МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Курсовая работа**

«Управление разработкой приложения учета инвестиций»

Обучающегося 3 курса

Гришутенко Павла Петровича

Научный руководитель:

Кандидат физико-математических наук, ассистент

Жуков Николай Николаевич

Санкт-Петербург

2020

ОГЛАВЛЕНИЕ

Стр.

[ВВЕДЕНИЕ 3](#_heading=h.gjdgxs)

[1. ТЕОРИЯ РАЗРАБОТКИ ПО 4](#_heading=h.30j0zll)

[1.1 Программный проект 4](#_heading=h.1fob9te)

[1.2 Жизненный цикл продукта 4](#_heading=h.3znysh7)

[1.3 Модели разработки программного продукта 5](#_heading=h.2et92p0)

[1.4 Управление содержанием проекта 6](#_heading=h.tyjcwt)

[1.5 Установление сроков реализации проекта 6](#_heading=h.3dy6vkm)

[1.6 Распределение человеческих ресурсов 7](#_heading=h.1t3h5sf)

[2 РАБОТА НАД ПРОЕКТОМ. 9](#_heading=h.4d34og8)

[2.1 Описание предметной области 9](#_heading=h.2s8eyo1)

[2.2 Задача продукта, техническое задание 9](#_heading=h.17dp8vu)

[2.3 Оценка времени работы над проектом 10](#_heading=h.3rdcrjn)

[2.4 Подготовка шаблона 11](#_heading=h.26in1rg)

[2.5 Создание UML-диаграммы 12](#_heading=h.lnxbz9)

[2.6 Инструменты для отслеживания прогресса разработки 13](#_heading=h.35nkun2)

[2.7 Этап тестирования, внедрения и поддержки 13](#_heading=h.1ksv4uv)

[2.8 Вывод 14](#_heading=h.2jxsxqh)

[ЗАКЛЮЧЕНИЕ 15](#_heading=h.z337ya)

[ЛИТЕРАТУРА 16](#_heading=h.3j2qqm3)

# ВВЕДЕНИЕ

Разработка программного обеспечения трудоемкий процесс, помимо создания самого приложения разработчикам приходится решать ряд задач, которые на первый взгляд никак не связаны с написанием кода. Все эти задачи имеют управленческий характер и отвечаю за способы налаживания работы над проектом.

Существует дисциплины, в рамках которой рассматриваются способы работы над проектами, включая проектирование продукта и способа его производства, управление техническими и человеческими ресурсами.

В курсовой работе будут рассмотрены теоретические знания необходимые для управления программными проектами на примере приложения по учету инвестиций.

Предмет исследования – процесс управления разработкой приложения.

Основная цель работы заключается в рассмотрении способов управления программными продуктами при создании приложения управления и учета инвестиций.

Основными задачами курсовой работы являются:

1. Рассмотрение и изучение всех, необходимых в работе над проектом, теоретических материалов по управлению программными продуктами.
2. Применение теории управления программными продуктами в разработке приложения.
3. Анализ применяемых в разработке средств управления проектом.

Поставленные задачи можно решить с использованием различных методов исследования, основными из которых являются следующие: анализ структуры курсовой работы, сбор и изучение информации об управлении программными проектами, определение способа создания приложения с использованием средств информационных технологий и выполнение работы с использованием индуктивного, дедуктивного метода для понятного изложения материала.

Практическая ценность проведенного исследования состоит в том, что её результат может быть использован в качестве вспомогательного средства при разработке собственных программных решений.

# ТЕОРИЯ РАЗРАБОТКИ ПО

В этой главе изложена теория управления разработкой программного продукта.

## Программный проект

Широкое распространение получили методы и принципы управления проектами в деятельности человека. Потому важно понимать, как их можно применить в своей работе. Прежде чем говорить об управлении программным проектом, необходимо разобраться что он из себя представляет: дать определение проекту.

Проект – комплекс работ, приводящих к определенному результату, которые оформляются в виде плана действий и продукта, который будет получен в результате выполнения плана [4].

