

Технологии разработки ПО. Командная работа над проектом

Цель работы:

- изучение жизненного цикла разработки программного обеспечения (ПО);
- изучение методологии проектирования ПО;
- ознакомление с гибкой (Agile) методологией разработки ПО;
- выполнение проектирования и разработки проекта в целом и его отдельных модулей;
- формирование списка требований;
- построение диаграммы вариантов использования;
- выполнение тестирования;
- тестирование;
- рефакторинг;
- документирование проекта;
- внедрение и сопровождение ПО;
- совместная работа в команде.

ВВЕДЕНИЕ.

Обозначения:

Владелец продукта (Product Owner) – преподаватель.

Скрам-мастер (Scrum master) – активный, заинтересованный студент, который координирует процесс, проводит ежедневные собрания (Scrum Meetings).

Скрам-команда (Scrum team) – команда из 5 человек, работающая над реализацией проекта.

Начальные требования к проекту/модулю предоставляются в виде списка задач - **Бэклог** (backlog), из которого формируется **Бэклог спринта** (sprint backlog).

Бэклог продукта (Product backlog) представляет собой упорядоченный по степени важности список требований, предъявляемых к разрабатываемому продукту.

Элементы этого списка называются **Пользовательскими историями** (User story).

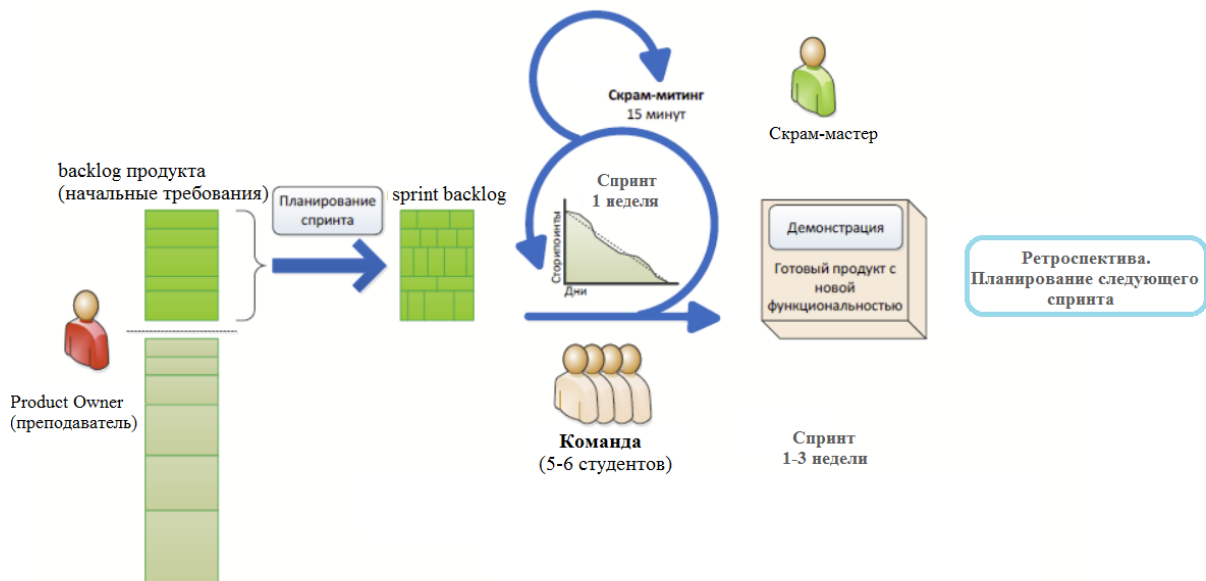
Шаблон пользовательской истории:

Как <персонаж>, Я хочу <что?>, Для того, чтобы <зачем>.

