



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

«Управление программным проектом по проектированию деятельности
ресторана с использованием "1С:Предприятие"»

Обучающейся 3 курса
Леонтьевой Анны Викторовны

Научный руководитель:
Кандидат физико-математических наук,
доцент кафедры ИТиТО
Жуков Николай Николаевич

Санкт-Петербург
2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
Описание проекта и выбор методологии управления проектом	4
1.1. Предметная область	4
1.2. Модели управления разработкой	4
Управление проектом и выполненные задачи	7
2.1. Внедрение основных принципов методики Scrum в процесс реализации программного продукта	7
2.2. Документальное сопровождение программного проекта	11
2.3. Анализ процесса реализации программного проекта по диаграмме сгорания задач	14
ЗАКЛЮЧЕНИЕ	16
ЛИТЕРАТУРА	17
ПРИЛОЖЕНИЕ	18

ВВЕДЕНИЕ

Деятельность ресторана в наши дни предполагает большие объемы отчетности по всем видам работ, происходящих в точках общественного питания. В связи с этим, целесообразна разработка информационной системы, которая позволит оптимизировать и упростить работу ресторана.

Реализация подобных информационных систем происходит в команде разработчиков, исходя из этого, актуальным вопросом является организация управления программным проектом. Для качественной реализации проекта необходим рациональный и корректный выбор методологии управления. Один из вариантов определения таковой – изучение существующих моделей и сопоставление ресурсов команды на начало проекта с соответствующими требованиями каждой модели.

Целью курсовой работы является реализация программного проекта по проектированию деятельности ресторана с применением методики управления Scrum.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

- Исследовать методологии и методики управления;
- Определить рациональную для использования методику управления программным проектом;
- Изучить выбранную методику управления и внедрить основные принципы в работу команду разработчиков;
- Разработать программный проект по проектированию деятельности ресторана;
- Подготовить сопроводительную документацию к программному продукту.

Объектом является процесс разработки программного проекта по проектированию деятельности ресторана.

Предметом является реализация методики управления программным проектом.

Описание проекта и выбор методологии управления проектом

1.1. Предметная область

Недавно открывшийся московский ресторан “Claude Monet” запросил разработку конфигурации для автоматизации деятельности заведения. Ресторан закупает продукты у поставщиков, осуществляет хранение и изготовление продуктов питания на собственном производстве. У ресторана есть зал, в котором осуществляется обслуживание гостей, и есть возможность бронирования. Заведение реализует систему лояльности в виде дисконтных карт и прочих акций.

1.2. Модели управления разработкой

Различаются следующие модели методологий разработки программного обеспечения:

- Каскадная модель
- RAD-модель (быстрая разработка)
- Итерационная модель
- Спиральная модель
- Гибкая модель и др.

Для определения подходящей модели разработки необходимо изучить основные принципы вышеописанных моделей.(1)

Каскадные методологии разработки программного обеспечения подразумевают, что разработка программного обеспечения делится на фазы, каждая из которых характеризуется своим набором работ. Сначала происходит выявление всех требований к проекту и их анализ. Затем проектная группа приступает к проектированию системы, разбивая создаваемую систему на подсистемы и далее детализируя их до уровня программных процедур и функций. После этого начинаются разработка кода и модульное тестирование. Затем наступает очередь сборки и системного тестирования. (2) И так далее — вплоть до передачи системы заказчику. Преимуществами каскадного подхода считаются: минимальный объем переработок написанного кода, возможность тщательно спроектировать архитектуру системы и однократное тестирование системы. Использование каскадной модели оправдано в том случае, если соблюдаются следующие пункты: требования

известны, понятны и зафиксированы; нет проблем с доступностью программистов нужной квалификации; работа в относительно небольших проектах.

RAD методология, представляет собой «быструю» технологию создания программного обеспечения информационных систем (Rapid Application Development). Она ориентирована на спиральную модель жизненного цикла. Для технологии характерны следующие основные особенности: разработка выполняется небольшим коллективом программистов от двух до десяти человек; график работ предполагает их выполнение за короткий период времени, при этом известен алгоритм каждой работы; разработка программного обеспечения выполняется итерациями, разработчики предоставляют заказчику прототип системы, в который затем вносят изменения в соответствии с пожеланиями заказчика.(4)

Итерационная модель подразумевает выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл: Планирование — Реализация — Проверка — Оценка (англ. plan-do-check-act cycle). Итеративный подход разбивает разработку на несколько итераций, в ходе каждой из которых выполняются практически все типы работ, и создается реальная работающая система с все более развитыми функциональными возможностями. Из недостатков модели можно выделить следующие пункты: каждая фаза самостоятельна, отдельные итерации могут не накладываться; могут возникнуть проблемы с реализацией общей архитектуры системы, поскольку не все требования известны к началу проектирования; нет фиксированного бюджета и сроков, а также нужна сильная вовлеченность Заказчика в процесс.

В спиральной модели делается упор на начальные этапы жизненного цикла программного проекта: анализ требований, проектирование спецификаций, предварительное и детальное проектирование. На этих этапах проверяется и обосновывается реализуемость технических решений путем создания прототипов. Каждый виток спирали соответствует в каскадной модели созданию фрагмента или версии системы; на нем уточняются цели и характеристики проекта, определяется его качество, планируются работы следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта и в результате выбирается обоснованный вариант, который доводится до реализации. Основная проблема спирального цикла — определение момента перехода на следующий этап. Для ее решения необходимо ввести временные ограничения на каждый из этапов жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся

запланированная работа закончена. План составляется на основе статистических данных, полученных в предыдущих проектах, и личного опыта разработчиков. (3)

Гибкая методология (Agile-метод) нацелена на минимизацию рисков путём сведения разработки к серии коротких циклов, называемых итерациями, которые обычно длятся две-три недели. Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для выдачи мини-прироста по функциональности: планирование, анализ требований, проектирование, программирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки. Agile-методы делают упор на непосредственном общении лицом к лицу. Большинство agile-команд расположены в одном офисе, рабочем пространстве. Команда может также включать тестировщиков, дизайнеров интерфейса, технических писателей и менеджеров.