Современные программные продукты сложны и вызывают множество проблем во время их разработки. Чтобы решить эти проблемы, были созданы различные правила и техники. Со временем они эволюционировали и уже сейчас применимы на различных фазах разработки программных продуктов [5].

Важно использовать подходящие техники в управлении программными проектами, так как они способны дать следующие преимущества: экономия, снижение конкурентного и технологического риска, планирование будущего дохода, ускоренный темп вывода новых продуктов в эксплуатацию [11].

## Жизненный цикл продукта

Говоря об управлении разработкой программного проекта, в первую очередь следует подумать о назначении продукта и его жизненном цикле.

Жизненный цикл – последовательность видов деятельности работников, охватывающий все этапы эксплуатации продукта до полного прегрешения его использования. Любой жизненный цикл продукта можно разделить на этапы планирования, разработки, тестирования, внедрения и поддержки [4].

Этап планирования является формальным началом разработки проекта, на этом этапе реализуются все технические вопросы управления, проводится анализ проекта, подготавливается рабочее пространство и определяется задачи и объем работ, результатом планирования для разработчика, как правило, является техническое задание и календарный план [4].

Этап разработки начинается после того, как все документы и инструменты готовы для старта проекта. Основные задачи этого этапа убедится, что составленные планы работ исполнимы, после чего приступать к реализации проекта в соответствии с календарными планами [10].

Этап тестирования не всегда идет после этапа разработки, так как может быть запущен параллельно ему [4].

Этап внедрения включает в себя выпуск продукта, его установку, верификацию и валидацию [4].

Этап поддержки заключается в обеспечении эффективной по затратам поддержки функционирования и модификации программного продукта [4].

На протяжении всего жизненного цикла проекта необходимо адекватно оценивать риски. Оценка рисков является отдельной проблемой управления, и эта оценка, как правило, получается после сопоставления календарных планов, ресурсов и сложности проекта со всеми ограничениями и допущениями. Календарный план строится на основе трех целей: определить риски, найти место на проекте каждому работнику и предоставить каждому участнику процесса средства для контроля над ходом работы [2, 11].

## Модели разработки программного продукта

Для реализации программных проектов, были предложены различные модели разработки.

Модель разработки ПО – структура, связывающая и систематизирующая различные виды деятельности, их последовательность, а также взаимодействие в процессе разработки программного обеспечения [6].

Выбор той или иной модели серьёзно влияет на все процессы в проекте и зависит от масштабов задачи и доступных ресурсов. К известными моделям разработки можно отнести: водопадная модель, интеграционная инкрементальная и спиральная модели, V-образная модель, гибкая модель.

Водопадная модель предполагает однократное последовательное выполнение каждой задачи проекта, при этом каждый участник может видеть текущею, предыдущею и следующие задачи без возможности охватить проект целиком. На сегодняшний день модель считается устаревшей и редко где используется [6].

V-образная модель стала идейным продолжением водопадной модели, отличие заключается в разделении этапов разработки и внедрения, а также добавлением этапов тестирования как связующего звена между всеми этапами.

Интеграционная инкрементальная модель фундамент современных подходов к разработке ПО. Представляет собой круговою схему, на вход которой подается задача, а на выходе получается готовый результат. Спиральная модель является частым случаем интеграционной инкрементальной модели, в основе которой лежит управление рисками со выделение контрольных точек проекта.

Гибкая модель разработки ПО, известная как Agile, представляет собой целый комплекс моделей разработки ПО. Agile выделяют как отдельную модель, потому что она обладает определенной идеологией: взаимодействие людей важнее процессов и инструментов, рабочий продукт важнее документации, сотрудничество с заказчиком важнее условий контракта, проект может быть изменен, даже если в начальный план изменения не входили [6].

Главный недостаток модели Agile сложность в реализации этого подхода, так как многие идеи и методики применяются не всегда в нужный момент.

