



# Пример планирования проекта по методологии Scrum

Объектно-ориентированное программирование  
Belarussian State University

15 pag.

---

---

---

---

---

---

---

---

Разработать план управления разработкой приложения «Заметки» (Notes) в соответствии с методологией SCRUM.

### Этап 1. Разработка Product backlog

Таблица 1 - Product backlog

ID	Название	Важность	Предварительная оценка	Демонстрация	Примечание
R1	Изменение размеров форм (как в ширину, так и в высоту)	85	6	Выделение области формы заметки	
R2	Задание размера, типа, цвета шрифта текста (HEX-код)	80	4	Выделение текста заметки, кнопка "Шрифт"	
R3	Создание маркеров и нижних/верхних индексов	80	2	Выделение текста заметки, кнопка "Маркер"	
R4	Выделение шрифта курсивом, полужирным и подчеркнутым цветом	75	1	Выделение текста заметки	
R5	Выравнивание текста по центру, левому и правому краям	75	3	Выделение текста заметки, кнопка "Выравнивание"	
R6	Создание списка задач для каждой заметки	73	3	Добавление заметки, кнопка "Добавить задачу"	
R7	Поиск данных по базе	71	5	Ввод данных в поле "Поиск", нажатие на кнопку "Найти"	
R8	Ведение истории поиска	70	2	Появление вариантов поиска при вводе символов	
R9	Добавление к заметке графических файлов	65	4	Кнопка "Добавить изображение"	
R10	Вывод списка задач на печать	65	3	Кнопка "Печать"	
R11	Присвоение задачам приоритетов	60	3	Кнопка "Приоритет"	
R12	Перемещение задач между собой	55	4	Выделение и перетаскивание задач в рабочей области	

Продолжение таблица 1 - Product backlog

R13	Режим рисования курсором мышки	50	4	Кнопка "Рисование"	
R14	Выбор цвета курсора мыши в режиме рисования	50	3	Кнопка "Цвет курсора"	
R15	Автосохранение каждые 30 секунд	45	2	Ожидание 30 секунд, проверка записи в файл	
R16	Сохранение txt-файлов	45	3	Кнопка "Сохранить"	
R17	Импорт txt-файлов	40	2	Кнопка "Импорт"	
R18	Размер приложения до 30 Мб	40	10	Проверка размера приложения	
R19	Краткое техническое руководство по использованию приложения	35	8	Запуск текстового файла руководства	
R20	Кроссплатформенность	20	9	Проверка на существующих платформах	
R21	Календарь	20	4	Отображение формы "Календарь" в левой части рабочей области	

Product backlog составляется на основании общения команды разработчиков (**Core roles**) со стороной заказчика (**Ancillary roles**). Для приложения «Заметки» Product backlog содержит 21 user story. Каждая user story имеет уникальный идентификатор (ID), название, приоритет по 100-бальной шкале, предварительную оценку объема работ в story point's (определяется как  $S=A*B$ , где  $A$  – количество исполнителей задачи, человек;  $B$  – количество рабочих дней, которые исполнители  $A$  будут работать над задачей, дни), демонстрацию (описание необходимых действий и их результатов, предназначенных для проверки работоспособности функциональностей) и колонку примечаний (комментариев).

## Этап 2. Определение состава команды и обязанностей ее членов.

На основании используемой в современных ИТ-компаниях структуры специалистов, была сформирована команда разработчиков, а также сторона заказчика, представленные в таблице 2. Дополнительно, в процессе разработки принимает участие Scrum-мастер, выполняющий организаторские функции и следящий за корректным соблюдением методологии Scrum.

