

Как Scrum-мастер перевернул тестирование в команде с ног на голову



01

Юра Акопян – немного
фокусник

Telegram: [@Yury_Ostr](https://t.me/Yury_Ostr)

02

С 2008 года: автоматизатор,
тестировщик

С 2019 года: немного Scrum-мастер

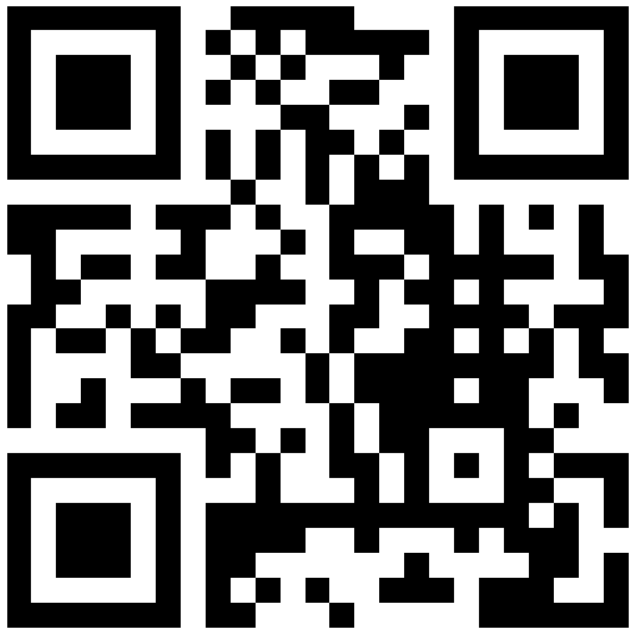
03

Авто-тесты (Java)

04

Помогаю коллегам с помощью
головы, рук
и Java / JS / sh / ...

Какие сложности бывают в командах?



Много багов



Мало тестировщиков



Нет времени тестировать



Длительные релизы



Низкое качество кода



Неожиданный результат



Колодцы внутри команды:
dev VS qa VS BA/FA



Очевидное неочевидно

Какие затруднения есть у вас?

0

Много багов

0

Мало тестировщиков

0

Нет времени тестировать

0

Длительные релизы

0

Низкое качество / культура кода

0

Результат отличается от ожидаемого

0

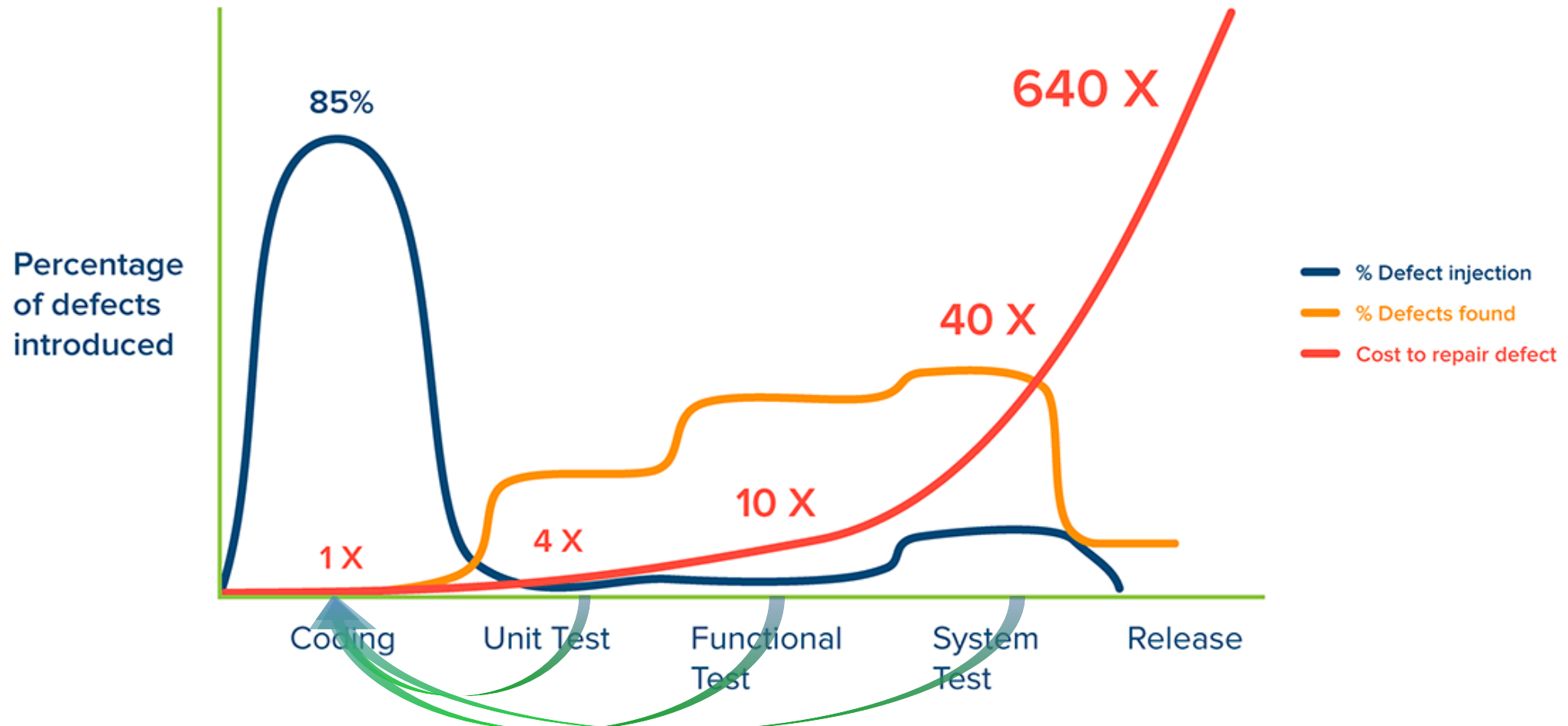
Колодцы внутри команды: dev VS qa
VS BA/FA