## Управление содержанием проекта

После завершение разработки проекта, наступает фаза внедрения и поддержки продукта. Очень часто после этого происходит ситуация, когда требуется внести изменения в проекте, например, добавить или изменить ряд функциональных возможностей. Подобная ситуация может наступить в связи с успехом продукта, по просьбе заказчика или в иных случаях, требующих адаптации программного решения к текущей ситуации. При использовании методологии Agile, как правило проблем на стадии управления не возникает, но, если используется другая модель разработки ПО, процесс работы над проектом может быть начат сначала в качестве нового проекта.

Чтобы вести учет будущих изменений рабочего проекта, как один из вариантов, применяется матрица координации изменения, которая интегрируется в базовый план содержания проекта. Матрица координации изменения представляет из себя дорожную карту разработки продукта на временном отрезке с возможностью быстрой адаптации при необходимости. В случае использования такой матрицы необходимо следить за потоком изменений, чтобы проект сохранил свою целостность [10].

## Установление сроков реализации проекта

Чтобы грамотно спланировать сроки реализации проекта строится схема работ в соответствии с выбранной моделью разработки, после чего назначается самый ранний временной промежуток выполнения каждой задачи, после чего выявляются все ограничения, налагаемые запланированными датами, всё время, выведенное после сопоставлений ограничений с указанными сроками, отправляется в резерв для задачи, анализ которой был проведен. Если в процессе выполнения задачи остается время в резерве, то это время становится общим резервом и может быть задействована в реализации других задач. Менеджер на основе данных о времени, затраченном на задачи, проводит анализ для лучшего планирования проекта в будущем [8].

Одним из вариантов представления задач относительно времени является линейная диаграмма Ганта, где слева отображены именования работ, а в право идет временная шкала, в таком случае время, затрачиваемое на задачу, отражается в виде горизонтальной линии, которая проецируется на временную шкалу [4].

## Распределение человеческих ресурсов

Правильное формирование команды сокращает время выполнения проекта и увеличивает его качество. Процесс создания команды непрерывен и протекает в течении всего жизненного цикла проекта. В процессе взаимодействия участники команды проходят стадии формирования, притирки, нормализации и работы. Особенность работы команды на каждой стадии учитываются управлением на процессе планирования. Задача, предъявляемая новой команде, добиться сплоченной работы за наименьшее время [10].

Команда ведет процесс разработки распределяя задачи на проекте между собой по наличию у человека определенных навыков. Есть несколько способов ведения работы в команде, один из таких методов является Scrum-метод управления проектами.

Команда, применяющая Scrum, самостоятельно решает, как выполнять задачи проекта, при этом происходят регулярные собрания для решения различных вопросов, возникающих в процессе работы. Этот подход хорош тем, что не требует координации каждой задачи на проекте от начальства [12].

Для отслеживания всех процессов, происходящих в проекте, была разработана табличная форма, получившая название Scrum-доска. Столбцы этой таблицы отражают основные этапы работы над задачей, например, разработка, тестирование, завершение. В качестве элементов таблицы используют конкретные задачи, которые перемещают по столбцам, чтобы каждый участник команды мог представлять текущее состояние проекта. Использование Scrum-доски помогает не только следить за проектом, но и дает возможность участникам команды выбирать оптимальные для себя задачи, чтобы повысить качество исполняемых работ.

Scrum-доска не единственная форма визуализации процессов. Существует также метод канбан, который базируется на визуализации не материальной деятельности, в канбан таковыми выступают сервисы. Представление визуальных сигналов механизмов происходит на канбан-досках. Канбан-доска содержит определенный лимит задач, и новые задачи поступают только после завершения всех задач на доске [1].

1.7 Вывод

В результате разобраны основные термины и этапы работы над проектом. Изученная теоретическая часть поможет в разработке собственного программного продукта.

# РАБОТА НАД ПРОЕКТОМ.

В главе будет будут рассмотрены основные этапы работы над программным обеспечением.

## Описание предметной области

В качестве примера работы над проектом, будет использоваться приложение по контролю и учету инвестиций. В качестве глобальной площадки инвестирования выступает фондовый рынок.

