МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И   
ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ**

**Кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Разработка программного проекта сервиса для управления задачами»

Обучающегося 3 курса

Моисеенко Павла Александрович

Научный руководитель:

Кандидат физико-математических наук,   
доцент кафедры ИТиЭО

Жуков Николай Николаевич

Санкт-Петербург

2020

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc59542887)

[1. ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 4](#_Toc59542888)

[1.1 Определение понятия 4](#_Toc59542889)

[1.2 Различные парадигмы и модели 4](#_Toc59542890)

[1.3 Инструменты и фреймворки гибкой методологии разработки 7](#_Toc59542891)

[ВЫВОД ИЗ I ГЛАВЫ 9](#_Toc59542892)

[2 ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ МЕТОДОЛОГИИ 10](#_Toc59542893)

[2.1 Техническое задание и диаграмма Ганта 10](#_Toc59542894)

[2.2 Инструменты управления проектом 10](#_Toc59542895)

[ВЫВОД ИЗ II ГЛАВЫ 10](#_Toc59542896)

[ЗАКЛЮЧЕНИЕ 11](#_Toc59542897)

[ЛИТЕРАТУРА 12](#_Toc59542898)

# ВВЕДЕНИЕ

Разработка программного обеспечения — это трудо- и времязатратный процесс. Чтобы это сделать максимально эффективно, стоит внедрить себе один из методов управления разработкой программного обеспечения. Сейчас появляется всё больше разработчиков, который разрабатываю ПО и которым нужны для этого подходящие им методы разработки. Поэтому мной были описаны различные методологии разработки программного обеспечения, для каких случаев они подходят и пример использования на основе моего проекта разработки программного проекта сервиса для управления задачами.

# 1. ПРОЦЕСС РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 1.1 Определение понятия

В программной инженерии процесс разработки программного обеспечения — это процесс разделения работы по разработке программного обеспечения на отдельные фазы для улучшения проектирования, управления продуктом и управления проектами. Он также известен как жизненный цикл разработки программного обеспечения (SDLC). Методология может включать предварительное определение конкретных результатов и артефактов, которые создаются и завершаются командой проекта для разработки или сопровождения приложения. Большинство современных процессов разработки можно скорее назвать гибкими. Но есть и другие методологии такие как водопад, прототипирование, итеративную и инкрементную разработку, спиральную разработку, быструю разработку приложений и экстремальное программирование. «Модель» жизненного цикла иногда считается более общим термином для категории методологий, а «процесс» разработки программного обеспечения — более конкретным термином, относящимся к конкретному процессу, выбранному конкретной организацией.

## 1.2 Различные парадигмы и модели

Для эффективного управления проектом менеджер или группа разработчиков должны выбрать методологию разработки программного обеспечения, которая лучше всего подходит для данного проекта. Все методики имеют разные сильные и слабые стороны и существуют по разным причинам. Ниже описаны наиболее часто используемых методологий разработки программного обеспечения и их плюсы и минусы.

Команды используют методологию гибкой разработки (Aglie), чтобы минимизировать риски (например, ошибки, перерасход средств и изменение требований) при добавлении новых функций. Во всех гибких методах команды разрабатывают программное обеспечение итерациями, которые содержат мини-приращения новой функциональности. Существует множество различных форм метода гибкой разработки, включая Scrum, Канбан, экстремальное программирование (XP) и разработку, управляемую функциями (FDD).

Основное преимущество гибкой разработки программного обеспечения состоит в том, что он позволяет выпускать программное обеспечение итерациями. Итерационные выпуски повышают эффективность, позволяя командам находить и исправлять дефекты и согласовывать ожидания на ранней стадии. Они также позволяют пользователям получать программное обеспечение с частыми постепенными улучшениями. Из минусов можно отметить то, что методы гибкой разработки полагаются на общение в режиме реального времени, поэтому новым пользователям часто не хватает документации, необходимой для быстрого освоения. Они требуют от пользователей огромных затрат времени, потому что разработчики должны полностью выполнять каждую функцию в каждой итерации для утверждения пользователем.

Методы гибкой разработки похожи на быструю разработку приложений и могут быть неэффективными в крупных организациях. Программисты, менеджеры и организации, привыкшие к каскадному методу, могут столкнуться с трудностями при адаптации к Agile. Поэтому им часто подходит гибридный подход.