По итогам изучения основных методологий была выбрана гибкая модель. Выбор в пользу Agile-метода был сделан по следующим причинам:

- Техническое задание озвучивалось в ходе выполнения проекта, то есть запрос пользователя изменялся;
- Квалификация разработчиков проекта определялась как учебная;
- Методология позволяла произвести небольшое планирование для старта разработки проекта.

Scrum — методика организации совместного рабочего процесса, в основе которой — поэтапная разработка и совершенствование продукта небольшой командой специалистов различного профиля. Scrum — это частный пример Agile-подхода. (5)

Из основных инструментов Scrum можно выделить следующие пункты:

- Сбор Scrum-команды и распределение ролей;
- Организация и ведение Scrum-доски;
- Проведение Scrum-митингов для анализа и планирования спринтов.

Методика Scrum была выбрана по нескольким причинам: Scrum-команда позволяет наличие хозяина проекта (Product Owner), который представляет интересы конечных пользователей; механика спринтов совпадала с механикой проведения учебного проекта с еженедельными встречами для презентации и подведения итогов.

Управление проектом и выполненные задачи

2.1. Внедрение основных принципов методика Scrum в процесс реализации программного продукта

Как и любая методика, Scrum предполагает свои инструменты для организации и контроля процесса реализации проекта. В случае проекта данной курсовой проекта были введены следующие инструменты методика Scrum:

- Распределение обязанностей по ролям Scrum;
- Визуализация выполнения задач с помощью Scrum-доски;
- Контроль выполнения работы командой проекта с помощью Scrum митингов.

Работу команды разработчиков было рационально начать с распределения по ролям и определения зоны ответственности. (6)

Scrum предписывает наличие 3-х ролей: Product Owner (отвечает за видение продукта и приоритеты), Команда (отвечает за реализацию продукта) и Scrum Master (устраняет препятствия в работе и руководит Scrum-процессом).

В начале работы над проектом были определены две основные роли: владелец продукта и Scrum-мастер. Преподаватель дисциплины в системе Scrum занимал место Product Owner'а, так как представлял интересы и приоритеты конечных пользователей программного продукта. Им же был определен Scrum-мастер, который в итоге набрал себе команду. По итогам распределения, получилась следующая система ролей в проекте (таблица 1.1)

Роль	Член команды
Scrum-мастер	Иванова Мария
Разработчик (код и структура)	Бережной Михаил, Смирнов Владислав, Столяренко Кирилл
Дизайн	Пузырев-Харьковский Владислав
Тестирование	Шадрин Андрей
Маркетинг и документация	Леонтьева Анна

Таблица 1.1 Распределение ролей в команде

Мной реализовывались и курировались необходимые документы проекта: устав проекта, документация о системе, документация PMBoK¹. Также мной была взята ответственность за представление результатов работы Владелец проекта, создание сопроводительного визуального контента. Разработка системы лояльности в легенде заведения также была моей задачей.

Непосредственным начальником всей команды являлась Мария Иванова - лидер, Scrum-мастер проекта, распределяющий необходимые для выполнения задачи между членами команды и следящий за их выполнением. В задачи Марии входило курирование особенно больших задач, помощь участникам команды с их выполнением по необходимости. Как Scrum-мастер, с ее стороны были организована система работа со Scrum-доской и проведение митингов.

Разработчиками проекта стали Бережной Михаил, Смирнов Владислав и Столяренко Кирилл. Ими была создана структура информационной системы, проведено внедрение необходимого функционала и алгоритмов автоматизации в программе “1С: Предприятие”.

Тестированием проекта занимался Андрей Шадрин. В его задачи входили тестирование актуальных версий проекта и создание сопроводительной документации к тестированию.

Дизайн проекта курировался Владиславом Пузыревым-Харьковским. В его задачи входило создание уникального дизайна проекта и его внедрение в разрабатываемую систему.

Для достижения основной цели – разработки программного продукта – каждую неделю владельцем проекта предлагался список задач на текущий спринт. Для контроля работы команды были использованы подходящие инструменты методики Scrum.

Scrum-доска – это визуальная презентация тех задач, что должны быть решены за один спринт. Scrum-доска разделена на 2 основные части:

- Product backlog (бэклог) - список задач, которые Product Owner хочет видеть в будущих спринтах.

¹ Project Management Body of Knowledge (PMBoK) — свод знаний, максимально полное изложение информации по управлению проектами.

- Sprint backlog – это просто часть общей картины, отображающая то, чем команда занимается в рамках текущего спринта. Разделена на 3 колонки: “надо сделать”, “в процессе”, “сделано”.

Владелец проекта может наблюдать за sprint backlog-ом, но, при этом, не может его менять. А вот product backlog он может менять когда угодно, но изменения вступят в силу только со следующего спринта (т.е. на текущую работу эти изменения не влияют). По окончании спринта команда предоставляет Product Owner-у потенциально готовый к использованию код. (7)

Для реализации Scrum-доски было взято веб-приложение для управления проектами в командах “Asana”. Этот веб-сервис предлагает удобный и интуитивно понятный интерфейс для реализации scrum-доски. Кроме доски для пользователей доступны список, хронология и календарь. Организованная доска представлена на рисунке 2.1.