Фондовый рынок – платформа для инвесторов, предоставляющая возможность торговать ценными бумагами, покупать ресурсы и инвестировать в различные проекты денежные средства с целью заработка [3].

Целью будущего приложение является сбор информации обо всех инструментах, которые клиент включает в свой инвестиционный портфель, а также проведения анализа и предоставление статистики по этим инструментам. Проект рассчитан на категорию людей, которые предпочитают использовать большое количество платформ для инвестирования. Приложение позволит собрать все необходимые данные со всех площадок и разместить их на едином сервисе.

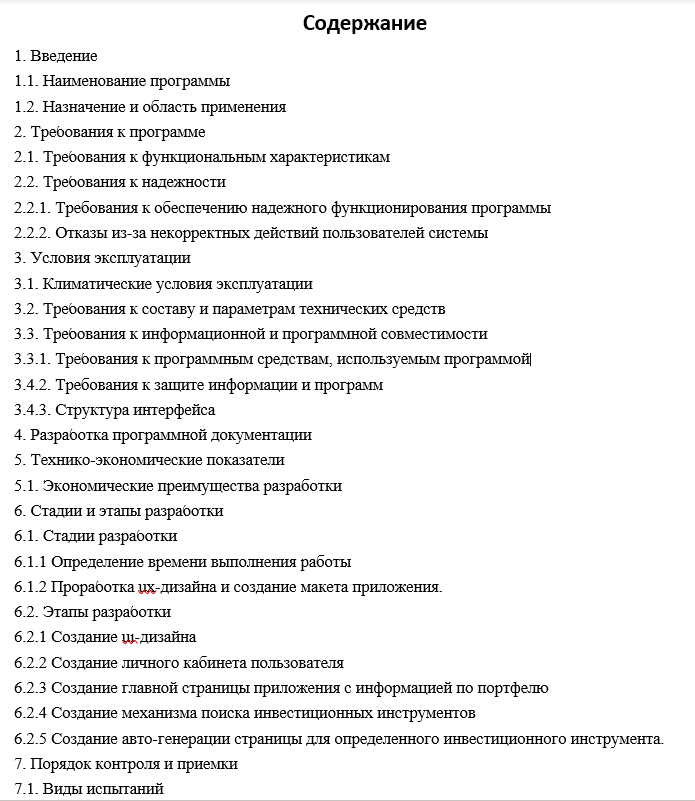
## Задача продукта, техническое задание

Рассказать об этапе составления технического задания и о самом ТЗ проекта.

При создании

При выполнении работы над проектом было составлено техническое задание в соответствии с ГОСТ 15.016-2016 «Система разработки и постановки продукции на производство. Техническое задание. Требования к содержанию и оформлению.». ГОСТ предусматривает следующие основные пункты для технического задания выбранного приложения: наименование, цель, технические и экономические требования, требование к безопасности, конструктивные требования [15].

Фрагмент, в котором отражено содержание технического задания, представлен на «рисунке 1».

Рисунок 1

Содержание технического задания предусматривает следующие этапы работы над проектом: распределение времени для работы над поставленными задачами, разработка макета будущего приложения в соответствии с требованиями к функциональным характеристикам, написание программного кода, отвечающего стандартам надежности и безопасности, тестирование продукта перед публикацией.

## Оценка времени работы над проектом

На основании составленного технического задания, была проведена оценка времени работы над проектом, представленной в виде упрощённой модели программного продукта. Упрощённая модель строится с помощью конкретных задач из технического задания, помещенных на календарную матрицу. Календарная матрица была визуализирована в виде диаграммы Ганта. Представленный фрагмент диаграммы (рис. 2), отражает временные затраты на задачи в течение недели.