0

Очевидное неочевидно

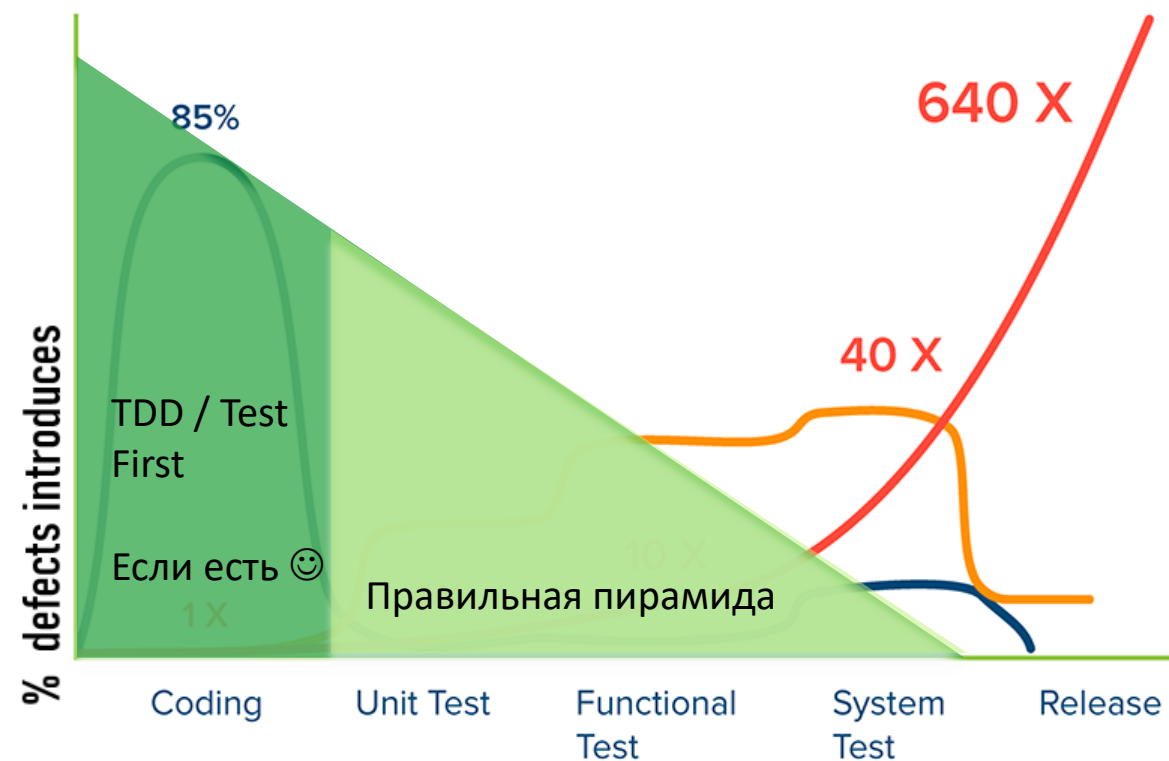
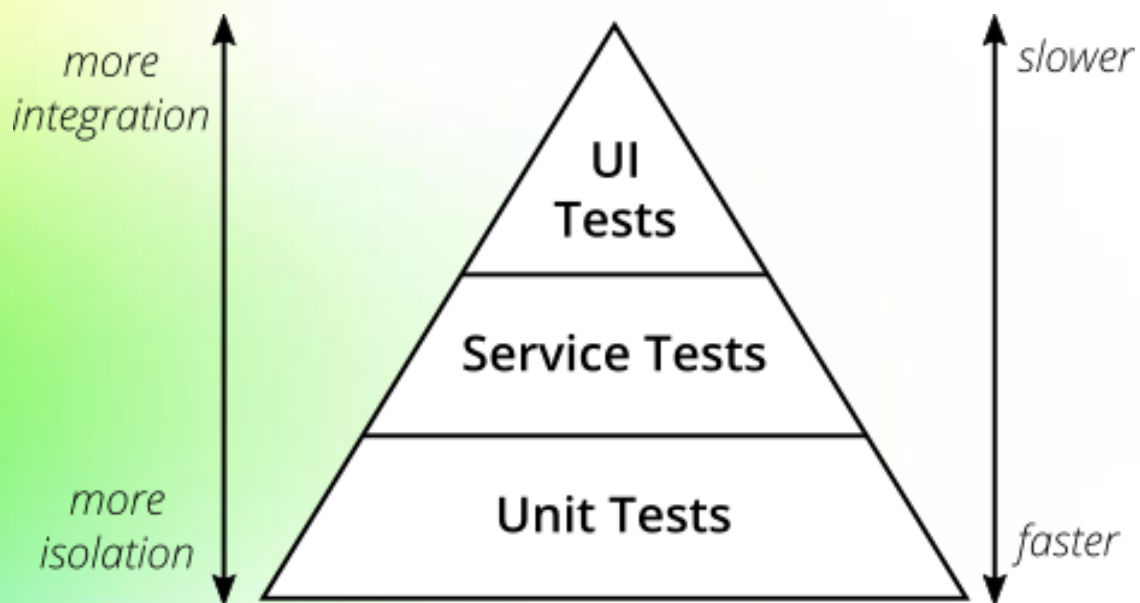


