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
   5. Определение критериев сравнения между Scrum и XP
2. Сравнительный анализ Scrum и XP
   1. Применение Scrum в рамках проекта
      1. Описание процесса разработки по методологии Scrum
      2. Выявление преимуществ и недостатков Scrum в контексте проекта
   2. Применение XP в рамках проекта
      1. Описание процесса разработки по методологии XP
      2. Выявление преимуществ и недостатков XP в контексте проекта
   3. Сравнительный анализ результатов применения Scrum и XP
      1. Сопоставление критериев
      2. Оценка соответствия каждой из методологий специфике проекта

Заключение

Список использованных источников

Приложения

# **Введение**

В современном мире программное обеспечение становится неотъемлемой частью повседневной жизни, внося значительный вклад в различные сферы деятельности человека. С развитием технологий и появлением новых потребностей пользователей, разработчики программного обеспечения сталкиваются с необходимостью эффективного управления процессом создания и обновления программных продуктов. В этом контексте Agile-методологии выходят на передний план, предлагая гибкие и адаптивные подходы к разработке, которые позволяют быстро реагировать на изменения и достигать лучших результатов.

Выбор подходящей Agile-методологии для конкретного проекта является неотъемлемой частью успеха его реализации. В связи с этим, **объектом исследования** нашей работы является процесс разработки программного обеспечения для фитнес-треккера, а **предметом** - методологии Scrum и XP, особенности их применения в контексте данного проекта, преимущества и недостатки, а также их соответствие специфике требований и задач, стоящих перед разработчиками в данной предметной области.

**Актуальность** данного исследования обусловлена постоянно растущим интересом к Agile-методологиям в области разработки программного обеспечения и необходимостью выбора наиболее подходящего подхода к управлению проектами в конкретных условиях. Проекты в области разработки программного обеспечения для фитнес-треккера, в силу своей специфики и динамичного характера, требуют особого внимания к процессу разработки и управления. Таким образом, результаты нашего исследования будут иметь прямое значение для практического опыта разработчиков и менеджеров проектов в данной сфере.

Научная проблема, лежащая в основе нашего исследования, заключается в определении оптимального подхода к управлению проектами в области разработки программного обеспечения для фитнес-треккера с учетом их специфики и требований рынка.

**Цель** данной курсовой работы – предоставить читателю полное и объективное представление о методологиях Scrum и Extreme Programming (XP), а также, путём проведения их сравнительного анализа, помочь в выборе наиболее подходящего подхода к управлению проектом в контексте разработки программного обеспечения для фитнес-треккера. Для достижения этой цели были выделены следующие **задачи**:

1. Изучение основных принципов и практик Agile методологий, а также их роли в разработке программного обеспечения.
2. Проведение обзора литературы по методологиям Scrum и XP, включая их основные принципы, преимущества и недостатки.
3. Выбор и описание конкретного проекта в области разработки ПО для фитнес-треккера, а также выделение его особенностей.
4. Определение критериев сравнения между Scrum и XP, учитывая специфику выбранного проекта.
5. Проведение сравнительного анализа применения Scrum и XP в рамках выбранного проекта, включая описание процесса разработки, выявление преимуществ и недостатков каждой методологии.
6. Формулирование выводов о применимости Scrum и XP в области разработки ПО для фитнес-треккера на основе результатов анализа.
7. Предложение рекомендаций по выбору наиболее подходящей Agile методологии для подобных проектов.

Таким образом, круг вопросов, на которые должна ответить наша курсовая работа, охватывает все аспекты сравнения методологий Scrum и XP и их применимости в контексте разработки программного обеспечения для фитнес-треккера.

# **Методология исследования**

Scrum представляет собой широко используемую методологию разработки программного обеспечения, ориентированную на гибкость и адаптивность. Ее основные принципы и практики были выработаны в начале 1990-х годов Кеном Швабером и Джеффом Сазерлендом и с тех пор завоевали широкое признание в индустрии.

