных преимуществ и улучшений:

Регулярное взаимодействие с заказчиком позволило лучше понимать его потребности и ожидания, а также быстрее реагировать на изменения в требованиях.

Разбиение проекта на релизы и итерации позволило эффективнее планировать работу, управлять рисками и достигать поставленных целей. Это также обеспечило более гибкую адаптацию к изменяющимся требованиям.

Практика парного программирования способствовала повышению качества кода, обмену знаниями и идеями между участниками команды, а также улучшила коммуникацию внутри коллектива.

Подход разработки через тестирование (Test-Driven Development, TDD) позволил обнаруживать ошибки на ранних стадиях разработки, улучшая качество и надежность продукта, а также сокращая время на отладку.

Стандартизация оформления кода способствовала повышению читаемости, поддержки и восприятия кода, а также улучшила совместную работу разработчиков.

Применение небольших и частых релизов позволило демонстрировать результаты работы заказчику как можно быстрее и получать обратную связь, что содействовало улучшению продукта и удовлетворению потребностей заказчика.

В целом, применение методологии XP сделало процесс разработки более гибким, прозрачным и адаптивным, что позволило достичь лучших результатов в разработке проекта ПО для фитнес-трекера.

## **2.3 Сравнительный анализ результатов применения Scrum и XP**

**2.3.1 Преимущества применения Scrum**

Методология Scrum, со своим упором на гибкость и прозрачность, привнесла несколько существенных преимуществ:

Формирование отчетливо определенной команды с ролями и обязанностями помогло обеспечить действенную командную работу и четкое понимание своей роли у каждого участника.

Организация подробного Backlog продукта на основе пользовательских историй обеспечило ясное представление о задачах, что позволило команде сконцентрироваться на ключевых задачах и достижении целей проекта.

Использование скрам-доски и диаграммы выгорания задач обеспечило прозрачность в ходе работы и помогло команде эффективно прослеживать прогресс и распределять задачи.

**2.3.2 Преимущества применения XP**

Применение методологии XP также принесло ряд значимых преимуществ:

Возможность частого взаимодействия с заказчиком позволило лучше понимать его потребности и ожидания, что содействовало более точной реализации требований.

Парное программирование способствовало повышению качества кода, обмену знаниями и улучшению коммуникации внутри команды.

Подход разработки через тестирование позволил обнаруживать ошибки на ранних стадиях разработки, что сократило время на отладку и улучшило качество продукта.

**2.3.3 Недостатки и ограничения методологий**

В методологии Scrum, несмотря на то, что уделяется значительное внимание организации команды и планированию проекта, нередко недостаточно времени отводится техническим аспектам разработки. В результате разработчики могут столкнуться с внезапными техническими проблемами или затруднениями в ходе выполнения задач, что в итоге может затормозить процесс разработки и повлиять на уровень качества конечного продукта.

В XP практика парного программирования играет важную роль в обеспечении качества кода и обмена знаниями внутри команды. Однако в некоторых случаях эта практика может оказаться неэффективной из-за различий в уровнях квалификации и стандартах оформления кодов членов команды. В результате чего парное программирование может стать препятствием для продуктивной работы, поскольку могут возникнуть недопонимание и противоречия в принятии решений. Конечно, при разном уровне квалификации, парное программирование позволяет обучиться одному из программистов. Но, в рамках сжатых сроков, это может стать проблемой, так как будет затормаживать процесс разработки.

**2.3.4 Соответствие методологий специфике проекта**

В процессе реализации проекта выяснилось, что под специфику данного проекта успешно подходят обе методологии, Scrum и XP, но с различными акцентами и преимуществами.

Scrum оказался эффективным методом организации командной работы и планирования проекта в комплексе. Благодаря четко определенным ролям и обязанностям в команде, а также использованию Scrum-доски для наглядного представления задач и отслеживания прогресса, удалось обеспечить четкое разделение обязанностей и повысить эффективность командной работы. Эффективными инструментами для своевременного достижения целей проекта оказались также планирование спринтов и взаимодействие с заказчиком для определения приоритетности задач.

С другой же стороны, XP принесла большую пользу в части технической реализации и качества продукта. Такие практики, как парное программирование, разработка через тестирование (TDD), стандартизация оформления кода, небольшие и частые релизы способствовали повышению качества кода, раннему обнаружению ошибок и повышению эффективности процесса разработки. Уделяя особое внимание техническим аспектам, XP обеспечила более надежный и стабильный конечный продукт.

# **Заключение**

Исследование особенностей применения методологий Scrum и XP в контексте разработки программного обеспечения для фитнес-трекера дало ценное понимание преимуществ и ограничений использования каждого из подходов. Обе методологии продемонстрировали свою эффективность в определенных аспектах управления проектами и технической реализации, что подтверждает их актуальность в современной практике разработки ПО.