Изучаем теорию



Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality*.

Продолжаем изучение теории



#PYRAMID

Пирамида - реальные ситуации



Лучше потратить 4 часа на юнит тесты и 1 час на регресс, чем 2 дня на регресс



Используем подготовленные «реальные» внешние источники [TestContainers](#)



Проверяем покрытие кода тестами (баги бегут во тьму) не гонитесь за 146% процентами



Применяем экономически выгодное тестирование (с) [Андрей Солнцев](#)

Пирамида - реальные ситуации



Лучше сделать 10 тестов по 20мс, чем
1 тест на 10 секунд с той же логикой



Тесты должны быть постоянно
зеленые (красный и серый это
не зеленый!)



Меняем код (особенно легаси) –
страхуемся тестами

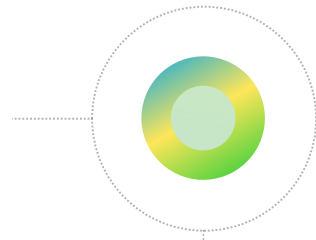


Было 100 e2e тестов ~ 32 часа, ~20% покрытия
(с [moon](#) уложились в 8 часов)

Стало:
~ 10 групп (юнит + интеграционные) * 10 минут
~ 1 группа E2E ~ 10 минут
~ 80% покрытия.

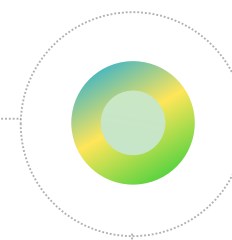
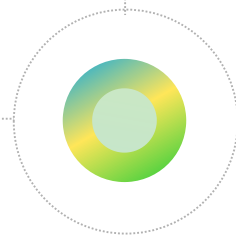
Экстремально развиваем технику

Полезный Code
Review + учимся
писать тесты + по
желанию TDD / ATDD



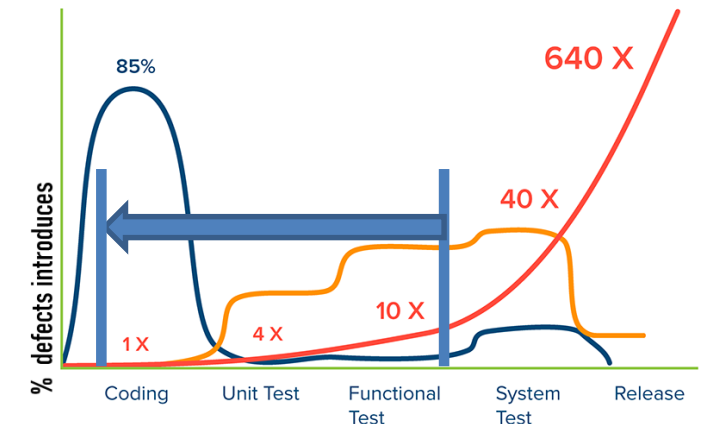
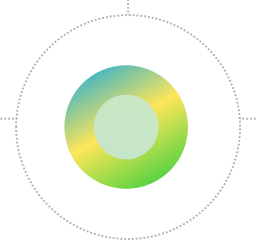
Код ревью – изучаем код

Создаём пирамиду тестов
и находим баги
danke [SebastianDaschner](#)!
thanks [Josh Long](#) !