В основе Scrum лежит итеративный и инкрементальный подход к разработке. Итеративный подход предполагает разбиение процесса разработки на небольшие, повторяющиеся циклы или итерации. Каждая итерация представляет собой полный цикл разработки, начиная с определения требований и заканчивая тестированием и релизом продукта. После каждой итерации команда получает обратную связь от заказчика или пользователя, что позволяет вносить корректировки и улучшения в следующие итерации. Такой подход позволяет команде быстро создавать рабочие версии продукта и реагировать на изменения в требованиях. Инкрементальный подход предполагает постепенное построение и улучшение продукта за счет добавления новых функций и возможностей на каждой итерации. Вместо того чтобы стремиться к созданию полностью завершенного продукта сразу, команда фокусируется на создании минимально жизнеспособного продукта (Minimum Viable Product, MVP), который постепенно дополняется новыми функциями и улучшениями. Это позволяет быстрее предоставить пользователю работающий продукт и получить обратную связь о его функциональности и удобстве использования.

Таким образом, в ходе разработки, проект разделяется на короткие временные отрезки, называемые спринтами, каждый из которых обычно длится от одной до четырех недель. Каждый спринт завершается созданием работоспособного продукта, что позволяет заказчику получать результаты на ранних стадиях разработки и вносить корректировки по мере необходимости.

## **1.1. Основные принципы Scrum**

Основные принципы Scrum могут быть разделены на несколько ключевых аспектов:

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.
   * Product Owner (Владелец продукта) представляет интересы заказчика и пользователей продукта. Его главная задача - определить требования к продукту и убедиться, что команда разработки понимает их. Product Owner отвечает за формирование и поддержание Product Backlog - списка всех задач и требований, которые должны быть выполнены для достижения целей проекта. Он также отвечает за приоритизацию задач в Product Backlog в соответствии с потребностями бизнеса и ожиданиями заказчика. В итоге Product Owner играет ключевую роль в обеспечении того, чтобы команда разработки создавала продукт, который наилучшим образом удовлетворяет потребности заказчика и пользователей.
   * Scrum Master (Мастер Scrum) – это человек, который обеспечивает правильное применение методологии Scrum в проекте. Его главная задача - помогать команде разработки использовать Scrum наилучшим образом и устранять препятствия, мешающие ее продуктивности. Scrum Master проводит обучение и коучинг членов команды по методологии Scrum, помогает им понять и применить принципы и практики Scrum. Он также служит защитником команды от внешних воздействий и давлений, чтобы команда могла работать эффективно и без прерываний.
   * Development Team (Команда разработки) – это группа специалистов, ответственных за создание продукта. Команда разработки является самоорганизующейся и автономной, что означает, что она сама принимает решения о том, каким образом выполнять свою работу и как достигать поставленных целей. Команда разработки должна обладать всеми необходимыми навыками и компетенциями для реализации задач из Product Backlog. Она работает коллективно, чтобы создавать новую работающую функциональность на протяжении каждого спринта.

Таким образом, эти три роли в Scrum взаимодействуют вместе, чтобы обеспечить успешное выполнение проекта. Product Owner определяет, что нужно сделать, Scrum Master обеспечивает, как это будет сделано, а Development Team выполняет задачи и создает результаты.