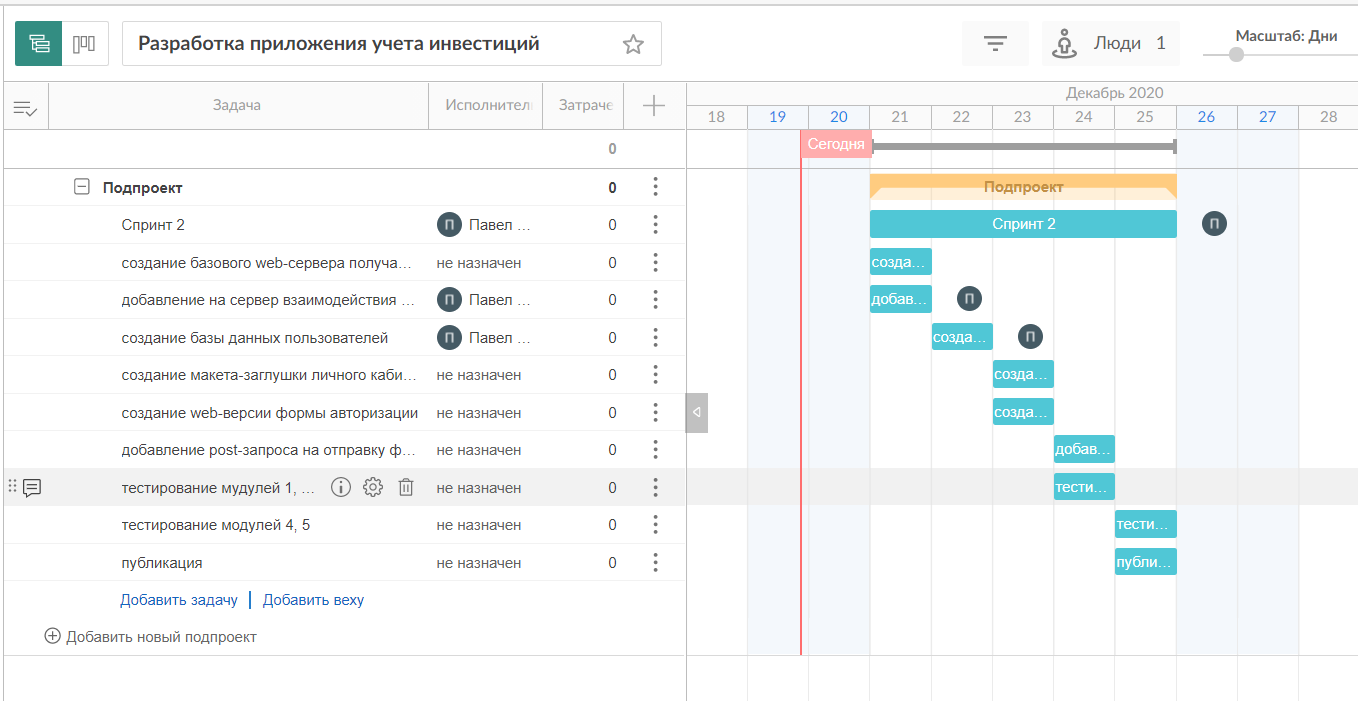


Рисунок 2

Диаграмма была получена с помощью сервиса GanttPRO. Этот сервис был выбран потому, что

## Подготовка шаблона

Чтобы понять, как конечные элементы системы будут представлены пользователю необходимо создать макет (рис. 3), на котором будут расположены необходимые элементы пользовательского интерфейса. Изображение выглядит как стол

Автоматически созданное описание

Рисунок 3

Главное меню следует расположить слева вверху, чтобы воспользоваться порядком восприятия элементов. При создании макета графического интерфейса были использованы следующие приемы: визуальная группировка элементов для создания четкой иерархии, это отражено разными размерами блоков информации; проложен графический маршрут между целостными структурами, это можно увидеть благодаря визуальному разделению и выделению разделов; использован целостный образ, все элементы соответствуют одной тематике [7].

## Создание UML-диаграммы

Чтобы понять удачно ли составлено техническое задание, описаны ли верно все необходимые компоненты, реализованы ли между всеми компонентами связи, была использована UML диаграмма.