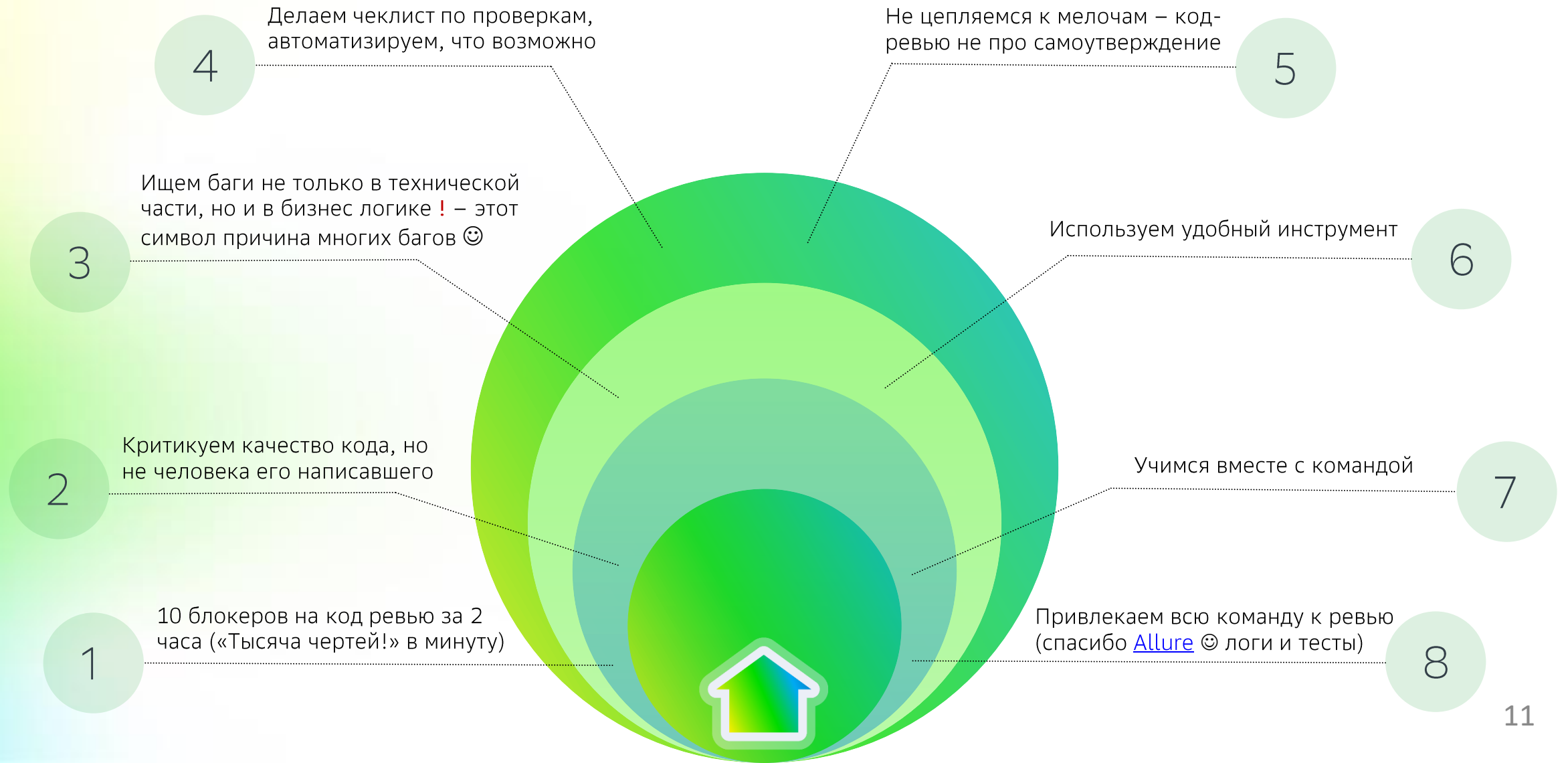


Настраиваем регулярный
запуск тестов

Применяем
подходящие
инструменты



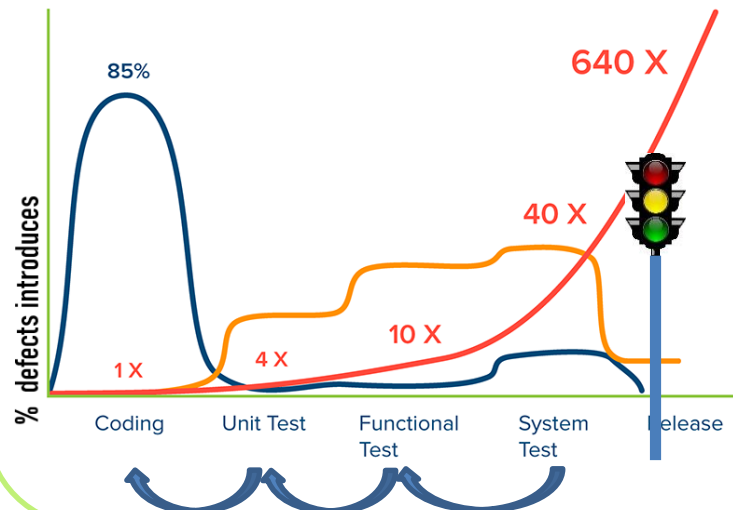
Через некоторое время тренировок...