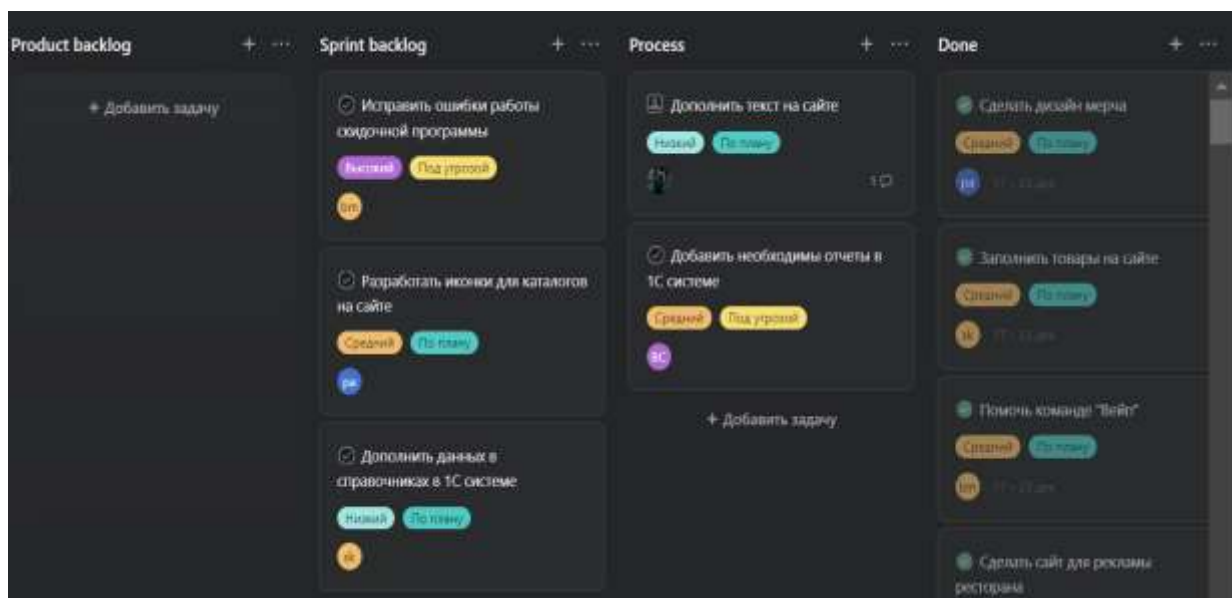


Рисунок 2.1 Scrum-доска команды

Приложение позволяет добавлять подзадачи, расставлять приоритеты, отмечать положение дел по выполнению задач для контроля Scrum-мастером. В задачах отмечался член команды через аккаунт, привязанных к почте (рисунок 2.2), что позволяло отслеживать свои задачи.

Новости:

задачи и проекты, в которых вы участвуете

Перейти в Asana

Управление настройками уведомлений

Ресторан

- ✓ детальный бюджет внедрения проекта ИМ Иванова Мария добавил(а) эту задачу
- ✓ Акции на определенные товары ИМ Иванова Мария добавил(а) эту задачу

Рисунок 2.2 Оповещение о задачах

Говоря о реализации этого подхода в контексте нашего проекта хочется уточнить, что к приложению Asana мы пришли не сразу. В моменте начала проекта и изучении всех тонкостей методики Scrum задачи распределялись Scrum-мастером после получения технического задания от владельца проекта и публиковались в рабочем чате. Контроль выполнения осуществлялся в личном порядке. После полного изучения методики, было принято решение перенести систему распределения и отслеживания сначала в Notion, а потом в Asana. Визуализация хода выполнения задач облегчила процесс контроля задач со стороны Scrum-мастера и дала возможность членам команды взаимодействовать в более структурированном формате.

Работа в методике Scrum предполагает актуализацию и планирование задач. Происходит это на различных встречах, которые проводятся силами Scrum-мастера.

В методике Scrum выделяется три вида встреч: Sprint Planning Meeting (планирование в начале спринта), Daily Scrum Meeting (ежедневный контроль выполнения задач) и Sprint Retrospective Meeting (анализ выполненной работы).

Во время Sprint Planning Meeting происходит определение цели спринта и формирование Sprint Backlog. (8)

На ежедневных встречах команды, именуемые Daily Scrum Meeting, члены команды отвечают на следующие вопросы: «что я сделал вчера?», «что я планирую сделать сегодня?», «какие препятствия на своей работе я встретил?». В ходе этой встречи определяются проблемы и препятствия, пути их решения, уточнение задач текущего спринта.

В конце спринта происходит Sprint Retrospective Meeting, на котором происходит анализ проделанной работы, мотивация членов scrum-команды, обсуждение проблем.

Спринты проекта длились неделю: от занятия по дисциплине до следующего занятия. Поэтому встречи Sprint Planning Meeting и Sprint Retrospective Meeting происходили на учебных занятиях. За это время мы успевали представить результаты нашей работы за спринт владельцу проекта (преподавателю), обсудить проблемы и пути их решения и запланировать работу на следующий спринт.

Как пример проблемы, который обсуждался на одной из встреч - проблема позднего релиза для тестирования. Разработчики не всегда успевали передать финальную конфигурацию тестировщику вовремя, чтобы тот успел протестировать версию и написать отчет. По итогам встречи, командой были определены временные рамки, которые должны соблюдаться, чтобы у всех было необходимое время для выполнения задач, особенно зависимых в рамках одного спринта. (9)

Ежедневную обратную связь мы осуществляли в рабочем чате. Ее организовывал Scrum-мастер. Уточнение положения дел у каждого участника команды помогало вовремя делегировать задачи и осуществлять поддержку.