UML диаграмма – это инструмент позволяющий визуализировать модель взаимодействия отдельных компонентов системы, а также помогает описать и спроектировать всю модель целиком в объектной форме [13].

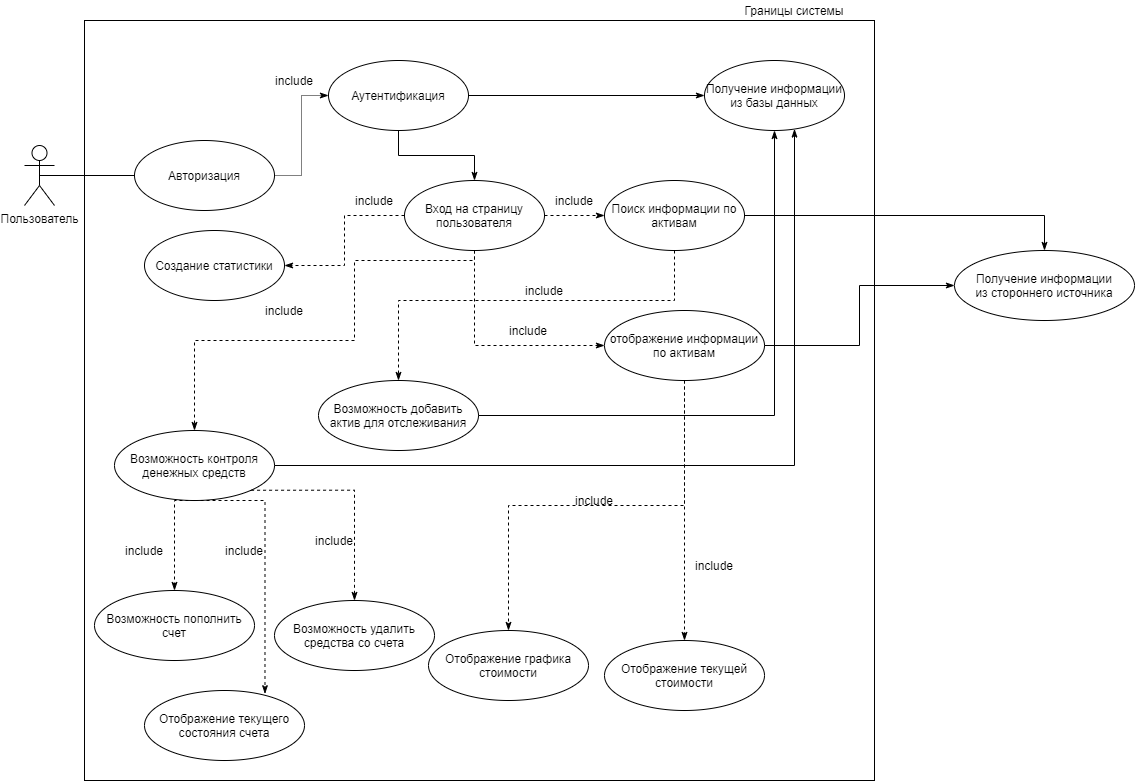
Для проекта была составлена диаграмма прецедентов (рис. 4), которая помогает оценить качество функциональных узлов в системе.

Рисунок 4

На диаграмме представлены экземпляры объектов и их взаимодействия между собой в рамках одного прецедента. В терминах этой диаграммы актер является лицом, осуществляющим взаимодействие с системой. Все действия, происходящие внутри системы, называются сценариями. Сценарии описываются прецедентами, которые взаимодействуют между собой [13].

## Инструменты для отслеживания прогресса разработки

В процессе разработки возникает необходимость отслеживать выполнения задач. Для этого во многих компаниях используют разные модели управления проектами. При работе над приложением был задействован Scrum метод. Работа была разделена на недельные спринты, а с помощью Scrum-доски была возможность отслеживать выполняемую работу. Scrum-доска была реализована с помощью сервиса Trello (рис. 5). Заключительная особенность этого сервиса состоит в том, что он простой, бесплатный и содержит все необходимые для выполнения работы функции.