идеальный DoD



Все работы, которые надо сделать, что бы поставить/отдать продукт



Done / Undone – проверка качества



Единое понимание для всех



Единый набор ТЕХНИЧЕСКИХ работ



Постоянное ужесточение



Undone это нормально



Не сделали «зелёный» DoD – это плохо!



- Не успеваешь? - Не бери больше!

Реальные ситуации Done / Undone

Текущий Done (DoD)



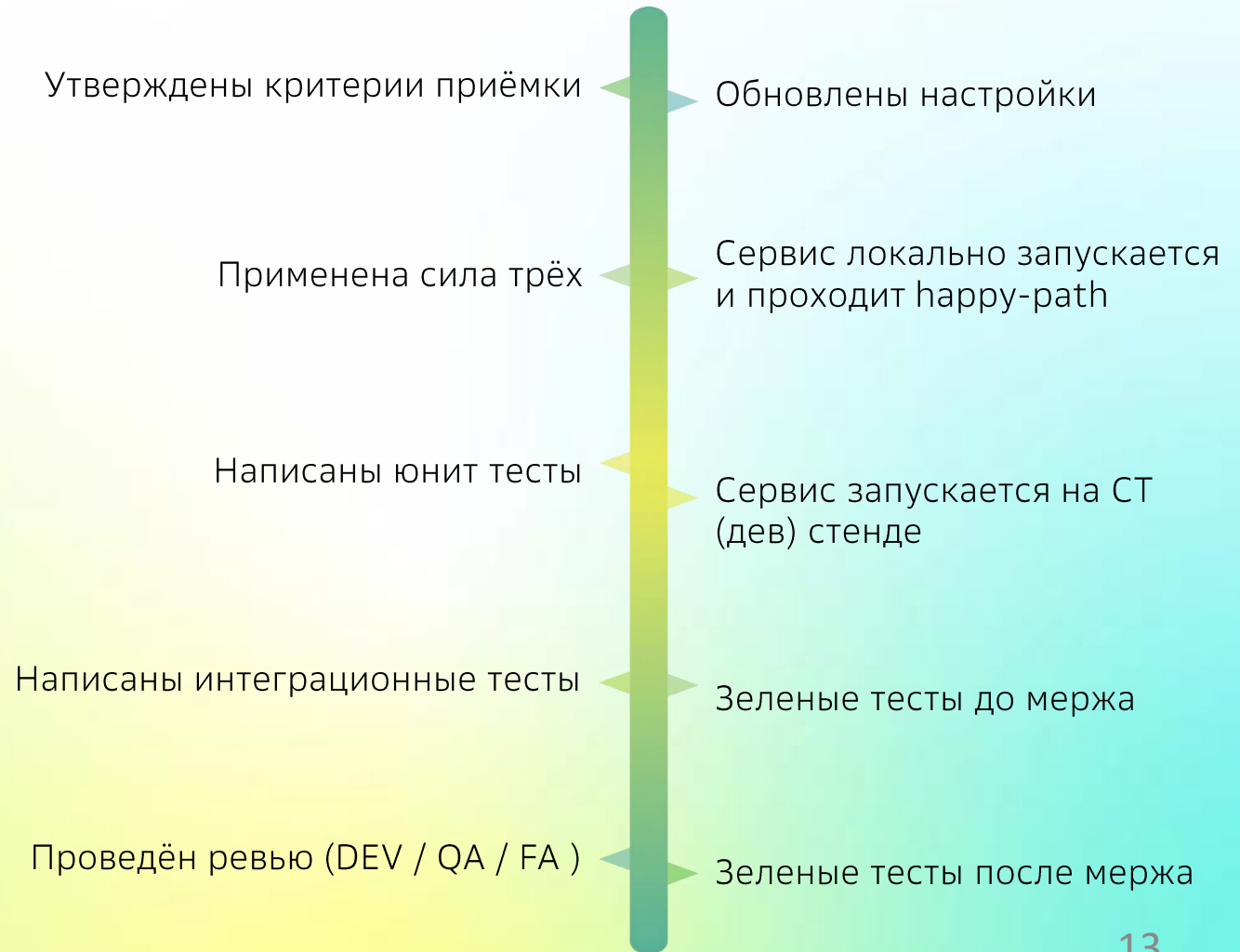
Долгие споры – из чего состоит Done?