2.2. Документальное сопровождение программного проекта

Сопроводительная документация является неотъемлемой частью программного проекта. Владелец проекта, реализуемого в данной курсовой работе, предоставил перечень обязательных документов, которые необходимо было подготовить.

Основные документы проекта:

- Документация о системе (на основании ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»)
- Устав проекта
- План управления проектом
- База знаний

Документация о системе представляет собой документ, определяющий функциональные, архитектурные и технические решения проектируемого программного обеспечения. В документе приводятся основные сведения о реализуемом проекте.

Документация о системе реализовывалась на основании ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы» по требованию владельца проекта.

Основными задачами при написании данного документа являлись следующие пункты:

- Описание основных сведений о сторонах заказчика и исполнителя;
- Описание назначения и целей создания программного проекта;
- Описание характеристик объектов автоматизации;
- Определение требований к системе, в том числе функций подразделений ресторана, подлежащих автоматизации;
- Определение требований к содержанию системы, порядку приёма и ввода в эксплуатацию;
- Определение требований к документированию со стороны заказчика.

Была проведена отдельная встреча с разработчиками проекта для достижения максимально точных формулировок по технической части документа. Документация о системе была разработана корректно, в нужные сроки и была предоставлена владельцу проекта.

Устав проекта – это документ, в котором излагается видение проекта, его масштаб, цели, команда проекта и их обязанности, основные заинтересованные стороны, а также то, как он будет осуществляться, план реализации.

Основными задачами при написании данного документа являлись следующие пункты:

- Определение общих положений устава проекта;
- Описание нормативных ссылок на другую документацию, использующихся терминов, обозначений и сокращений;
- Описание измеримых целей и критериев успеха проекта;
- Описание высокоуровневых требований;
- Определение и описание допущений и ограничений;
- Описание совокупного риска проекта;
- Утверждение расписания и ресурсов проекта;
- Определение критериев выхода из проекта;

Устав проекта является важным документом для заинтересованных сторон проекта, так как обеспечивает юридическую безопасность в процессе реализации проекта.

Документация о системе была разработана корректно, в нужные сроки, согласована со Scrum-мастером и была предоставлена владельцу проекта.

План управления проектом (PMBoK) — это документ, описывающий, как проект будет исполняться, как будет происходить его мониторинг и контроль.

План управления проектом включает в себя:

- базовый план по содержанию;
- базовое расписание;
- базовый план по стоимости;
- план управления содержанием;
- план управления требованиями;
- план управления расписанием;
- план управления стоимостью;
- план управления качеством;
- план совершенствования процессов;
- план управления человеческими ресурсами;
- план управления коммуникациями;
- план управления рисками;
- план управления закупками;
- план управления заинтересованными сторонами.

Каждый документ-план определяет деятельность исполнителя проекта по конкретному вопросу, определяющему план.

План управления был написан силами всей команды разработчиков. Мною вносились правки в предоставленные планы при сведении в общий план управления проектом. Документация о системе была разработана корректно, в нужные сроки и была предоставлена владельцу проекта.

База знаний по проектной деятельности (База знаний) – это электронная база знаний по организации деятельности предприятия. Она обеспечивает доступное и единое информационное поле для сотрудников предприятия. Написание и оформление базы знаний также стало задачей для команды от владельца проекта.

Для базы знаний прописывались следующие пункты:

- Сведения о компании;
- Инструкция первого дня;
- Инструкция каждого дня;
- Технические требования;
- Правила поведения;

- Документация (должностные инструкции сотрудников) ;
- Конференции;
- Ответы на часто задаваемые вопросы.

База знаний была написана силами всей команды разработчиков. Мною вносились правки в предоставленные планы при сведении в общий план управления проектом.

При написании всей документации изучались шаблоны написания соответствующих документов.

2.3. Анализ процесса реализации программного проекта по диаграмме сгорания задач

Диаграмма сгорания задач показывает скорость выполнения задач и оставшийся объем работы в течение спринта или проекта. По диаграмме сгорания можно спрогнозировать, успеет ли команда выполнить запланированные задачи и проанализировать работу команды. (10)

По вертикальной оси отмечается суммарный объем задач команды, по горизонтальной – временные промежутки (дни одного спринта или спринты проекта). Таким образом, точка начала проекта располагается в верхнем левом углу, а точка конца проекта – правом нижнем.

На диаграмме можно увидеть две линии:

- Идеальная рабочая линия (голубая). Представляет собой прямую, построенную путем соединения точек начала и конца проекта. Эта линия – идеальный вариант рабочей линии, который показывает, как будут выполняться задачи, если все пойдет по плану;
- Актуальная рабочая линия (оранжевая). График визуализирует реальный ход работы над проектом.

Под анализом диаграммы сгорания задач понимается сравнение положения вышеописанных графиков относительно друг друга. Подобный способ визуализации помогает оценивать работу программы как в масштабах всего проекта, так и в отдельно взятых спринтах.