Таблица 2 - Scrum-роли

Тип-роли	Название роли	Описание
Core roles	Аналитик	Общение с заказчиком по согласованию концепции и требований желаемого приложения. Разработка ТЗ для команды в техническом виде.
	Архитектор	Разработка архитектуры приложения. Выбор языка программирования, технологий, средств реализации, шаблонов проектирования. Разработка модульной структуры приложения, диаграмм классов.
	Дизайнер UI	Создание внешнего вида приложения, разработка цветовой гаммы элементов, деталей анимации и визуализации.
	Front-end программист	Разработка структуры пользовательского интерфейса, визуальных функциональных компонентов. Реализация взаимодействия между пользователем и приложением, определение взаимосвязей между компонентами.
	Back-end программист	Разработка серверной части и внутренней механики обработки пользовательских действий и запросов. Реализация способов хранения данных, расчетных алгоритмов. Создание структуры БД и обеспечение ее взаимодействия с приложением.
	Тестировщик №1	Проведение автоматизированных тестов и white-box тестирование приложения.
	Тестировщик №2	Проведение различных видов тестирования приложения (позитивного, функционального, регрессионного, интеграционного, системного, не функционального).
	Интегратор	Написание технической документации для заказчика, внедрение приложения на оборудовании заказчика, обучение пользованию разработанным приложением.
Ancillary roles	Клиент (заказчик)	Формулирование концепции, основных требований, целей, предметной области требуемого приложения команде разработчиков. Внесение изменений в требования, постановка технических задач в общем виде.
	Менеджер заказчика по техническим вопросам	Формирование предложений и требований к разрабатываемому приложению с учетом специфики производственного процесса компании заказчика. Участие в тестировании приложения.
	Сотрудник отдела технической поддержки компании заказчика	Согласование технических вопросов по эксплуатации разработанного приложения. Участие в тестировании приложения. Высказывание своих замечаний и пожеланий.

### Этап 3. Составление Sprint backlogs

На основании проведения Sprint Planning Meeting и с учетом специфики разработки приложения «Заметки» разработано 2 Sprint backlog'а, длительностью 3 рабочих недели(15 дней) каждый, с рабочим графиком команды разработчиков с 8.30 до 19.00. Результаты представлены в таблицах 3 и 4.

Таблица 3– Sprint backlog #1

ID	Название	Важность	Предварительная оценка	Демонстрация	Примечание
R1	Изменение размеров форм (как в ширину, так и в высоту)	85	1	Выделение области формы заметки	Использование фреймворка Swing
R2	Задание размера, типа, цвета шрифта текста (HEX-код)	80	4	Выделение текста заметки, кнопка "Шрифт"	Разбить на несколько классов
R4	Выделение шрифта курсивом, полужирным и подчеркнутым цветом	75	1	Выделение текста заметки	-
R6	Создание списка задач для каждой заметки	70	3	Добавление заметки, кнопка "Добавить задачу"	Реализовать по иерархии наследования
R7	Поиск данных по базе	70	5	Ввод данных в поле "Поиск", нажатие на кнопку "Найти"	Заполнить базу тестовыми данными разных типов
R9	Добавление к заметке графических файлов	65	4	Кнопка "Добавить изображение"	Форматы jpg и png
R11	Присвоение задачам приоритетов	65	3	Кнопка "Приоритет"	Высокий, средний, низкий
R12	Перемещение задач между собой	55	4	Выделение и перетаскивание задач в рабочей области	Использование фреймворка Swing
R15	Автосохранение каждые 30 секунд	45	2	Ожидание 30 секунд, проверка записи в файл	В отдельный поток
R16	Сохранение txt-файлов	45	3	Кнопка "Сохранить"	Предусмотреть безопасный режим
R21	Календарь	20	4	Отображение формы "Календарь" в левой части рабочей области	Синхронизация по текущему дню