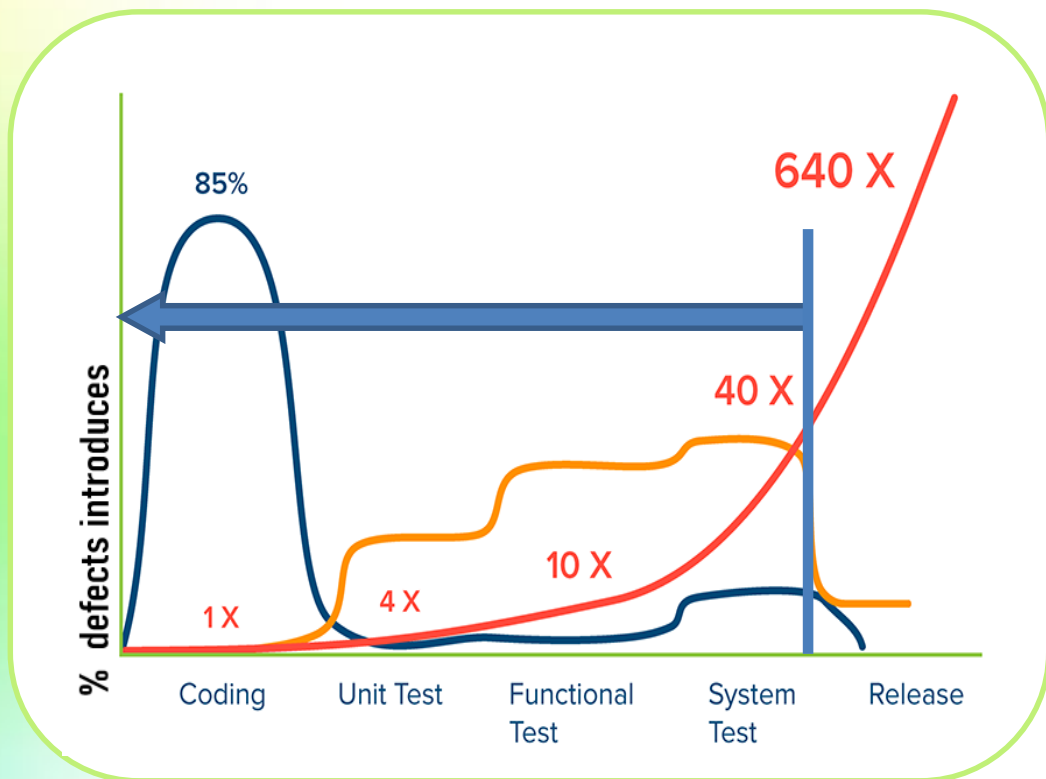
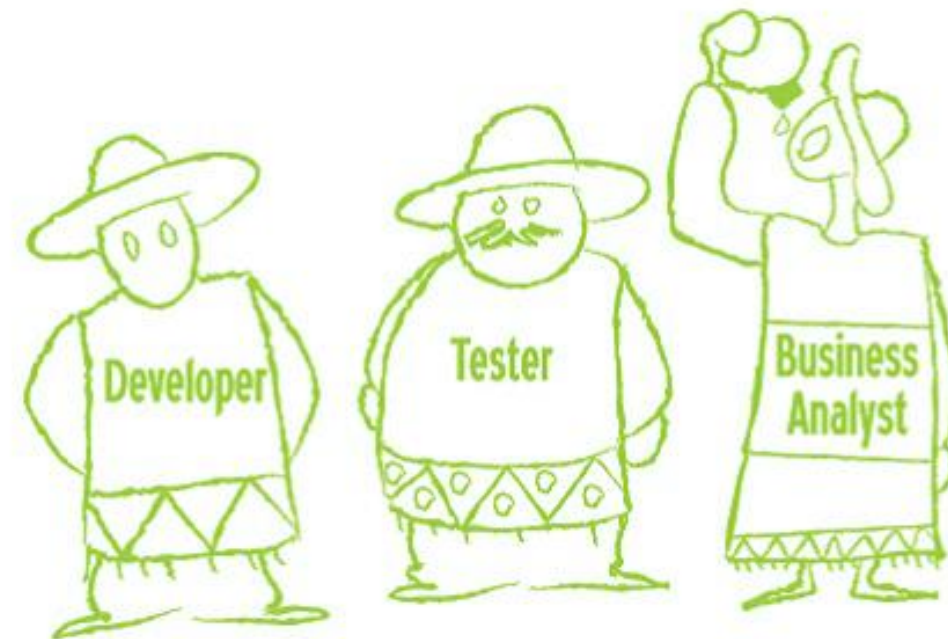
- Ты сделал?
- Да!
- По ДоД?
- Нет ☹



– У меня нет времени делать тесты по DoD! И на сам ДоД нет времени!



