

TDD

От сказки до рабочей фичи

Злата Занина



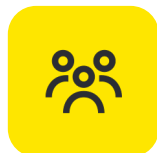


Про меня

- Старший Java-разработчик продуктовой команды Credit cards
- Опыт работы в команде – 2 года
- Опыт разработки – 8 лет



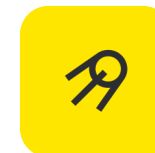
Про команду



3-5 разработчиков
3 тестировщика



Обширная предметная
область и около 80 сервисов в
зоне ответственности



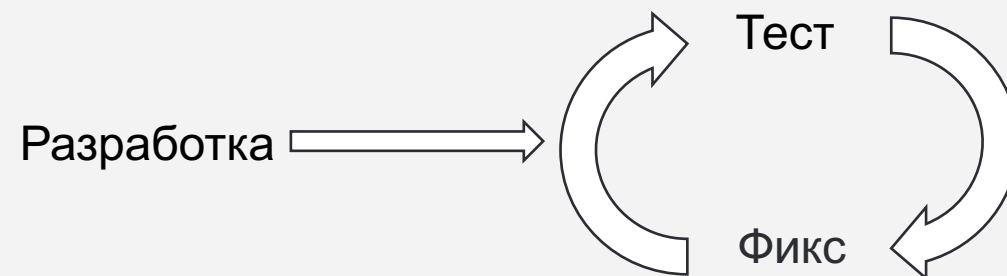
Амбициозные цели на год



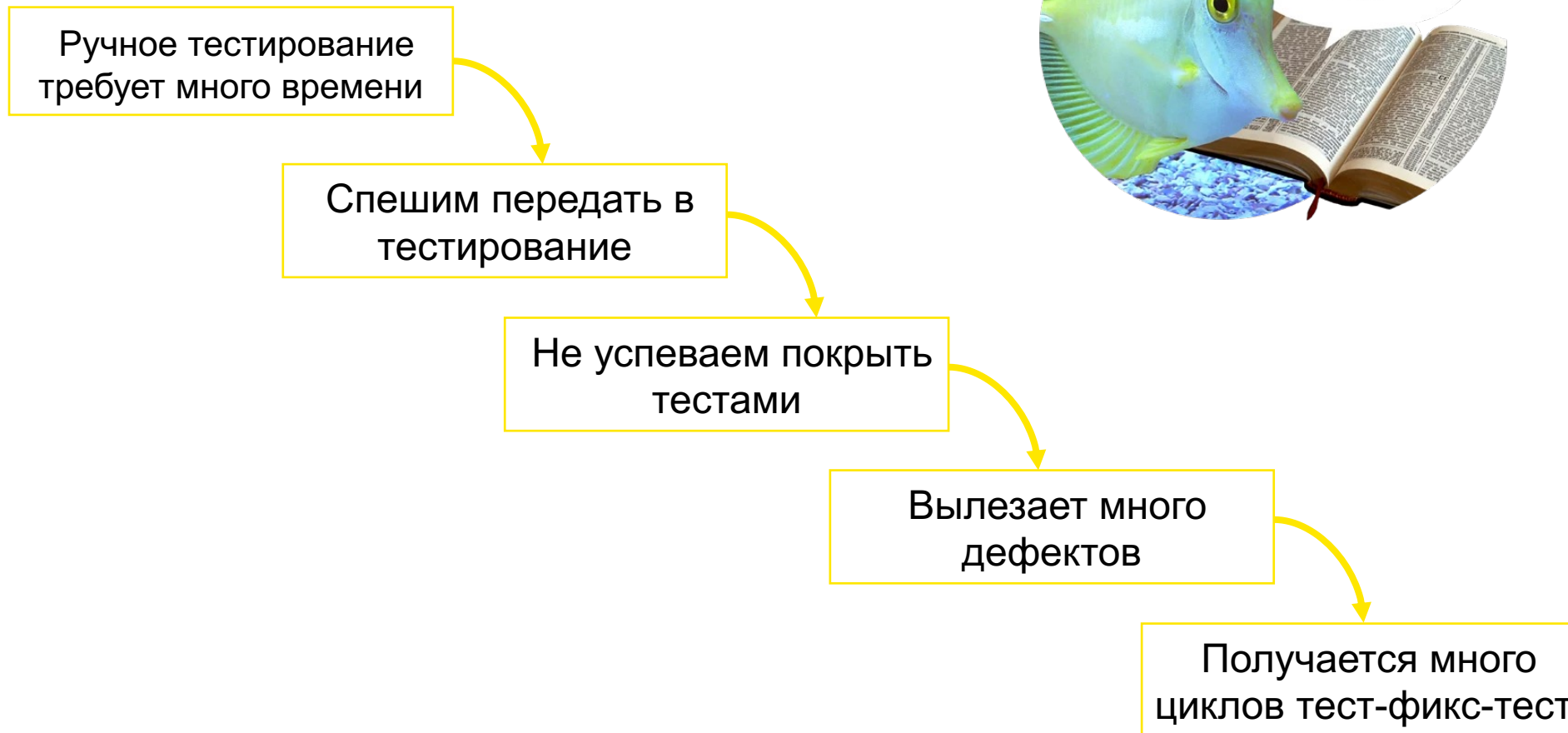
Про разработку

- Двухнедельные спринты
- Ручное тестирование
- ~3-5 задач в бэклоге спринта

**Не хотим много итераций
«разработка – тест – фикс – тест»**



Проблема





Последствия

- Задача не влезает в спринт
- Не выполняем обещания
- Генерируем техдолг
- Код не покрыт тестами –
потом сложнее дорабатывать
и регрессить





Данные – как было

50%

Выполнение целей спринта

40-80%

Покрытие кода тестами

7-10

Итерации доработок по одной задаче



Что можно сделать?