Таблица 4 – Sprint backlog #2

ID	Название	Важность	Предварительная оценка	Демонстрация	Примечание
R3	Создание маркеров и нижних/верхних индексов	80	1	Выделение текста заметки, кнопка "Маркер"	-
R5	Выравнивание текста по центру, левому и правому краям	70	1	Выделение текста заметки, кнопка "Выравнивание"	Отдельный компонент
R8	Ведение истории поиска	70	2	Появление вариантов поиска при вводе символов	Запросы через Hibetnate
R10	Вывод списка задач на печать	65	3	Кнопка "Печать"	Поиск библиотеки
R13	Режим рисования курсором мышки	50	4	Кнопка "Рисование"	Canvas
R14	Выбор цвета курсора мыши в режиме рисования	50	3	Кнопка "Цвет курсора"	Canvas
R17	Импорт txt-файлов	40	2	Кнопка "Импорт"	Парсер через xml
R18	Размер приложения до 30 мб	40	10	Проверка размера приложения	-
R19	Краткое техническое руководство по использованию приложения	35	8	Запуск текстового файла руководства	Добавить скриншоты
R20	Кроссплатформенность	20	9	Проверка на существующих платформах	Mac, Android, Windows

Целью Sprint №1 является разработка базовой функциональной части приложения, пригодного для использования. Целью Sprint №2 является дополнение функционала приложения для расширения его возможностей и внедрения.

#### **Этап 4. Определение технических средств реализации и разработка структурной схемы этапов разработки приложения**

На основании анализа современных методов и средств, используемых при разработке программных продуктов, для разработки приложения «Заметки» были выбраны средства реализации (приведены в таблице 5). С учетом основных положений Scrum-методологии была разработана структурная схема этапов недельной разработки приложения «Заметки»(рис.1).

Таблица 5 – Выбранные средства реализации приложения

<b>Используемые средства реализации</b>	<b>Название</b>
Язык программирования	Java
Интегрированная среда разработки	NetBeans
Библиотека тестирования	Junit
Сервер приложений	GlassFish
Фреймворк автоматизации сборки	Maven
Библиотека для объектно-реляционного отображения	Hibernate
Система контроля версий	Git
Система непрерывной интеграции	Hudson
Система отслеживания ошибок	BugZilla

