# Вступление

# Риски – это большая проблема

Риски:

1. Проваленный срок
2. Отмененный проект
3. Замена системы из-за непригодности в долгосрочном использовании
4. Огромное количество дефектов в готовом продукте
5. Созданный продукт не решает задач бизнеса
6. Бизнес изменился – пока делали пол года продукт рынок поменялся
7. Большое количество задач было с точки зрения программирования, но не бизнеса
8. Люди уходят не выдержав проекта

# Кривая затрат

# Тестирование

Как обычно происходит создание системы?

Мы проектируем систему, создаем ее, тщательно тестируем и отдаем заказчику.

А вот так происходит разработка в XP. Мы начинаем с тестов. Это обязательное правильно.

# Парное программирование

В прямом смысле этого слова. Два программиста за одним компьютером.

Как бороться с сопротивлением:

* Не обязательно явно говорить о том, что с этого дня вы работаете только в парах

Почему я не хочу программировать в паре:

* Я хочу стать героем
* Боюсь продемонстрировать свои недостатки
* Я не уверен в том, что моя идея работает
* У меня плохое настроение
* Я люблю чувство медитации

# Рефакторинг

Рефакторинг есть следствие эволюции системы, ее естественного развития

Сравнить с ремонтом. Можно забыть о подтекающем кране, о сломавшейся форточке, о плитке, которая отходит. Ремонт каждого из этих поломок длиться пол дня. Но в один прекрасный момент кухня дойдет до такого состояния, что в ней невозможно будет находится. И тогда вы отколупаете всю плитку, которая и так на ладан дышит, снимите окна, из которых стекла вылетают. Ремонт будет длиться неделю и при этом на кухне будет невозможно находиться.

# Игра в планирование

# Постоянная интеграция

# Экспресс-совещания

# Простой дизайн

Простой не значит плохой или некачественный.

# Заказчик в команде

# Приемочные тесты

# Метафора системы

# Коллективное владение кодом

Автор кода – вся команда

# Стандарты кодирования

# 40 часовая рабочая неделя