Три амиго / Сила трёх



Ограничено по времени

1 стори = 1 встреча

Не «раздуваем» задачу

Начинаем общаться

Выравниваем понимание business value

Где «туман войны»?

Уточняем критерии приёмки + определяем тесты

Три амиго / Сила трёх ! – реальные ситуации



Разработчик не собирает встречу
(или не приходит) :
«вас всех нереально собрать – нет
времени, надо писать код!»



Что будет если: - упадёт база, поле будет
слишком длинное, возникнет неожиданная
ситуация (вес > 1000 кг)



Очевидное неочевидно: «в поле
должно быть указано значение X,Y,Z»
- как это понять?



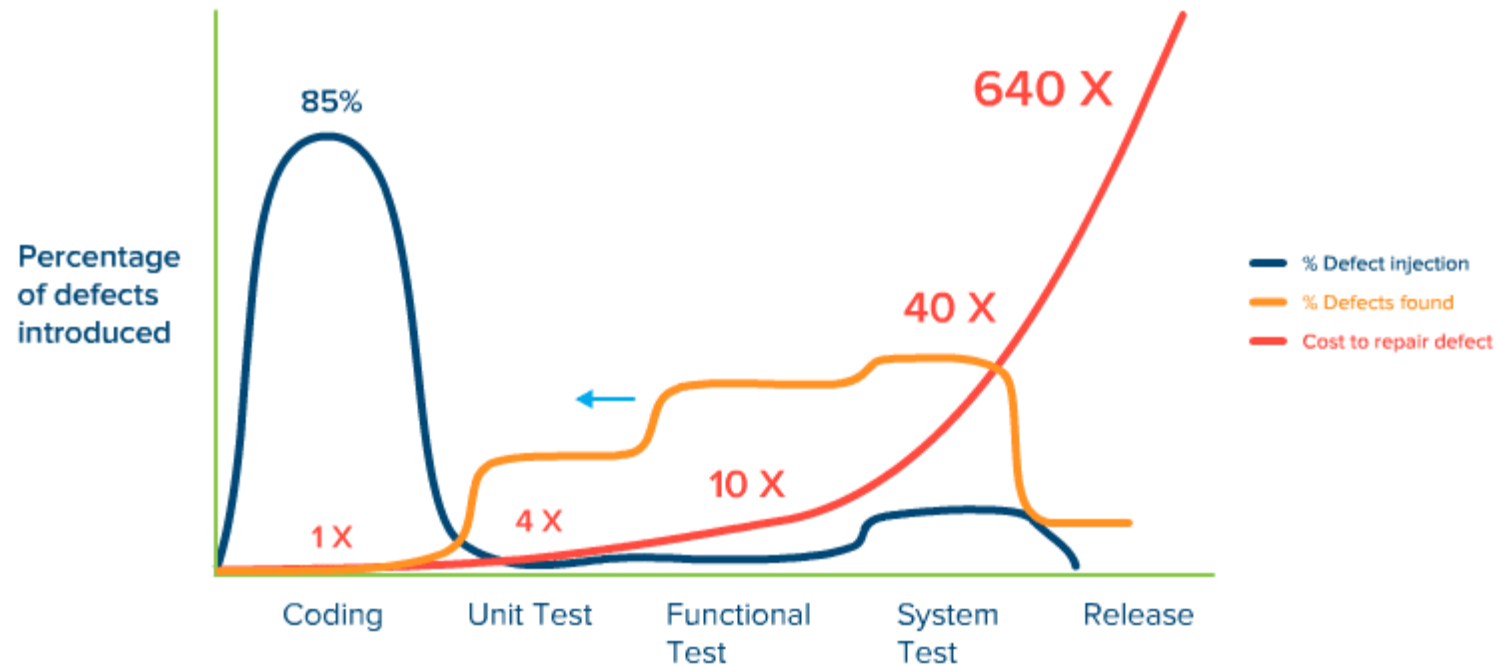
Фиксируем описание тестов до
разработки кода

Заключение – новые командные практики

Цена ошибки

Критерии
приёмки

Три Амиго



Пирамида
Тестирования

TDD
+ CI + ...

Done + UnDone

Jones, Capers. *Applied Software Measurement: Global Analysis of Productivity and Quality*.

Вопросы ?

Титры: целеполагание

| | |
|----------|------------|
| S | Specific |
| M | Measurable |
| A | Achievable |
| R | Relevant |
| T | Time-bound |

| | |
|----------|--------------|
| В | Время |
| О | Оценка |
| Д | Достижимость |
| К | Конкретность |
| А | Адекватность |

Титры: о чём хотелось бы сказать, но....

Q1

Страховочная сеть
+ тех.качество

Q1

CI/CD + TDD + XP + ...