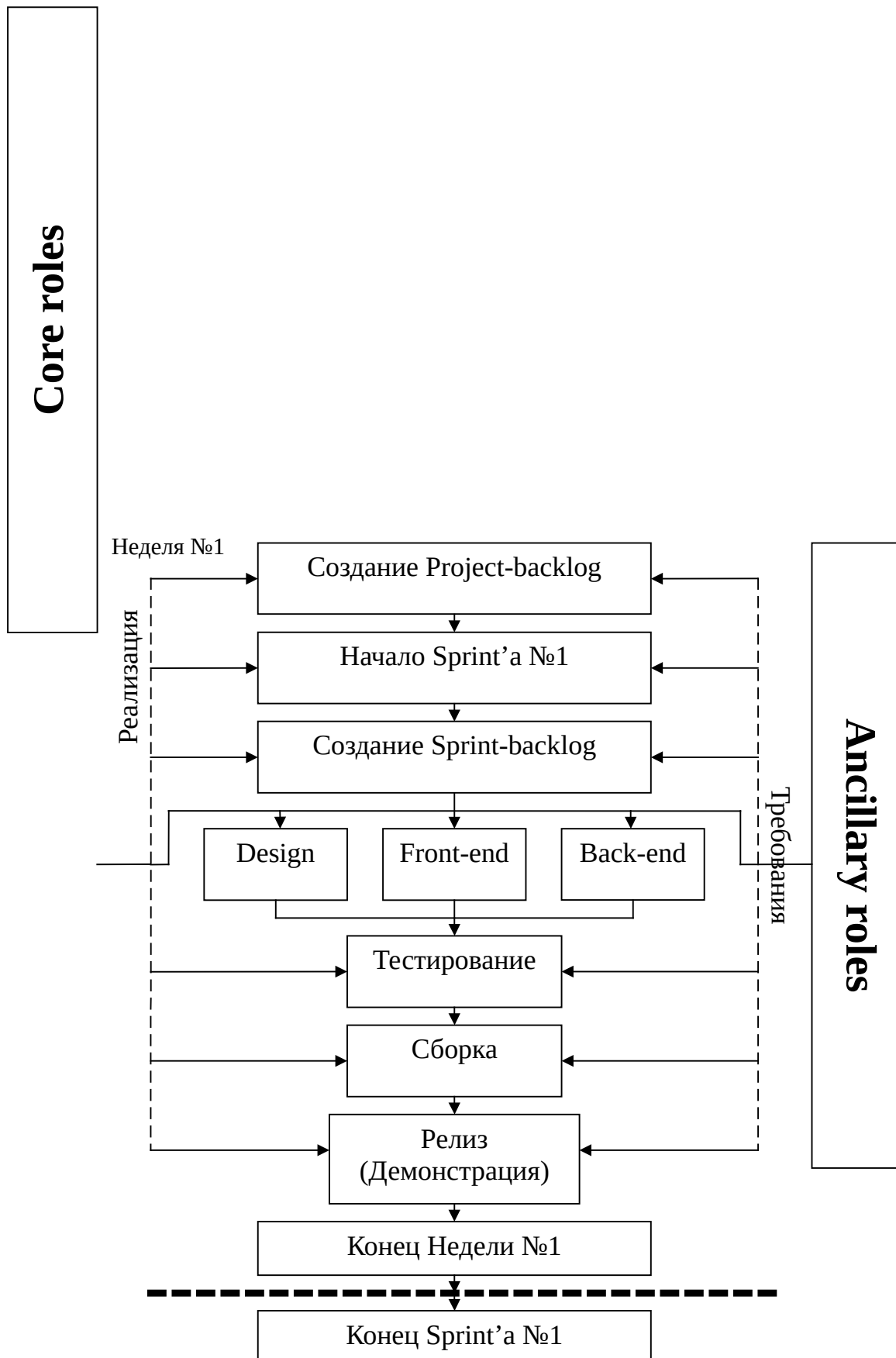


Рис.1 – Структурная схема этапов недельной разработки приложения «Заметки»



Этапы разработки приложения «Заметки» (рис.1.) детальнее состоят из:

- Создание Project-backlog:
  - Согласование всех требований и деталей реализации с заказчиком;
  - Распределение ответственности по выполнению технических задач между исполнителями;
- Создание Sprint-backlog:
  - Выбор длины Sprint'а равным 3 недели;
  - Формулирование инкремента продукта (выполнение всех высоко-приоритетных задач и получение корректно работающего приложения согласно установленным требованиям заказчика);
  - Расстановка приоритетов поставленных задач;
  - Определение фокус-фактора;
  - Оценка трудозатрат и распределение времени на выполнение отдельных задач по конкретным исполнителям;
- Разработка графического дизайна:
  - Инфографическая концепция приложения;
  - Цветовая гамма шрифтов, форм, окон, компонентов;
  - Геометрические пропорции и распределение рабочей области;
- Разработка пользовательского интерфейса(front-end):
  - Реализация форм и окон;
  - Реализация визуальных функциональных компонентов;
  - Организация взаимодействия между визуальными компонентами;
  - Написание модульных тестов;
- Разработка серверной части(back-end):
  - Разработка структуры БД;
  - Разработка компонентов обработки запросов;
  - Организация протоколов обмена данными;
- Тестирование:
  - Позитивных сценариев;
  - Функциональной части;

- Регрессионное;
- Интеграционное;
- Системное;
- Не функциональной части;
- Негативных сценариев;
- Сборка:
  - Скачивание разработанных программных модулей на удаленную машину из репозитория;
  - Запуск тестов в автоматическом режиме;
  - Развертывание приложения;
  - Проверка работоспособности приложения;
- Релиз:
  - Формирование презентационных материалов;
  - Демонстрация заказчику реализованного функционала приложения, в соответствии с согласованным Sprint-backlog (час в неделю);
  - Предоставление заказчику технической документации;
  - Получение обратной связи со стороны заказчика;
  - Планирование и разработка нового Sprint-backlog.