Диаграмма сгорания задач по проекту данной курсовой работы представлена на рисунке 2.3:

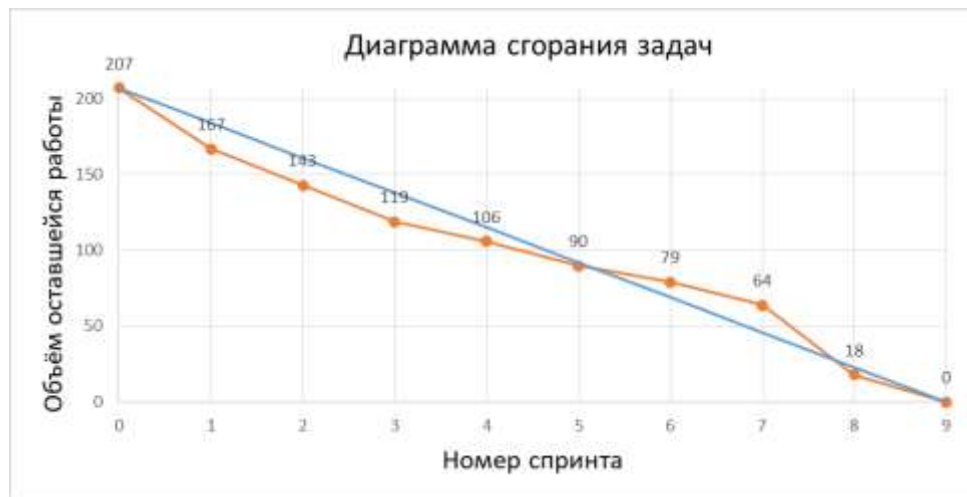


Рисунок 2.3 Диаграмма сгорания задач проекта

Путём сравнения графиков на диаграмме, можно сделать следующие заключения.

Первую половину проекта (1-4 спринты) команда перевыполняла поставленные планы, так как актуальная рабочая линия находится под идеальной рабочей линией. Также можно говорить о том, что составленный план реализации проекта на этот промежуток времени был составлен некорректно.

Начиная со второй половины проекта, команда начинает отставать в выполнении плана. Этот вывод был сделан на основании того, что актуальная рабочая линия на этом промежутке лежит выше идеальной рабочей линии. Это можно связать с некорректно составленным планом работ или с усталостью членов команды.

В самом конце проекта команда снова начинает перевыполнять план по графику и, в итоге, закрывает проект вовремя.

Анализ диаграммы показал, что, несмотря на, местами, приличные отклонения актуальной рабочей линии от идеального графика в обе стороны, команда выполнила работу достаточно эффективно и в поставленный срок.

ЗАКЛЮЧЕНИЕ

В ходе курсовой работы были выполнены следующие задачи:

- Были исследованы методологии и методики управления;
- Была определена рациональная для использования методика управления программным проектом;
- Была изучена выбранную методику управления. Были внедрены основные принципы в работу команду разработчиков;
- Был разработан программный проект по проектированию деятельности ресторана;
- Была подготовлена сопроводительная документация к программному продукту.

Все поставленные задачи были выполнены, что позволяет говорить о достижении цели курсовой работы.

ЛИТЕРАТУРА

1. Швабер Кен Исчерпывающее руководство по Скраму: Правила Игры / Кен Швабер. – Санкт-Петербург : Питер, 2017. – 26 с.
2. Что такое scrum, зачем он нужен, и как работает scrum команда : сайт. – URL: <https://kachestvo.pro/kachestvo-upravleniya/proektnoe-upravlenie/chto-takoe-scrum-zachem-on-nuzhen-i-kak-rabotaet-scrum-komanda/> (дата обращения: 24.11.2022)
3. Scrum Treck : сайт. – URL: <https://scrumtrek.ru/blog/business-agility/1162/leading-and-lagging-indicators/> (дата обращения: 13.10.2022)
4. Модели и методологии разработки ПО // GeekBrains : сайт. – URL: <https://gb.ru/posts/methodologies> (дата обращения: 10.12.2022)
5. Модели и методологии разработки ПО : сайт. – URL: <https://gb.ru/posts/methodologies> (дата обращения: 13.10.2022)
6. Модели жизненного цикла, принципы и методологии разработки программного обеспечения (ПО) // evergreens.com : сайт. – URL: <https://evergreens.com.ua/ru/articles/software-development-metodologies.html> (дата обращения: 10.12.2022)
7. Agile против Waterfall: в чем разница? // IT and Digital : сайт. – URL: <https://itanddigital.ru/agile> (дата обращения: 11.12.2022)
8. Швабер К. Ш. Руководство по Scrum. Исчерпывающее руководство по Scrum: Правила игры / К. Ш. Швабер, Д. С. Сазерленд. – США : а., 2020. – 17 с. – URL: https://de.unecon.ru/pluginfile.php/965269/mod_resource/content/1/2020-Scrum-Guide-Russian.pdf (дата обращения: 20.12.2022). – Режим доступа: для авторизованных пользователей
9. Коул Р. Блистательный Agile. Гибкое управление проектами с помощью Agile, Scrum и Kanban / Р. Коул, Э. Скотчер. – Санкт-Петербург : Питер, 2019. – 304 с. – ISBN 978-5-4461-1051-3
10. Лаунис Р. Топ-7 методов управления проектами: Agile, Scrum, Kanban, PRINCE2 и другие / Р. Лаунис // Проектные сервисы : электронный журнал. – URL: <https://www.pmservices.ru/project-management-news/top-7-metodov-upravleniya-proektami-agile-scrum-kanban-prince2-i-drugie/>. – Дата публикации: 08.05.2016.