- Увеличить спринт
- Брать в спринт меньше задач
- Увеличить команду
- Нанять «рок-звезд» мира разработки
- Составить матрицы компетенций
- Увеличить масштабы T-shape
- Continuous Delivery & Trunk Based Development



Что мы сделали

- TDD
- Парное программирование
- Tres Amigos



Парное программирование

- Zoom + Share screen
- Группа проектов для обучения в нашем GitLab-репозитории
- IDEA Code with me plugin





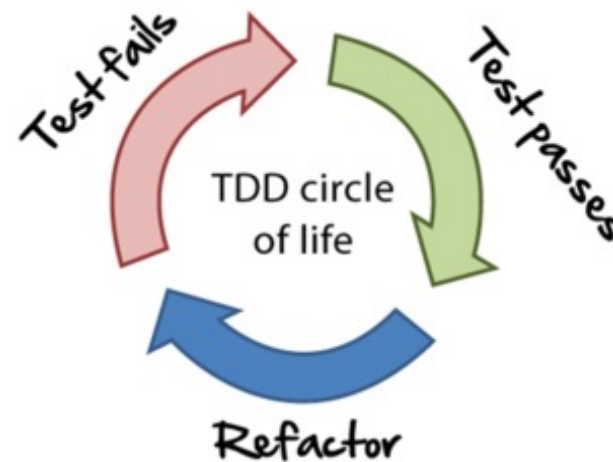
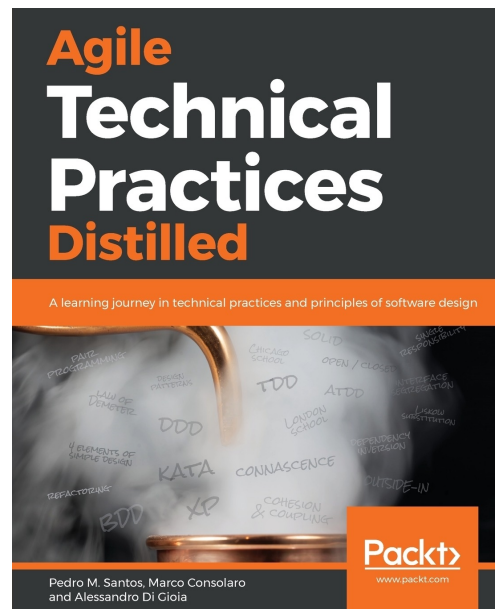
Профит и проблемы

парного программирования

- Делимся экспертизой
 - Повышаем концентрацию
 - Ускоряем разработку
 - Улучшаем качество кода
- Навигатор и драйвер должны хорошо понимать, в чем состоят роли и почему так
 - Требуется схожего уровня инициативности у драйвера и навигатора
 - Повышает уровень стресса

Test Driven Development

- Книга Agile Technical Practices Distilled: чтение и каты по главам
- Пишем парно: один цикл TDD, потом меняемся



Проблемы

- Чем дольше делаем каты, тем меньше интерес
- Сложно перейти от кат к реальным проектам:
 - Тесты писать сложнее, чем для упражнений
 - Рабочим задачам нужно больше времени, чем мы отвели на эксперимент



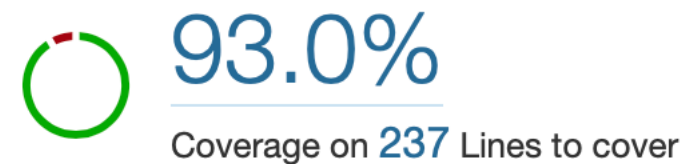
Как боролись с проблемами

- Позвали тестировщиков и аналитиков
- Освоили техники написания юнит-тестов
- Втащили реальные рабочие задачи по чуть-чуть



Профит TDD

- Почти тот же цикл «разработка – тест – фикс – тест», но гораздо быстрее
- Код сразу хорошо и качественно покрыт тестами
- Работа трех амиго улучшает понимание и быстро приближает код к соответствию ФТ





Данные – как стало

71,4%

Выполнение целей спринта

70-93%

Покрытие кода тестами

0-3

Итерации доработок по одной задаче

50%

40-80%

7-10

Рецепт

- Мало экспертизы – научите одного, распространите в парах
- Плохие тесты – соберите команду и пишите вместе
- Скучно с упражнениями – пора переходить на рабочие задачи



Спасибо

Злата Занина

zanina.zlata@yandex.ru