**Этап 5. Разработка структуры нормативных документов(планов) собраний по Scrum-методологии**

Таблица 5 – План Sprint Planning Meeting

Тема	Время начала	Время окончания	Описание
Цели	9.00	9.50	Формулирование Product owner'ом главных целей конкретного спринта, информирование о специфике бизнес-процессов, которые отражены в Project backlog'e. Выбирается время и место проведения будущей демонстрации.
Оценка задач	10.00	11.30	Проведение командой разработчиков оценки временных затрат на разработку бизнес-процессов. Декомпозиция особо емких задач на подзадачи. Согласование с Product owner'ом приоритетов выполнения задач. Для наиболее критичных задач формируются критерии демонстрации.

User Story	11.40	12.30	Формирование набора user story, входящих в следующий спринт. Определение производительности команды.
Организация	12.30	13.00	Обсуждение места и времени проведения Daily Scrum meeting. Разбиение user story на задачи.

**Таблица 6 – Пример плана Daily Scrum meeting**

Докладчик	Время начала	Время окончания	Сделано	Будет сделано	Проблемы
Аналитик	9.00	9.03	Сформулировано описание функциональности добавления задачи	Структура хранения данных при импорте	-
Архитектор	9.03	9.06	Разработана диаграмма классов модуля экспорта данных	Изменение шаблона проектирования для модуля UI	-
Дизайнер UI	9.06	9.09	Разработано графическое оформление главной формы	Цветовая гамма при выборе шрифтов	Нужно мнение заказчика
Front-end программист	9.09	9.12	Созданы визуальные компоненты управления формой заметки	Создание компонентом окна создания маркеров	-
Back-end программист	9.12	9.15	Написан модуль поиска по БД	Добавление возможности отображения истории поиска	Отсутствие диаграммы классов модуля
Тестировщик №1	9.15	9.18	Протестированы функциональности создания заметки	Тестирование совместного использования выделения шрифтов	Отсутствие окна создания маркеров
Тестировщик №2	9.18	9.21	Протестирована корректная работы автосохранения	Нагрузочное тестирование сервера	Написать скрипт
Интегратор	9.21	9.24	Написана документация по работе с формой заметок	Создание документации по развертыванию приложения	-

**Таблица 7 – Пример плана Sprint review meeting**

Докладчик	Инкремент продукта	Время начала	Время окончания	Индикатор
Аналитик	Главные преимущества разработанного приложения по сравнению с	15.00	15.30	Реализованные функциональности согласно sprint backlog

	продуктами на рынке			
Архитектор	Архитектура клиент-серверного приложения "Заметки"	15.30	16.00	Разработанные диаграммы классов
Дизайнер UI	Зрительное удобство работы с приложением	16.00	16.30	Созданная цветовая гамма форм и окон приложения

Таблица 7 – Продолжение примера плана Sprint review meeting

Front-end программист	Эргономичность UI приложения	16.30	17.00	Разработанные визуальные компоненты
Back-end программист	Надежность и безопасность работы приложения	17.00	17.30	Внедренные технологии шифрования данных, механизмы защиты доступа к БД
Тестировщик №1	Низкая требовательность к ресурсам	17.30	17.45	Результаты тестирования на различных архитектурах
Тестировщик №2	Корректность работы приложения при больших нагрузках	17.45	18.00	Результаты тестирования приложения на различных нагрузках