ПРИЛОЖЕНИЕ

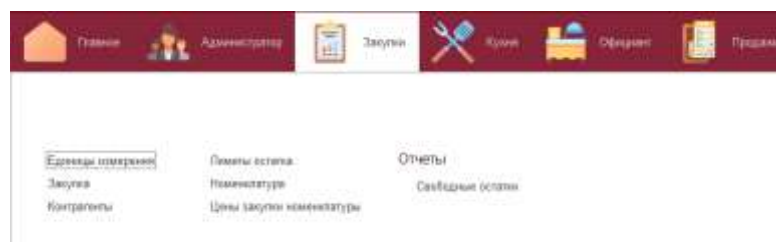
Навигационное меню конфигурации



Блок «Администратор»



Блок «Закупки»



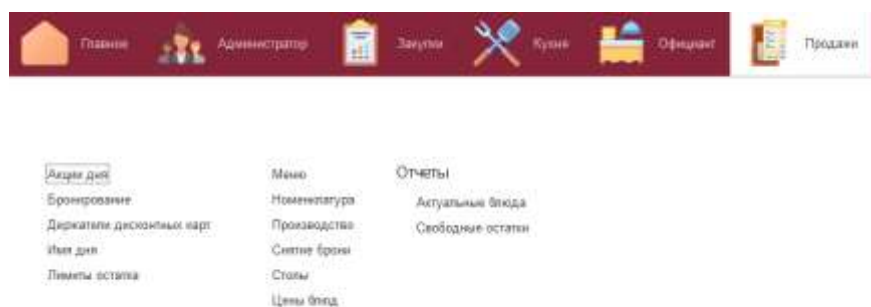
Блок «Кухня»



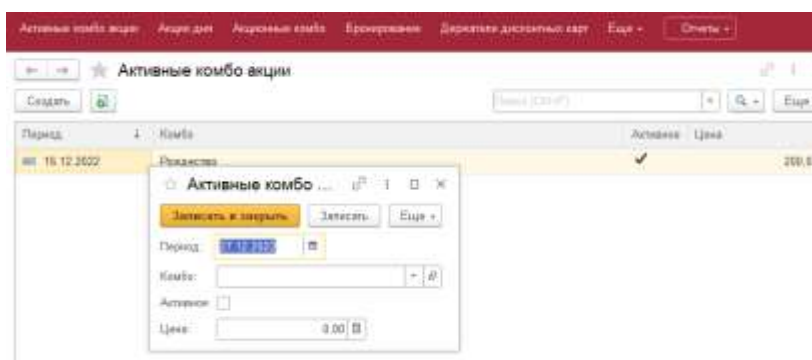
Блок «Официант»



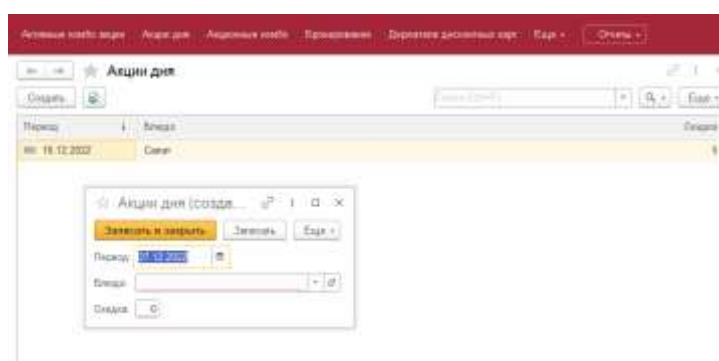
Блок «Продажи»



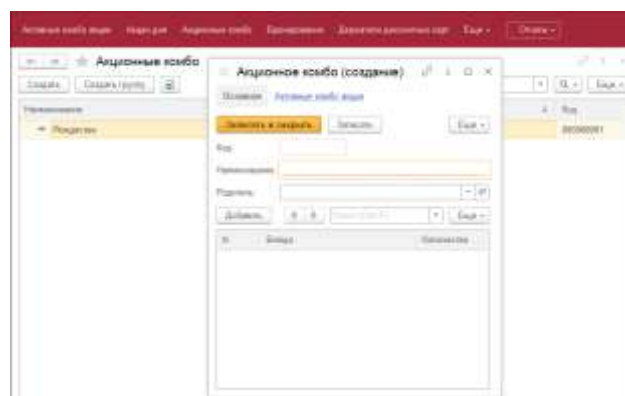
Регистр сведений «Активные комбо акции»



Регистр сведений «Акции дня»



Справочник «Актонные комбо»



Документ «Бронирование»

The screenshot shows the 'Бронирование' (Reservation) document form. It includes a table with columns for 'Дата' (Date), 'Имя' (Name), and 'Номер' (Number). The table contains several rows of reservation data. A modal window titled 'Держатели дисконтных карт (создание)' (Discount card holders (creation)) is open, showing fields for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number).

Справочник «Держатели дисконтных карт»

The screenshot shows the 'Держатели дисконтных карт' (Discount card holders) reference form. It includes a table with columns for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number). A modal window titled 'Держатели дисконтных карт (создание)' (Discount card holders (creation)) is open, showing fields for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number).

Справочник «Единицы измерения»

The screenshot shows the 'Единицы измерения' (Units of measurement) reference form. It includes a table with columns for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number). A modal window titled 'Единицы измерения (создание)' (Units of measurement (creation)) is open, showing fields for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number).

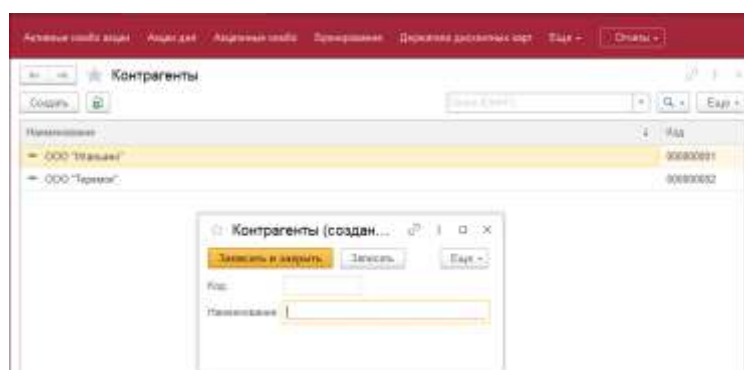
Документ «Закупка»

The screenshot shows the 'Закупка' (Purchase) document form. It includes a table with columns for 'Дата' (Date), 'Имя' (Name), 'Единица измерения' (Unit of measurement), and 'Пол' (Gender). The table contains several rows of purchase data. A modal window titled 'Единицы измерения (создание)' (Units of measurement (creation)) is open, showing fields for 'Имя' (Name), 'Пол' (Gender), 'Дата рождения' (Date of birth), 'Паспорт' (Passport), 'Имя отчество' (Name and patronymic), 'Пол' (Gender), 'Дата рождения' (Date of birth), and 'Номер дисконтной карты' (Discount card number).