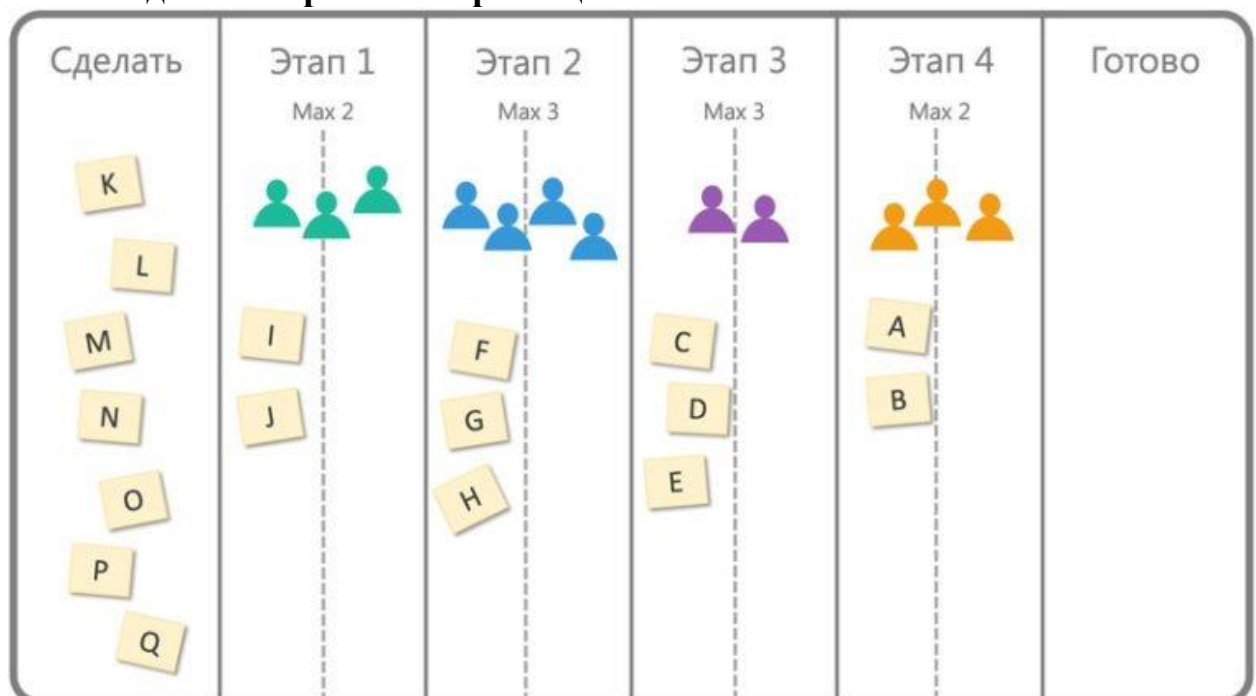
Пример пользовательских историй из бэклога продукта.

ID	User Story
a-001	Как <i>Пользователь</i> , я хочу <i>использовать «калькулятор»</i> , чтобы <i>выполнять арифметические операции</i> .

Схема процесса работы по Scrum:



Канбан-доска: карточки перемещаются по колонкам-этапам



Задание

1. Используйте при выполнении лабораторной работы материал лекций 12-17.
2. В качестве методологии процесса разработки ПО использовать гибкие методологии разработки *Agile/Scrum*, *Agile/Kanban*, методологию *XP* или на выбор.

Создание команды и знакомство с проектом.

1. Разделиться на 3 *Scrum*-команды (5 человек в команде), определить и обсудить *Scrum*-роли (можно прибегнуть к жеребьевке). Каждая такая группа будет являться *Scrum*-командой, работающей над реализацией проекта в рамках одного *sprint*.
2. Провести начальное планирование проекта, который предстоит реализовать. Определить назначение, цель проекта, модули, входящие в него.
3. Уточнить неясные вопросы по заданию с *Product Owner*.
4. Создать командный проект – результат совместной работы *Scrum*-команды.
5. Подготовить следующие документы:
 - a. Краткое описание основной концепции разрабатываемого проекта.
 - b. Список требований в формате *product backlog*.
 - c. Список пользовательских историй из *product backlog*

Создание проекта.

1. Создать новый удаленный репозиторий в **GitHub** для вашего проекта.
2. Добавить членов команды. Настроить права доступа для членов команды.
3. Сформировать *список требований*, предъявляемых к проекту. Подробно рассмотреть пользовательские истории.
4. Обсудить возникшие вопросы с *Product Owner*.
5. Назначить ответственного (*Скрам-мастер*) за проект, распределить модули среди участников проекта.
6. Составить *список пользовательских историй*, описывающих необходимую функциональность *каждого модуля*. Описание должно быть достаточно детальным, чтобы максимально точно определить структуру модуля.
7. Отчитаться перед *Product Owner* о том, какие пользовательские истории были запланированы.

Планирование спринта и создание Бэклога спринта.