Таблица 8– Пример плана Retrospective meeting

Докладчик	Время начала	Время окончания	Замечания	Предложения
Аналитик	12.00	12.15	Отсутствие возможности ведения статистики	Использование облачного сервиса для сбора статистики
Архитектор	12.15	12.30	-	-
Дизайнер UI	12.30	12.45	-	Возможность подключения сторонних шрифтов
Front-end программист	12.45	13.00	Усложнение UI из-за управляющих кнопок	Подробнее обсудить с заказчиком
Back-end программист	13.00	13.15	Не достаточный уровень надежности шифрования данных	Внедрение новых механизмов по защите данных
Тестировщик №1	13.15	13.30	Необходимость написания автоматизированных тестов	Покупка скрипта автоматизации
Тестировщик №2	13.30	13.45	Большое количество багов при импорте данных	Внесение исправлений в код импорта данных
Интегратор	13.45	14.00	Необходимость добавление функциональностей для шрифтов	-

## Этап 6. Разработка диаграмм сгорания задач

Согласно данным в таблицах 9 и 10 были построены диаграммы сгорания задач для Sprint №1 и №2 (рис.2,3). Итоговая диаграмма сгорания задач для выпуска проекта представлена на рис.4.

Таблица 9 – Данные статистики выполнения задач по Sprint backlog №1

<b>Спринт№1</b>			
<b>Дни</b>	<b>Часы</b>	<b>Задачи</b>	<b>Оставшиеся задания</b>
0	150	11	11
1	140	11	11
2	130	10	11
3	120	9	10
4	110	9	10
5	100	8	10
6	90	6	6
7	80	6	5
8	70	5	8
9	60	4	7
10	50	4	6
11	40	3	5
12	30	2	4
13	20	1	5
14	10	1	2
15	0	0	0

Таблица 10 – Данные статистики выполнения задач по Sprint backlog №2

<b>Спринт№2</b>			
<b>Дни</b>	<b>Часы</b>	<b>Задачи</b>	<b>Оставшиеся задания</b>
0	150	10	10
1	140	9	8
2	130	9	7
3	120	8	9
4	110	7	9
5	100	7	9
6	90	7	8
7	80	7	5
8	70	6	4
9	60	6	4
10	50	5	3
11	40	4	3
12	30	3	4
13	20	2	2
14	10	1	1
15	0	0	0