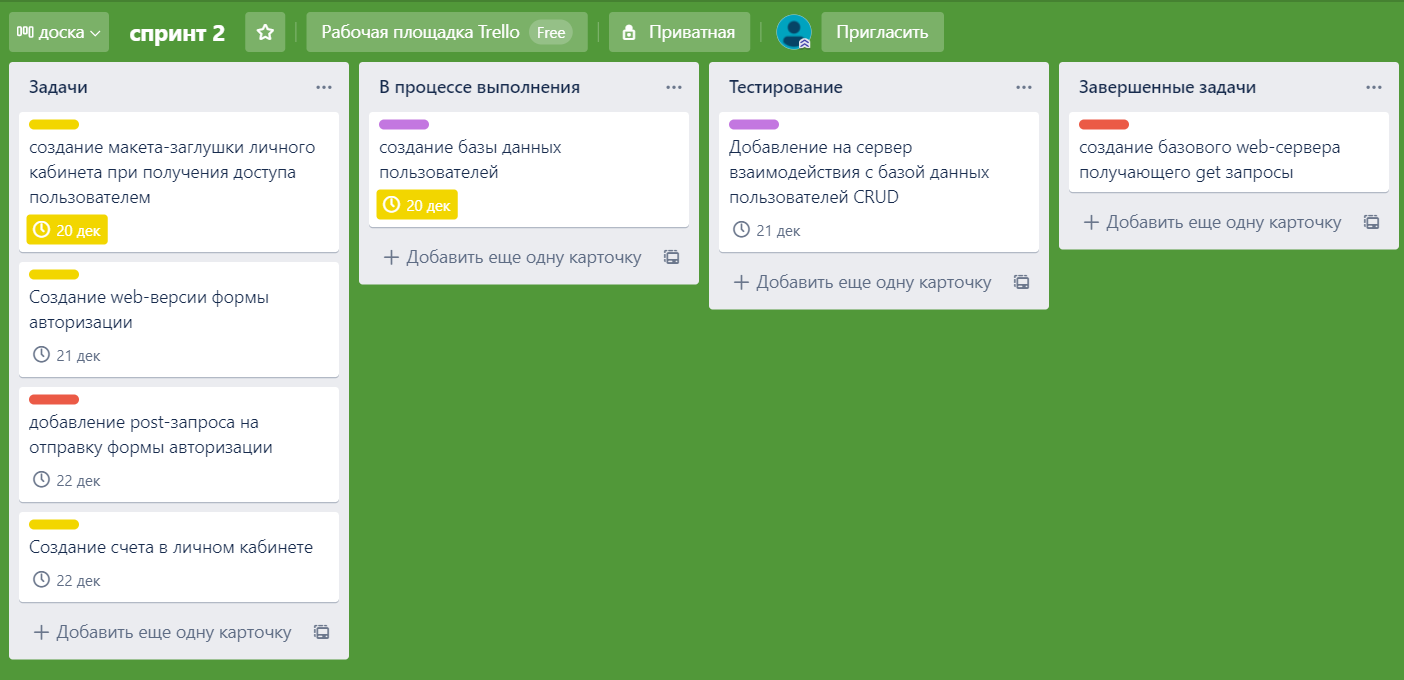


Рисунок 5

Scrum-доска представленная в виде 4 столбцов: задачи, процесс исполнения, процесс тестирования. завершенные задачи.

Для отслеживания истории публикаций была использована система контроля версий Git. Она была выбрана, потому что обладает простой архитектурой, скоростью, хорошей поддержкой не линейной разработки и возможностью эффективной работы над большим проектом. Система контроля версий позволяет регистрировать историю разработки проекта, а также возвращать предыдущие версии продукта в случае возникновения непредвиденных ситуации, например ошибок. Системы контроля версий являются стандартом индустрии и используются везде, где необходимо следить за состоянием разрабатываемого продукта [14].

В проектной работе был задействован локальный способ организации работы с системой Git.