1. Назвать проект (решает команда).
2. В качестве *sprint* указать *Release1/Sprint1*.
3. Организовать совместную работу одним из способов:
 - создать *канбан*-доску, добавить в нее задачи, назначить исполнителей;
 - организовать обсуждение, назначение задач/проблем членам команды средствами *Issues* в *GitHub*;

- в порядке обсуждения распределить роли и задачи между членами команды и согласовать правила совместной работы с удаленным репозиторием.

Составить список пользовательских историй для каждого модуля и проекта в целом.

Построить для каждого модуля диаграмму вариантов использования.

Разработку проекта вести в IDE Visual Studio с использованием системы контроля версий.

Разработать систему тестов и составить набор тест кейсов. Выполнить несколько видов тестирования.

Представить готовый Release1, проанализировать полученный продукт. Принять решение о продолжении/завершении работы над проектом.

При необходимости выполнить планирование следующего спринта.

Разработать документацию на проект.

Представить готовый проект.

Варианты заданий

Назначение; цель; имя проекта; описание; входные/выходные данные проекта/модуля; модули, входящие в состав проекта, определяются командой разработчиков и документируются.

Для лидера команды:

1. создать html-страницу, в которой представить отчет по проекту, обязательна следующая информация:
 - a. Название проекта.
 - b. Цель проекта.
 - c. Дата начала и завершения работ над проектом.
 - d. Список членов команды, название разработанного каждым участником модуля.
 - e. Оценка вклада каждого по 10-балльной шкале (по желанию).
 - f. Документация (можно использовать средства **Wiki** в GitHub)
 - g. Другое.

Список модулей, которые можно включить в проект (на выбор команды):

1. Планирование дел на день и контроль за их выполнением (можно добавить текстовые напоминания).
2. Программа для чтения текстовых файлов.
3. Календарь (определить по текущей дате: год, месяц, день недели, порядковый номер дня в году, сколько дней осталось до Нового года).
4. Простой калькулятор для работы с целыми числами (арифметические операции).
5. Часы - вывод текущего времени (формат, частоту обновления выбрать самостоятельно).
6. Калькулятор для работы со строками (сравнение строк, поиск подстроки в строке, замена подстроки в строке, счетчик гласных в строке).
7. Программа для сохранения заметок (в стиле блокнота).
8. Список студентов в группе (с разделением на подгруппы).
9. Расписание занятий на заданный день.
10. Перевод числа из двоичной системы в десятичную и обратно.
11. Представление числа в системах счисления: двоичной, восьмеричной, десятичной, шестнадцатеричной.
12. Таймер.
13. Текстовое название введенного численного значения.
14. Напоминалка: создание напоминаний на различные случаи из вашей жизни, например: «Возьми зонтик» (если идет дождь).
15. На выбор студента.

Дополнительно:

1. **Черепашья графика:** проект, в котором создается плоскость из квадратов 20x20. С помощью различных команд вы управляете

черепахой, которая рисует линию на плоскости. Она может двигаться вперед, назад, двигаться влево и вправо, поднимать и опускать перо и т.д.

2. **Реализуйте шифр Цезаря:** шифрование и дешифрование. Ключом является целое число от 1 до 25. Этот ключ сдвигает буквы алфавита (от А до Z). При шифровании каждая буква алфавита заменяется буквой, находящейся на выбранное количество позиций дальше (алфавит закольцовывается). Например, при использовании ключа 2 «НІ» становится «JK», а при использовании ключа 20 «НІ» превращается в «BC».

Вопросы:

1. Что такое жизненный цикл разработки программного обеспечения?
2. Какие модели жизненного цикла разработки ПО вы знаете?
3. Какими способами можно организовать общение с заказчиком?
4. С какой целью выполняется планирование разработки ПО, составление технического задания?
5. Поясните, как выполняется детальный анализ предметной области и принимается окончательное решение о необходимости создания ПО.
6. Перечислите технологии быстрой разработки программного обеспечения.
7. Что такое технология экстремального программирования?
8. Что такое SCRUM-технология?
9. Что такое Kanban-технология?
10. Какие преимущества и недостатки технологий быстрой разработки программного обеспечения?
11. Как организована коллективная работа над проектом при использовании технологий быстрой разработки?
12. Назначение диаграммы вариантов использования.
13. Назначение этапа внедрение и сопровождение программных продуктов.
14. Назовите основные задачи, решаемые на этапе внедрения.
15. Как выполняется процесс устранения ошибок на этапе внедрения.
16. Какие решения могут приниматься при выявлении ошибок на этапе внедрения.
17. Назначение документирования программного обеспечения.