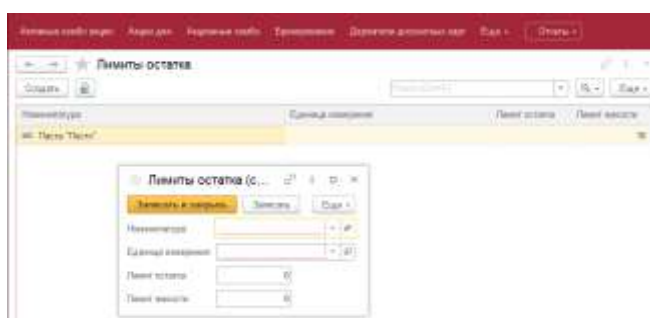
Регистр сведений «Имя дня»



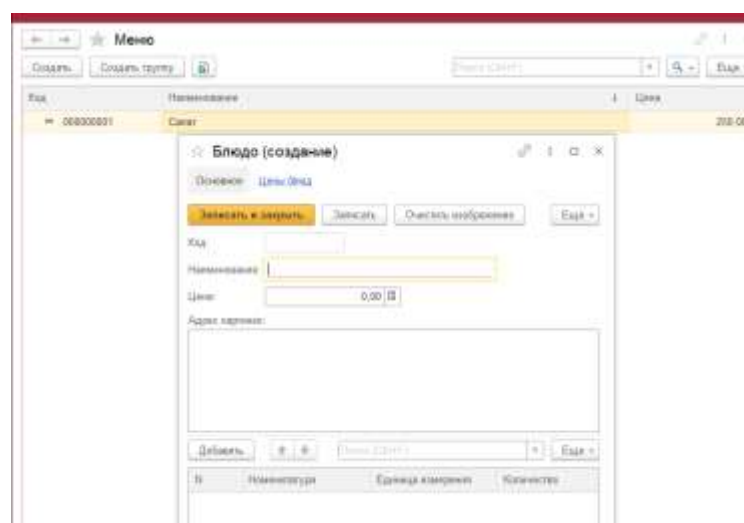
Справочник «Контрагенты»



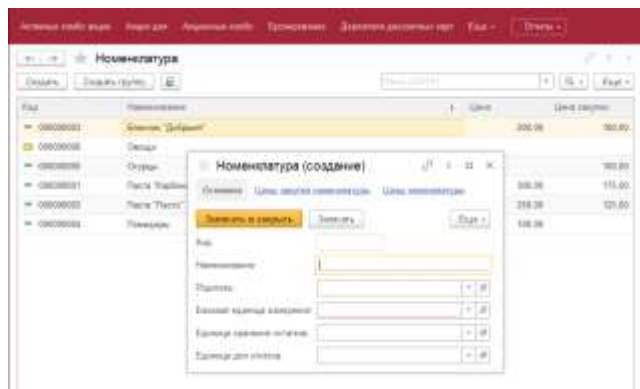
Регистр сведений «Лимиты остатка»



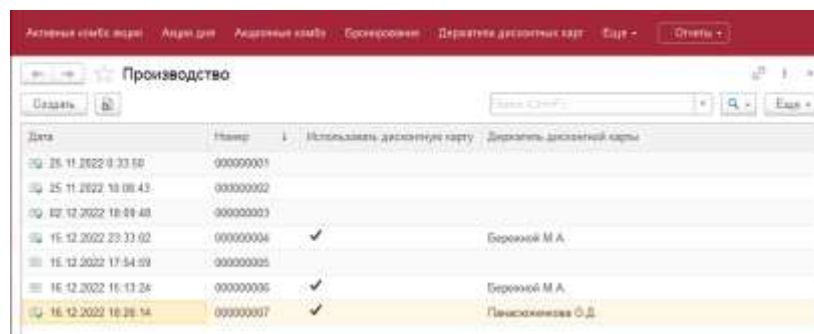
Справочник «Меню»



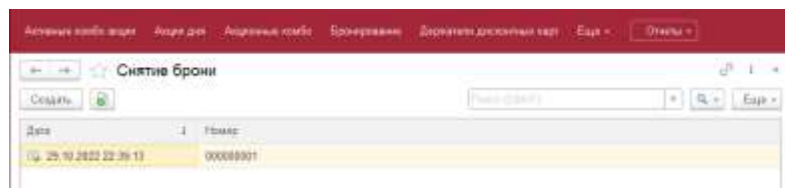
Справочник «Номенклатура»



Документ «Производство»



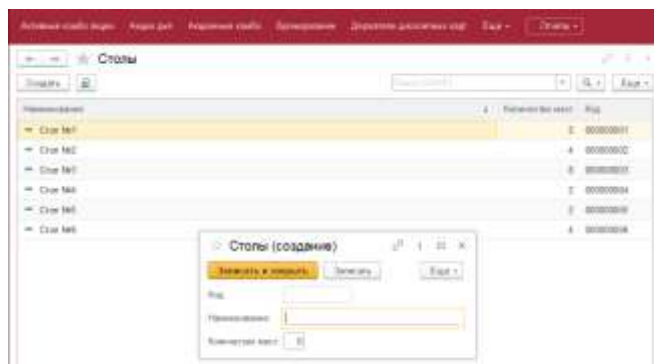
Документ «Снятие брони»