1. В Scrum используются различные артефакты, такие как Product Backlog, Sprint Backlog и Increment, 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.
   * Product Backlog (Бэклог продукта) представляет собой список всех задач, требований и идей, которые могут быть реализованы в рамках проекта. Этот список поддерживается и обновляется Product Owner'ом и содержит задачи, которые необходимо выполнить для достижения целей проекта. Задачи в Product Backlog описываются в виде User Stories или других формализованных элементов и часто сопровождаются приоритетами, оценками сложности и другой информацией. Product Backlog служит источником задач для планирования спринтов и определения направления развития продукта.
   * Sprint Backlog (Бэклог спринта) – это список задач, выбранных командой разработки для выполнения в рамках текущего спринта. Он формируется на основе элементов из Product Backlog, выбранных для реализации в рамках конкретного спринта. Sprint Backlog содержит подробное описание каждой задачи, включая требуемые действия, сроки выполнения и ответственных исполнителей. В процессе выполнения спринта Sprint Backlog может подвергаться изменениям в зависимости от появления новой информации или изменения условий.
   * Increment (Инкремент) представляет собой работающую версию продукта, которая достигается в конце каждого спринта. Каждый спринт направлен на создание нового Increment’а продукта, который может быть демонстрирован заказчику или конечным пользователям. Increment должен быть полностью завершенным и работоспособным, что означает, что он должен 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.

Таким образом, артефакты в методологии Scrum, такие как Product Backlog, Sprint Backlog и Increment, являются важными инструментами для управления процессом разработки и обеспечения прозрачности и понимания состояния проекта как самой командой разработки, так и заинтересованными сторонами.

1. Существуют регулярные события, такие как Sprint Planning, Daily Scrum, Sprint Review и Sprint Retrospective, которые обеспечивают прозрачность и позволяют команде оценивать свой прогресс и адаптировать процесс разработки.
   * Sprint Planning (Планирование спринта) – это событие, которое происходит в начале каждого спринта и ориентировано на определение целей и плана работы на следующий спринт. На этом мероприятии команда разработки, вместе с Product Owner'ом, обсуждает элементы из Product Backlog, определяет задачи, которые будут включены в Sprint Backlog, и устанавливает цели, которые они планируют достигнуть к концу спринта. Sprint Planning помогает команде понять, какие задачи требуют выполнения и как они будут достигнуты в течение спринта.
   * Daily Scrum (Ежедневный Scrum) – это короткое ежедневное событие, которое проводится каждый день в течение всего спринта. На Daily Scrum каждый член команды разработки отвечает на три вопроса: "Что я сделал с момента последнего Scrum?", "Что я планирую сделать к следующему Scrum?" и "Есть ли у меня какие-либо препятствия?". Это событие помогает команде оставаться в курсе текущего состояния проекта, обмениваться информацией и идентифицировать любые проблемы или препятствия, которые могут возникнуть на пути выполнения задач.
   * Sprint Review (Обзор спринта) – это событие, которое происходит в конце каждого спринта и ориентировано на представление выполненной работы заказчику или заинтересованным сторонам. На Sprint Review команда разработки демонстрирует работающий Increment продукта и обсуждает его с заказчиком, получая обратную связь и рекомендации для дальнейшего развития продукта. Sprint Review способствует прозрачности и обеспечивает возможность заказчику вносить изменения в продукт в соответствии с его потребностями и ожиданиями.
   * Sprint Retrospective (Ретроспектива спринта) – это событие, которое проводится после завершения каждого спринта и направлено на анализ процесса работы команды и выявление возможностей для улучшения. На Sprint Retrospective члены команды разработки обсуждают, что прошло хорошо в прошлом спринте, что можно улучшить, и какие действия могут быть предприняты для повышения эффективности работы в следующем спринте. Sprint Retrospective способствует постоянному улучшению процесса разработки и помогает команде обучаться на своем опыте.

Эти регулярные события в методологии Scrum обеспечивают прозрачность, позволяют команде оценивать свой прогресс и корректировать свою работу в соответствии с изменяющимися условиями и требованиями заказчика. Они помогают команде оставаться синхронизированной, обмениваться информацией и решать любые проблемы или препятствия, которые могут возникнуть на пути выполнения задач.

Исходя из этого, Scrum является мощным инструментом для управления проектами, который позволяет командам быстро адаптироваться к изменениям в требованиях и условиях рынка. Его принципы и практики нашли применение в различных отраслях и типах проектов, делая его одним из наиболее популярных методов разработки программного обеспечения в настоящее время.