## Этап тестирования, внедрения и поддержки

Тесты являются изолированным компонентом системы, так как не нужны для её функционирования. Задача тестирования поддерживать разработку продукта. Фактически тесты представляют модель поведения, которому должна следовать программа [9].

Тестирование продукта предусмотрено в техническом задании. Было проделано ui-тестирование на основе макета следующим образом: были составлены инструкции, в которых описано поведение пользователя, после чего макет приложения был показан группе лиц, которые давали оценку на основе опыта взаимодействия с макетом.

Проект предусматривает проведение тестирования приложения на локальной машине заказчика с использованием согласованных методик. Ход проведения испытаний должен быть зарегистрирован в протоколе.

Этап внедрения продукта предполагает размещения сервиса в интернете для свободного использования другими пользователями. Такое требование было представлено в техническом задании проекта. Для реализации этой задачи необходимо использовать серверное оборудование с доступом в сеть.

Этап поддержки осуществляется в соответчики с договором оформленном с заказчиком. Поддержка может включать в себя решение технических неисправностей и внедрении новых функции на основе нового технического задания. При этом возобновляется цикл разработки.

## 2.8 Вывод

На основе полученных знаний об управлении разработкой программного проекта был создан программный продукт. Его создание включало в себя следующие этапы: проектирование, создание технического задания, разработка и тестирование макета, визуализация взаимодействия компонентов, декомпозицию задач и распределение по времени, разработка и завершение.

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы рассмотрены способы управления программным продуктом на примере создания приложения учета инвестиций.

В процессе были решены следующие задачи:

1. Изучен весь необходимый материал необходимый для понимания процесса создания и управления программным продуктом.
2. Полученные знания были применены для тестового проекта по разработке приложения
3. Был проведен анализ инструментов, применяемых для управления проектами.

В результате курсовой работы были исследованы темы связанный с управлением программными продуктами и составлен план управления разработкой, на примере приложения учета инвестиций.

В процессе выполнения работы были использованы инструменты позволяющие отследить и проконтролировать время и действия, которые использует разработчик для выполнения плана по созданию программного продукта.

# ЛИТЕРАТУРА

1. Андерсон Д. Канбан: альтернативный путь в Agile: учебное пособие. —2016—№ 1.— С. 9.

2. Беркун С. Искусство управления IT-проектами: учебное пособие. —2014—№ 1.— С. 37-39.

3 Вилли Д., Сонс Д. Фондовый рынок курс для начинающих: учебное издание. —2012—№ 1.— С. 14-17.

4. Ехлаков Ю.П. Управление программного проекта: учебное пособие. —2014 —№ 1.— С. 6, 10-12, 17-20, 26-41, 127-128.

5. Иванова Г.С. Технология программирования: учебное пособие. —2016—№ 3.— С 20-22.

6. Куликов С. Тестирование программного обеспечения. Базовый курс: учебное пособие. —2020—№ 2.— С. 18-25.

7. Купер А. Об интерфейсе. Основы проектирования взаимодействия. —2009—№ 2.— С. 339-340.

8. Локк Д. Основы управления проектами: учебное пособие. —2004—№ 1.— С. 117-128.

9. Мартин Р. Чистая архитектура: монография. —2018 —№ 1.— С. 242-246.

10. Милошевич Д. Набор инструментов для управления проектами: монография. —2008—№ 1.— С. 252-265, 405-411, 455-460.

11. Рассел Д. Арчибальд. Управление высокотехнологичными программами и проектами. —2010.—№ 3.— С. 35-37, 47.

12. Сазерленд Д. Scrum. Революционный метод управления проектами: монография. —2016—№ 1.— С. 46-55.

13. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования: учебное пособие. —2005—№ 3.— С. 26-33, 127-130.

14. Шакон С. Pro Git: учебное пособие —2012. —№ 1.— С. 18-21.

15. ГОСТ 15.016-2016 «Система разработки и постановки продукции на производство. Техническое задание. Требования к содержанию и оформлению.» —Пкт. 6.1.