Рис. 2 – Диаграмма сгорания задач для Sprint №1

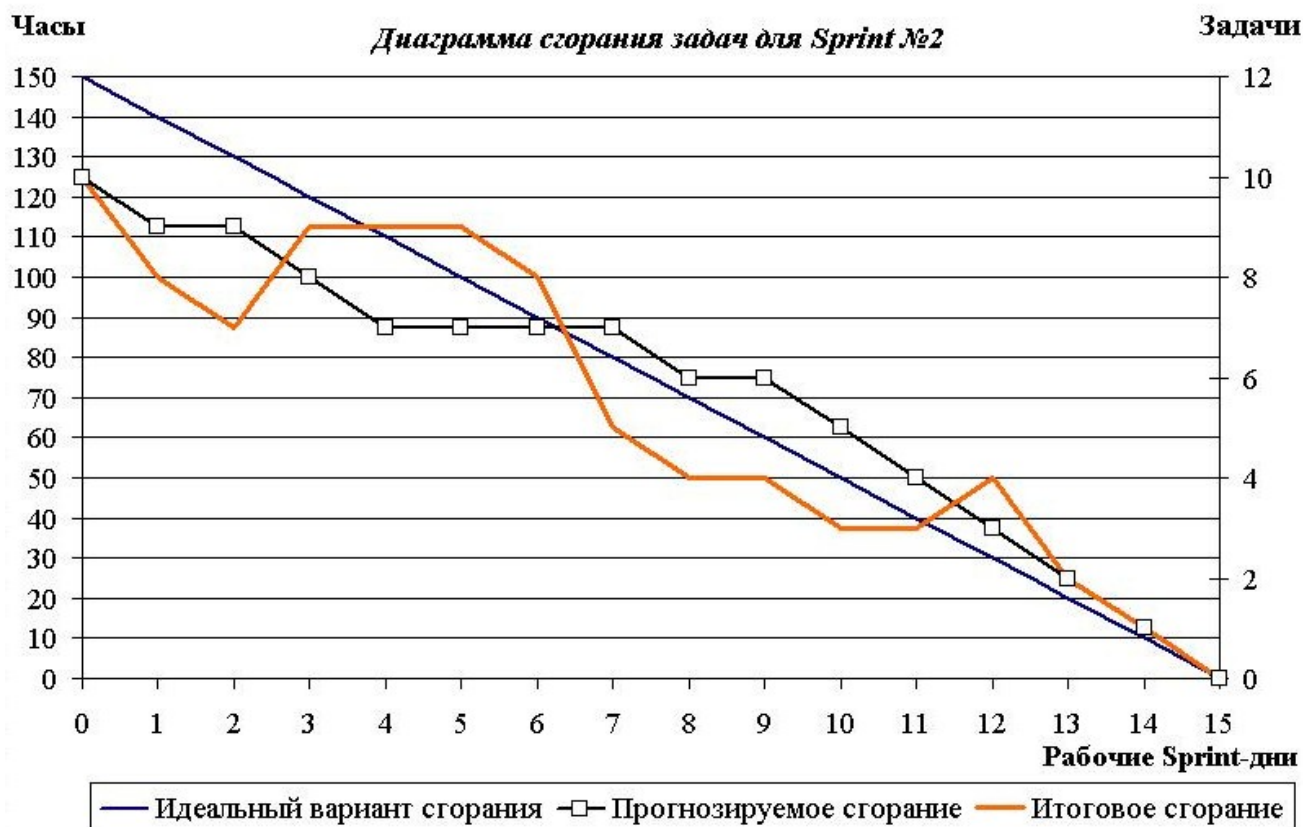
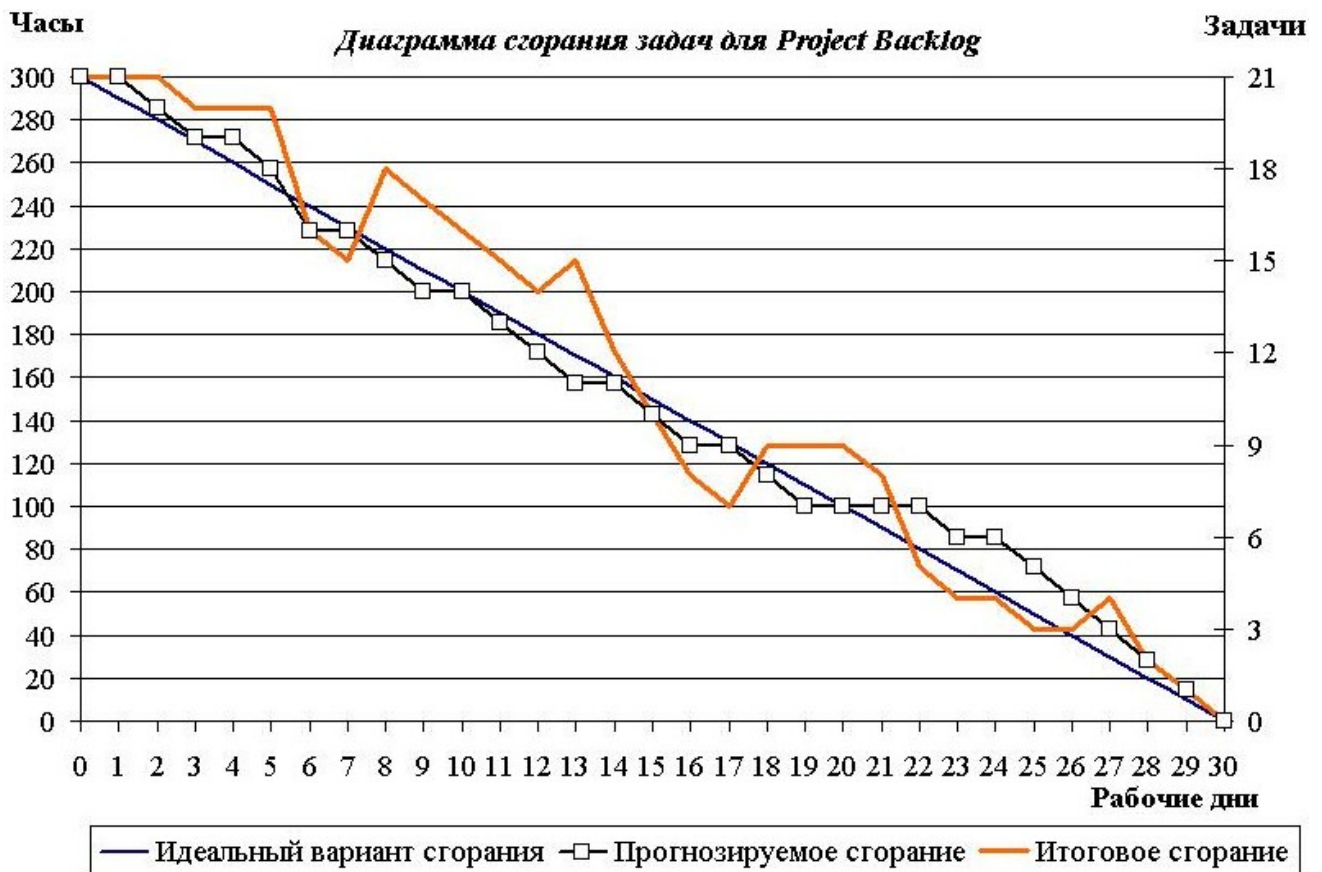


