

# **Лекция по СиП 2 семестр**

# Лекция 1 (09.09.2020) тема:

## Гибкие методологии разработки ПО

**Гибкая методология разработки (agile)** — серия подходов к разработке программного обеспечения ориентированный на использование итеративной разработки, динамическое формирование требований и обеспечение их реализации в результате постоянного взаимодействия внутри самоорганизующихся рабочих групп (**дописать**)

Большинство нацелено на минимизацию рисков путем сведения разработки к серии коротких циклов, которые длятся по 2-3 недели

### Итерации:

- Планирование
- Анализ требований
- Проектирование
- Программирование
- Тестирование
- Документирование

Команда как минимум включает:

- Заказчик
- Представитель
- Тестировщик
- Дизайнер
- Разработчик

### Преимущества Agile:

- Agile методы уменьшают объем письменной документации по сравнению с другими методами
- Получение быстрого результата и молниеносная реакция на изменения требований

### Недостатки Agile:

- Отсутствует единое управление требованиями и архитектуры
- Задачи решаются простейшим способом, что приводит к снижению качества продукта и накоплению дефектов

### Основные ценности:

- Люди и взаимодействие важнее процессов и инструментов
- Работающий продукт важнее исчерпывающей документации
- Сотрудничество с заказчиком важнее согласования условий контракта

- Готовность к изменениям важнее следования первоначальному плану

## Основные принципы:

- Наивысшим приоритетом является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения
- Изменения требований приветствуется, даже на поздних сроках разработки
- Работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев
- На протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе
- Над проектом должны работать мотивированные профессионалы. Чтобы работа была сделана, создайте условия обеспечьте поддержку и полностью доверьтесь им.
- Непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды.
- Работающий продукт — основной показатель прогресса
- Инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно
- Постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта
- Простота — искусство минимизации лишней работы — крайне необходима
- Самые лучшие требования, архитектурные и технические решения рождаются самоорганизующихся команд
- Команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль

# Методики Agile:

---

Экстремальное программирование (Авторы — Кент Бек, и еще кто-то)

Название методологии исходит из идеи применить полезные традиционные методы и практики разработки

## **Короткий цикл обратной связи**

- Разработка через тестирование
- Игра в планировании
- Заказчик всегда рядом
- Парное программирование

## **Непрерывный, а не пакетный процесс**

- Непрерывная интеграция
- Рефакторинг
- Частые небольшие релизы

## **Понимание, разделяемое всеми**

- Простота проектирования
- Метафора системы
- Коллективное владение кодом
- Стандарт оформления кода

## **Социальная защищенность программиста**

- 40 часовая рабочая неделя

---

## Feature driven development

FDD включает в себя 5 базовых видов деятельности:

- Разработка общей модели
- Составление списка необходимых функций системы
- Планирование

---

## Canban

По своей сути — trello. Для распределения и деления задач.

# Scrum

Основные роли в методологии скрам.

- Скрам-мастер — проводит совещания и следит за соблюдением всех принципов скрама, разрешают противоречия и защищает команду от отвлекающих факторов. Данная роль не предполагает ничего иного, кроме корректного ведения скрам процесса
- Владелец продукта — представляет интересы конечных пользователей и других заинтересованных в продукте сторон.
- Команда разработки — кросс-функциональная команда разработчиков

## Планирование спринта

- Из бэклога проекта выбираются задачи, обстоятельства по выполнению которых за спринт принимает на себя команда
- На основе выбранных задач создается эклог спринта. Каждая задача оценивается в идеальных человека-часов
- Решение задачи занимает не более 12 задач или одного дня. При необходимости разбивается на подзадачи
- Обсуждается и определяется, каким образом будет реализован этот объем работ
- Продолжительность совещания ограничивается 4-6 часами

## Ежедневное совещание

- Начинается вовремя
- Все могут наблюдать, но только свиньи говорят
- Длится не более 15 минут
- Проводится в одном и том же месте в теч спринта

## Скрам над скрамом

Проводится после ежедневного скрам совещания. Позволяет нескольким скрам командам обсуждать работу, фокусируясь на общих областях и взаимной интеграции.

## Обзор итогов спринта

Проводится в конце спринта

- Команда демонстрирует прирост функциональности продукта всем заинтересованным лицам
- Привлекается максимальное количество зрителей
- Все члены команды участвуют в демонстрации

## Ретроспективное совещание

Что было хорошо? Что надо улучшить?

**Покер планирования** — техника оценки, основанная на достижении договоренности, главным образом используемая для оценки сложности предстоящей работы или относительного объема решаемых задач

## Лекция 2 (23.09.2020) тема: Разработка программного обеспечения

Лекция обзорная, поэтому ее тут не будет.