ТСПП

19.03.2018

Лекция №6

Тема: Экстремальное программирование (XP)

Экстремальное программирование одна из наиболее известных гибких методологий. Основатели: Кэнт Бэк, Уорт Канингем.

Итерационная модель: длительность месяц.

Небольшие команды ~10 чел.

Впервые была применена в 1997-1998 год, при разработке системы по начислению зарплат.

В основе XP лежит 4 основных концепции:

* коммуникация между членами команды и
* обратная связь от заказчика
* простота разработки
* смелость подходов к разработке

Итерационная модель похожа на модель Scrum.

XP предлагает 12 основных приемов (разбитых на 4 группы):

1. **Короткий цикл обратной связи(получение обратной связи от заказчика в** **минимальные сроки)**:

* Игра в планирование
* Разработка через тестирование
* Заказчик на рабочей площадке
* Парное программирование

1. **Непрерывный, а не пакетный процесс разработки:**

* Непрерывная интеграция
* Частый выпуск версий
* Постоянный рефакторинг кода

1. **Понимание проекта всеми участниками:**

* Метафора системы
* Простота разработки
* Коллективное владение кодом
* Стандарты кодирования

1. **Социальная защита программиста:**

* 40-ка часовая раб. неделя

ОСНОВНЫЕ ПРИЕМЫ:

**Игра в планирование**(начало разработки). В данном приеме подразумевается активное взаимодействие заказчика и разработчика. Проводится каждые 3-6 недель, перед началом след. итерации. Позволяет корректировать требования к проекту в соответствии поступившим требованиям.

**Тестирование до начала разработки(разработка через тестирование)**

TDD(Test Driven Development) Написание модульных тестов до создания классов. Процесс разработки завершается при прохождении всех указанных наборов тестов.

**Парное программирование**

Парная работа разработчиков за 1 компьютеров.

Возможно совместная работа менее опытного разработчика с более опытным, или двух с примерно одинаковым уровнем. В результате снижается количество ошибок. Растет качество кода, а следовательно снижается время на тестирование и анализ. Распределяется ответственность между разработчиками за каждый участок кода.

**Постоянный рефакторинг (переработка)** **кода**

Изменение внутреннего кода без явных изменений поведения программы. Используется для дальнейшего удобства использования программистами.

**Простота разработки**

Простота написания кода.

**Коллективное владение кодом**

Выдвигается требование, что за каждый участок кода отвечает как минимум 2 человека.

**Непрерывная интеграция(CI – continue integration)**

Непрерывная инкрементная интеграция.

**Заказчик на рабочей площадке**

Наем человека сведущего в сфере разрабатываемого ПП, отвечающего на возникающие вопросы в процессе разработки

**Быстрый выпуск версий**

Формирования корректных требований от заказчика зависит от представляемого промежуточного продукта, на основе которого корректируются уже сформированные требования. Требования к продукту, выполненные части проекта должны быть подвержены тестированию!

**40-ка часовая рабочая неделя**

5 дней в неделю по 8 часов. Работа команды в одном ритме.

**Стандарты кодирования**

Определение стандартов написания кода и строгого соответствия кода этим требования.

**Метафора системы**

Подход к анализу и проектированию задачи, предполагающий, что мы опираемся на уже завершенные проекты. Ассоциация текущих заданий с ранее выполненными.