Рис. 3– Диаграмма сгорания задач для Sprint №2



Результат выполнения Sprint №1.

- Реализованы user stories: R1, R2, R4, R6, R7, R9, R11, R12, R15, R16, R21;
- Представлено работающее приложение «Заметки» заказчику;
- Рассмотрены все инкременты продукта за пройденный Sprint;
- Получена обратная связь со стороны заказчика;
- Составлен Sprint backlog №2.

Результат выполнения Sprint №2.

- Реализованы user stories: R3, R5, R8, R10, R13, R14, R17, R18, R19, R20;
- Представлено работающее приложение «Заметки» заказчику с дополненным функционалом;
- Рассмотрены все инкременты продукта за пройденный Sprint;
- Получена обратная связь со стороны заказчика;
- Проект интегрирован и внедрен на стороне заказчика;
- Осуществляется поддержка разработанного программного продукта.

Основные вехи проекта представлены в таблице 11.

Таблица 11 – Основные вехи проекта

Название	Описание	Длительность	Результат
----------	----------	--------------	-----------

Product backlog	Составление списка user stories, согласование всех деталей со стороной заказчика	4 часа	Разработанный Product backlog на 21 user stories, с расстановкой приоритетов и трудозатрат
Sprint Planning Meeting	Согласование перечня задач в рамках выполнения Sprint'a, длительность 3 недели(15 дней)	4 часа	Разработанные Sprint backlogs на 11 и 10 user stories, с расстановкой приоритетов, трудозатрат и примечаний по реализации
Sprint review meeting	Демонстрация стороне заказчика завершённой функциональности приложения в конце Sprint'a	3 часа	Получена обратная связь со стороной заказчика, уточнены требования и детали дальнейшей разработки функциональностей
Retrospective meeting	Обсуждение члена команды разработчиков пройденного Sprint'a. Выяснение качества реализованных задач и определение действий для повышения качества и эффективности работы на следующем Sprint'e	2 часа	Определены положительные и отрицательные результаты прошедшего Sprint'a. Зафиксированы идеи и пожелания для дальнейшей разработки приложения на следующем Sprint'e.