DevOps — это не только методология разработки, но и набор практик, поддерживающих организационную культуру. Развертывание DevOps сосредоточено на организационных изменениях, которые расширяют сотрудничество между отделами, отвечающими за различные сегменты жизненного цикла разработки, такие как разработка, обеспечение качества и операции. Плюсы: DevOps ориентирован на сокращение времени выхода на рынок, снижение количества отказов новых выпусков, сокращение времени между исправлениями и минимизацию сбоев при максимальной надежности. Для этого организации с DevOps стремятся автоматизировать непрерывное развертывание, чтобы все происходило гладко и надежно. Компании, использующие методы DevOps, получают выгоду за счет значительного сокращения времени выхода на рынок и повышения удовлетворенности клиентов, качества продукции, а также производительности и эффективности сотрудников. Минусы: некоторым клиентам не нужны постоянные обновления своих систем; в некоторых отраслях есть правила, которые требуют всестороннего тестирования, прежде чем проект может перейти в операционную фазу; если разные отделы используют разные среды, необнаруженные проблемы могут перейти в производственную среду; некоторые атрибуты качества требуют вмешательства человека, что замедляет конвейер доставки.

Многие считают метод водопада самым традиционным методом разработки программного обеспечения. Каскадный метод — это жесткая линейная модель, состоящая из последовательных этапов (требования, проектирование, реализация, проверка, обслуживание), ориентированных на определенные цели. Каждый этап должен быть завершен на 100%, прежде чем можно будет начать следующий этап. Обычно нет процесса возврата, чтобы изменить проект или направление. Плюсы: линейный характер метода разработки водопада упрощает понимание и управление; в проектах с четкими целями и стабильными требованиями лучше всего использовать метод водопада; менее опытные руководители проектов и проектные группы, а также команды, состав которых часто меняется, могут получить наибольшую выгоду от использования методологии разработки водопада. Минусы: метод создания водопада часто медленный и дорогостоящий из-за его жесткой структуры и жесткого контроля; эти недостатки могут побудить пользователей метода водопада к изучению других методологий разработки программного обеспечения.

Быстрая разработка приложений (RAD) — это сжатый процесс разработки, в результате которого создается высококачественная система с низкими инвестиционными затратами. Скотт Стинер, генеральный директор и президент UM Technologies, сказал в Forbes: «Процесс RAD позволяет нашим разработчикам быстро приспосабливаться к меняющимся требованиям на быстро меняющемся и постоянно меняющемся рынке». Возможность быстрой настройки — это то, что обеспечивает такие низкие инвестиционные затраты. Метод быстрой разработки приложений состоит из четырех этапов: планирование требований, пользовательское проектирование, создание и переключение. Этапы пользовательского проектирования и строительства повторяются до тех пор, пока пользователь не подтвердит, что продукт соответствует всем требованиям. Плюсы: быстрая разработка приложений наиболее эффективна для проектов с четко определенной бизнес-целью и четко определенной группой пользователей, но которые не требуют вычислительной сложности. RAD особенно полезен для малых и средних проектов, которые зависят от времени. Минусы: для быстрой разработки приложений требуется стабильный состав команды с высококвалифицированными разработчиками и пользователями, хорошо разбирающимися в области приложения. Глубокие знания необходимы для сжатого графика разработки, который требует утверждения после каждого этапа строительства. Организации, не отвечающие этим требованиям, вряд ли получат выгоду от RAD.

## 1.3 Инструменты и фреймворки гибкой методологии разработки

Начнем со Scrum, так как это самый распространенный и самый плохо реализованный из фреймворков Agile. Официально из руководства по Scrum: Scrum состоит из 3 ролей (Скрам-мастер, владелец продукта и команда разработчиков), 5 событий (Sprint, Sprint Planning, Daily Scrum, Sprint Review и Sprint Retrospective) и 3 артефактов (бэклог проекта, журнал пожеланий спринта и инкремент продукта). Scrum идеально подходит для проектов по разработке программного обеспечения, требования к которым меняются. А также для: сложной работы, требующей дисциплины и навыков; отрасли, сталкивающейся с меняющимися рынками и сложной конкуренцией при разработке продуктов или программного обеспечения; организаций, которым для достижения успеха необходимо произвести более интенсивные или серьёзные изменения в своей культуре или структуре. Если к проекту относится один или несколько элементов, то Scrum будет хорошим кандидатом в качестве Agile-фреймворка для внедрения. В обратном случае, стоит рассмотреть что-то другое.