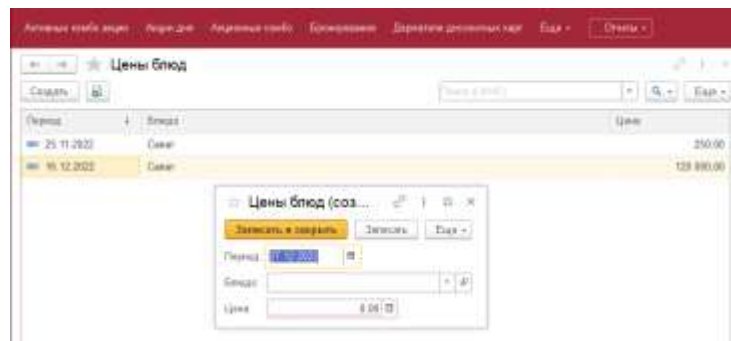
Документ «Списание продуктов»



Справочник «Столы»

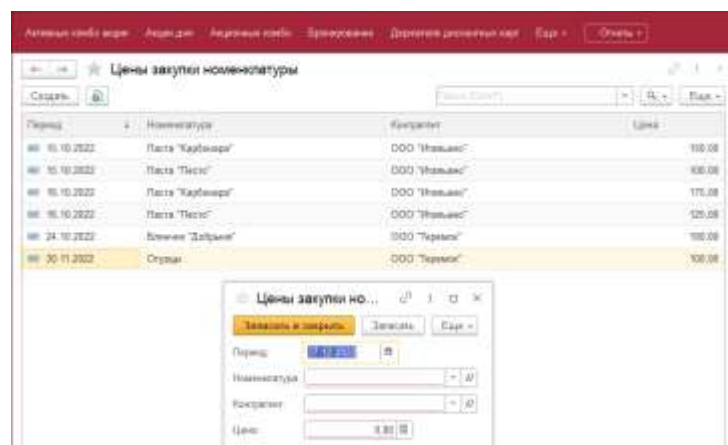


Регистр сведений «Цены блюд»



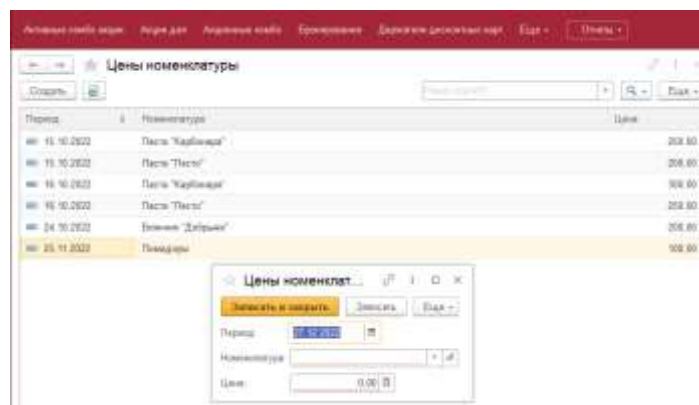
Период	Наименование	Цена
25.11.2022	Салат	250.00
16.12.2022	Салат	129.890.00

Регистр сведений «Цены закупки номенклатуры»



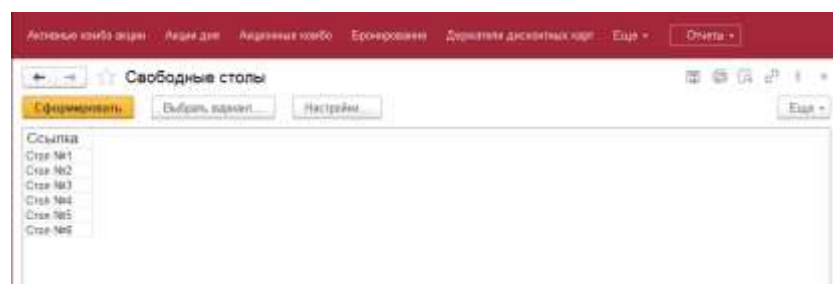
Период	Наименование	Поставщик	Цена
16.10.2022	Паста "Карбонара"	ООО "Италия"	110.00
16.10.2022	Паста "Песто"	ООО "Италия"	100.00
16.10.2022	Паста "Карбонара"	ООО "Италия"	170.00
16.10.2022	Паста "Песто"	ООО "Италия"	125.00
24.10.2022	Винегрет "Добрый"	ООО "Пермь"	100.00
30.11.2022	Орехи	ООО "Пермь"	300.00

Регистр сведений «Цены номенклатуры»



Период	Наименование	Цена
16.10.2022	Паста "Карбонара"	200.50
16.10.2022	Паста "Песто"	200.00
16.10.2022	Паста "Карбонара"	300.00
16.10.2022	Паста "Песто"	250.50
24.10.2022	Винегрет "Добрый"	200.00
25.11.2022	Пива	100.00

Отчет «Свободные столы»



Отчет «Актуальная бронь на данный момент»

Актуальная бронь на данный момент

Сформировать | Выбрать вариант... | Настройки... | Блок...

Параметры: Период: 27.12.2022 0:00:00

Начало	Окончание	Зарезервированный столы
--------	-----------	----------------------------

Отчет «Свободные остатки»

Акционные комбо | Меню | Производство | Списание продуктов | **Отчеты**

← → ☆ **Свободные остатки**

Сформировать | Выбрать вариант... | Настройки...

Номенклатура	Количество Остаток
Блинчик "Добрыня"	3,000
Огурцы	193,000
Паста "Карбонара"	10,000
Паста "Песто"	5,000
Помидоры	47,000