Q2

Внешнее качество (AC)

Q2

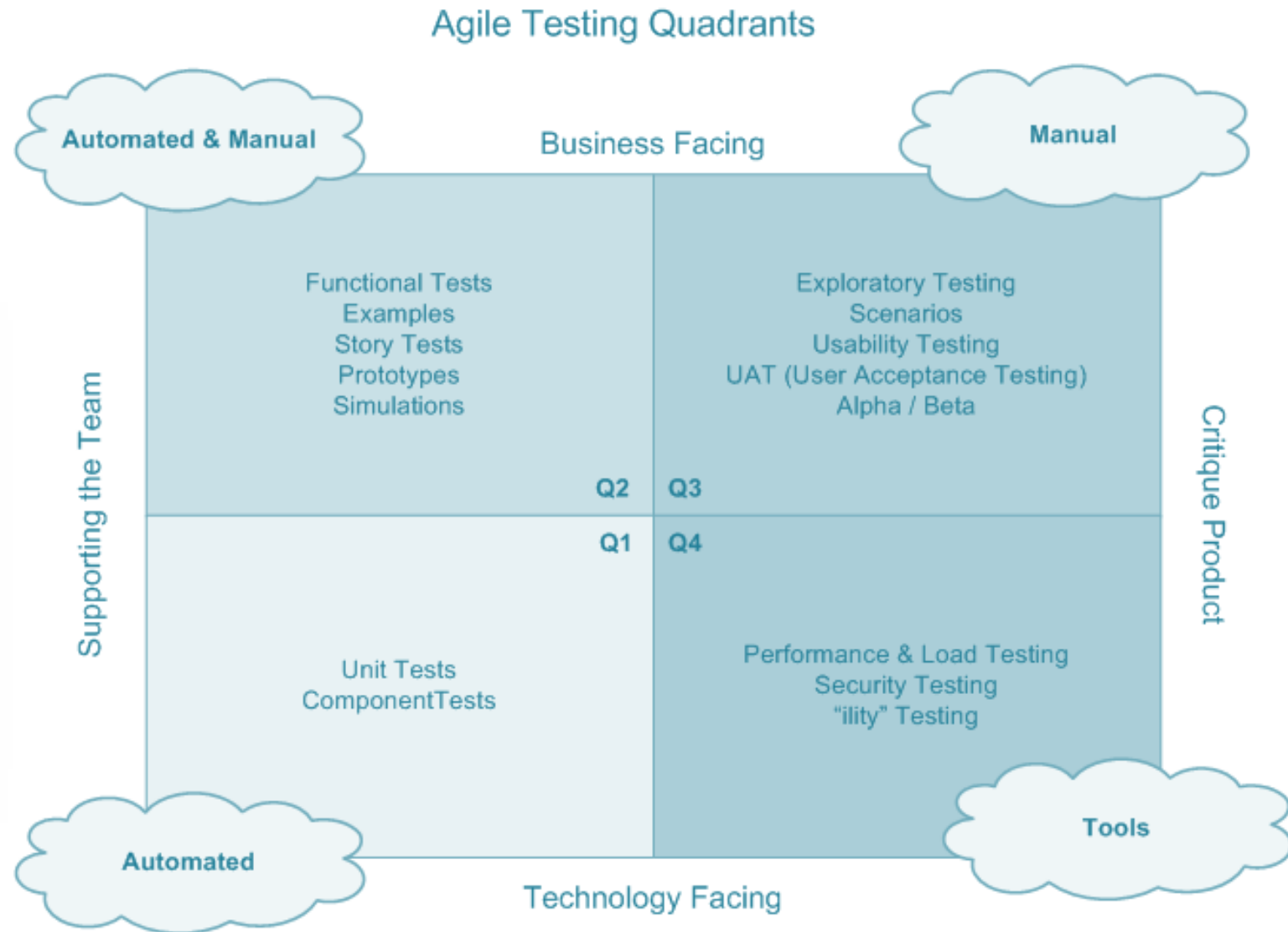
Декомпозиция историй

Q3

Исследуем и критикуем

Q4

Не забываем про ility тесты



Титры: список полезной литературы

- <http://www.agilemodeling.com/essays/costOfChange.htm>
- <https://martinfowler.com/articles/practical-test-pyramid.html>
- <https://less.works/less/framework/definition-of-done.html>
- <https://www.agilealliance.org/glossary/three-amigos/>
- <http://kanban.club/all/otlichiya-definition-of-done-ot-acceptance-criteria/>
- <https://www.parasoft.com/what-is-shift-left-testing/> +
<https://learning.oreilly.com/library/view/applied-software-measurement/9780071502443>
- <https://lisacrispin.com/downloads/AdpTestPlanning.pdf>
- [TestContainers](#) + [Андрей Солнцев](#) + [moon](#) + [Allure](#)

The End 😊

Telegram: [@Yury_Ostr](https://t.me/Yury_Ostr)