При этом, по результатам проведенного исследования, было установлено, что ни одна из методологий не представляет собой универсального решения. Поэтому мною было решено, что применять обе методологии в симбиозе, может оказаться оптимальным вариантом. Проектное управление с помощью Scrum обеспечит структурирование, прозрачность и высокую эффективность процессов, а техническая реализация с использованием XP улучшит качество и надежность продукта.

Таким образом, поставленные цели и задачи исследования были полностью выполнены. Полученные результаты представляют ценность для разработчиков, менеджеров проектов и всех заинтересованных сторон в области разработки программного обеспечения для фитнес-трекеров. В будущем, опираясь на результаты исследования, можно и дальше усовершенствовать и максимально оптимизировать процесс разработки ПО, что позволит повысить его качество и степень удовлетворенности пользователей и заказчиков.

# **Список использованных источников**

1. Джефф Сазерленд SCRUM. Революционный метод управления проектами – Манн , Иванов и Фербер, 2016. – с. 5-6.
2. Корягина Ю.В., Копанев А.Н., Нопин С.В., Абуталимова С.М. Анализ онлайн систем тестирования для спорта и фитнеса // Современные вопросы биомедицины – 2020. – Т. 4, № 4. – с. 101-107.
3. Процессы методологии SCRUM. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SCRUM (дата обращения 03.04.2024).
4. SCRUM: Понимание и применение фреймворка. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/otr/articles/553670/ (дата обращения 05.04.2024).
5. Метод управления проектами Scrum: что это такое и как его применять. – [Электронный ресурс]. – Режим доступа: https://www.consultant.ru/edu/student/useful\_advice/metod\_scrum/ (дата обращения 05.04.2024).
6. Экстремальные практики программирования: что, как и зачем. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/yandex\_praktikum/articles/787084/ (дата обращения 11.04.2024).
7. Методики и принципы экстремального программирования. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/197760/ (дата обращения 11.04.2024).
8. Разработка фитнес-приложения: функциональность, цена, сроки. – [Электронный ресурс]. – Режим доступа: https://livetyping.com/ru/razrabotka-fitnes-prilozheniya-funkcionalnost-cena-sroki-razrabotki (дата обращения 25.04.2024).
9. В России вырос спрос на спортивные приложения. – [Электронный ресурс]. – Режим доступа: https://secrets.tinkoff.ru/novosti/spros-na-online-fitnes/ (дата обращения 02.05.2024).
10. Экстремальное программирование. – [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2000/01-02/178193 (дата обращения 10.05.2024).

1. Процессы методологии SCRUM. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SCRUM (дата обращения 03.04.2024). [↑](#footnote-ref-1)
2. SCRUM: Понимание и применение фреймворка. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/otr/articles/553670/ (дата обращения 05.04.2024). [↑](#footnote-ref-2)
3. Джефф Сазерленд SCRUM. Революционный метод управления проектами – Манн , Иванов и Фербер, 2016. – с. 5-6. [↑](#footnote-ref-3)
4. Экстремальные практики программирования: что, как и зачем. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/yandex\_praktikum/articles/787084/ (дата обращения 11.04.2024). [↑](#footnote-ref-4)
5. Экстремальные практики программирования: что, как и зачем. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/companies/yandex\_praktikum/articles/787084/ (дата обращения 11.04.2024). [↑](#footnote-ref-5)
6. В России вырос спрос на спортивные приложения. – [Электронный ресурс]. – Режим доступа: https://secrets.tinkoff.ru/novosti/spros-na-online-fitnes/ (дата обращения 02.05.2024). [↑](#footnote-ref-6)
7. Разработка фитнес-приложения: функциональность, цена, сроки. – [Электронный ресурс]. – Режим доступа: https://livetyping.com/ru/razrabotka-fitnes-prilozheniya-funkcionalnost-cena-sroki-razrabotki (дата обращения 25.04.2024). [↑](#footnote-ref-7)
8. Метод управления проектами Scrum: что это такое и как его применять. – [Электронный ресурс]. – Режим доступа: https://www.consultant.ru/edu/student/useful\_advice/metod\_scrum/ (дата обращения 05.04.2024). [↑](#footnote-ref-8)
9. Процессы методологии SCRUM. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SCRUM (дата обращения 03.04.2024). [↑](#footnote-ref-9)
10. Процессы методологии SCRUM. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SCRUM (дата обращения 03.04.2024). [↑](#footnote-ref-10)
11. Процессы методологии SCRUM. – [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SCRUM (дата обращения 03.04.2024). [↑](#footnote-ref-11)
12. Экстремальное программирование. – [Электронный ресурс]. – Режим доступа: https://www.osp.ru/os/2000/01-02/178193 (дата обращения 10.05.2024). [↑](#footnote-ref-12)
13. Методики и принципы экстремального программирования. – [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/197760/ (дата обращения 11.04.2024). [↑](#footnote-ref-13)