Теперь рассмотрим Канбан. Это самый легкий из фреймворков Agile, он довольно популярен, это самый быстрорастущий метод, тесно связанный с Lean. От создателя Дэвида Дж. Андерсона: Канбан состоит из трех основополагающих принципов: (1) начните с того, что вы делаете сейчас, (2) согласитесь проводить постепенные, эволюционные изменения и (3) уважайте текущий процесс, роли, обязанности и звания. Затем опирайтесь на это понимание с помощью пяти основных свойств: визуализируйте рабочий процесс; ограничить незавершенное производство (незавершенное производство); управление потоком; сделайте политики процессов явными; совершенствуйте вместе (используя модели и научный метод). Канбан идеально подходит для сложных систем или процессов, требующие длительные сроки выполнения заказа (недели и месяцы); для организаций, которые хотят небольших, регулярных и постепенных изменений в своей структуре или процессах; для аппаратного обеспечения, сложного продукта или программного обеспечения, которое необходимо улучшать в течение более длительных периодов времени.

Обратимся к экстремальному программированию, так как это один из самых популярных фреймворков Agile, который более интенсивен, чем Scrum и Kanban. Экстремальное программирование (также известное как XP) имеет пять ценностей: (1) простота, (2) общение, (3) обратная связь, (4) уважение и (5) смелость.

XP также имеет множество правил в пяти областях, в том числе: планирование (пользовательские истории, планирование выпуска и т. д.), управление (открытое рабочее пространство, устойчивый темп и т. д.), дизайн (простота, карточки CRC и т. д.), кодирование (стандарты, парное программирование и др.), тестирование (модульное тестирование, приемочные тесты и т. д.). Экстремальное программирование идеально подходит для любого проекта разработки программного обеспечения. XP требует другого уровня поддержки как со стороны членов команды, так и со стороны среднего и высшего руководства. Для получения желаемых качественных и количественных преимуществ требуется гораздо больше дисциплины и инвестиций.

# ВЫВОД ИЗ I ГЛАВЫ

Есть множество разных моделей и фреймворков для управления разработкой программного обеспечения. У каждого из них есть как плюсы, так и минусы. Для разработки программного проекта сервиса для управления задачами, который изначально будет разработан как минимально жизнеспособный продукт (MVP) и будет часто обновляться, подойдёт методология гибкой разработки и фреймворк канбан.

# 2 ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ МЕТОДОЛОГИИ

## 2.1 Техническое задание и диаграмма Ганта

## 2.2 Инструменты управления проектом

Код приложения было решено хранить в сервисе GitHub, и Git в качестве системы управления версиями, так как это самая популярная и удобная система для хранения кода.

Также я выбрал GitHub Projects для управления проектом, так как он бесплатный и удобно, что код и система управления проектом находятся в одном месте.

# ВЫВОД ИЗ II ГЛАВЫ

# ЗАКЛЮЧЕНИЕ

# ЛИТЕРАТУРА

1. Agile. Оценка и планирование проектов / Кон М.; Пер. с англ. — М.: Альпина Паблишер, 2018.
2. Agile-менеджмент. Лидерство и управление командами / Аппело Ю., Пер. с англ. — М.: Альпина Паблишер, 2019.
3. Проектирование информационных систем / Чистов Д. В., Мельников П. П., Золотарюк А. В., Ничепорук Н. Б.; под ред. Чистов Д. В. — М.: Юрайт, 2020.
4. Top 4 software development methodologies // Synopsys URL: https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/ (дата обращения: 17.12.2020).
5. Choosing the Right Agile Framework for You // LinkedIn URL: https://www.linkedin.com/pulse/choosing-right-agile-framework-you-paul-j-heidema (дата обращения: 17.12.2020).
6. Сравнение Kanban и Scrum // Atlassian URL: https://www.atlassian.com/ru/agile/kanban/kanban-vs-scrum (дата обращения: 17.12.2020).