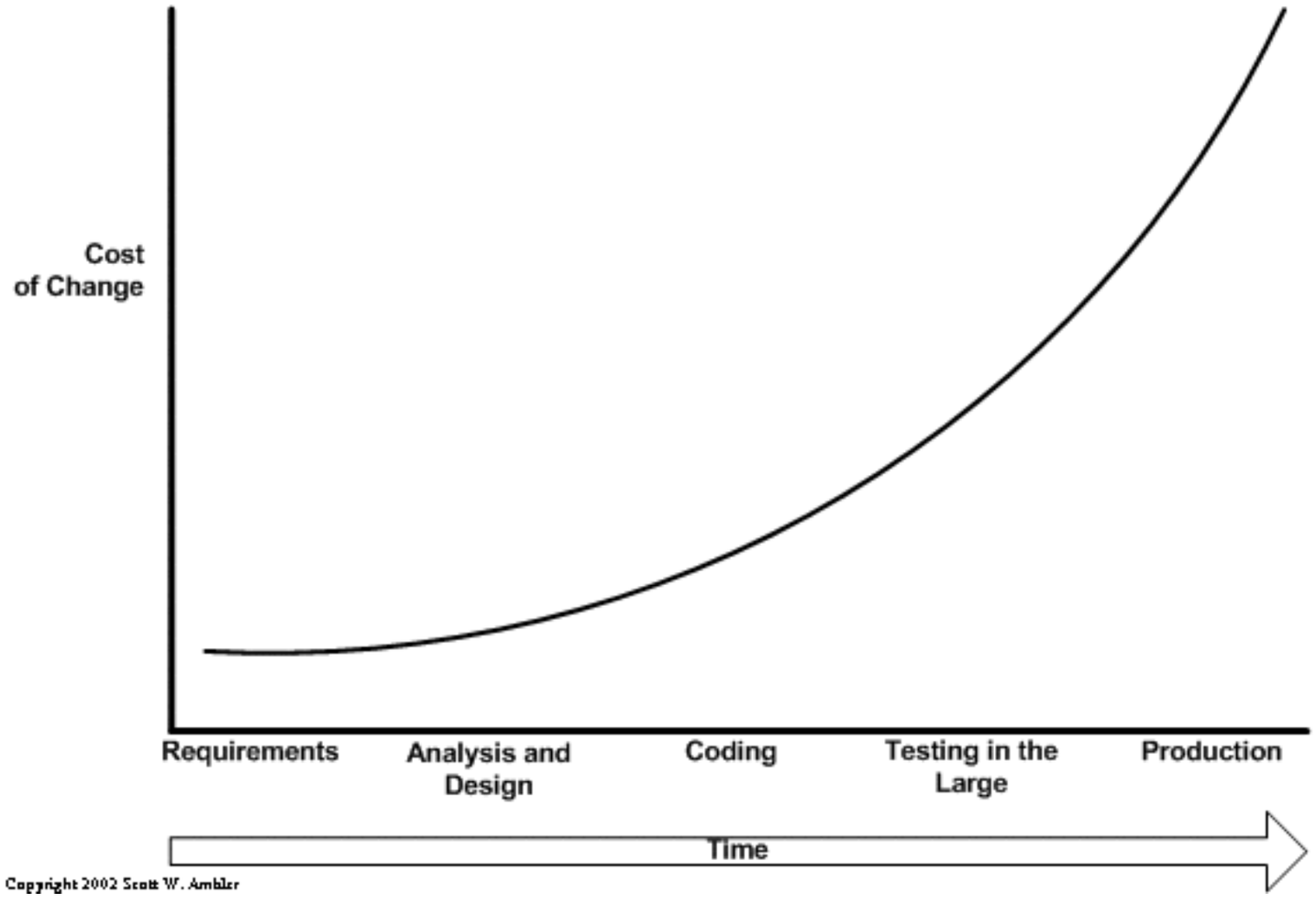

django

Test Driven Development

Или когда писать тесты?

В чем плюсы?

- Помогает заранее продумать архитектуру
- Тесты изначально показывают корректность работы приложения
- Позволяет быстро находить ошибки



В чем минусы?

- Медленно
- Требует большего навыка от программиста
- При изменениях стоимость сильно увеличивается

Когда следует использовать?

- Ведущий разработчик обозначает основной функционал (и API), который должен быть реализован остальной командой
- Проект достаточно прост, чтобы писать тесты вначале

Behavior Driven Development

- Позволяет участвовать в тестировании всем участникам команды

- Отлично подходит для agile-методологий

- Тесты представляю собой естественный язык, структурированный специальным образом - https://github.com/cucumber/cucumber/wiki/Gherkin

- http://pythonhosted.org/behave/tutorial.html#features

Feature: Fight or flight
  In order to increase the ninja survival rate,
  As a ninja commander
  I want my ninjas to decide whether to take on an opponent based on their skill levels

  Scenario: Weaker opponent
    Given the ninja has a third level black-belt
     When attacked by a samurai
     Then the ninja should engage the opponent

Scenario: Stronger opponent
  Given the ninja has a third level black-belt
   When attacked by Chuck Norris
   Then the ninja should run for his life

Зачем?

- Не все могу писать тесты, но многие хотят

- Понимание проекта у разных участников - разное, иногда крайне важно работать с каждым из них

Когда хорошо

- Когда вся ваша команда - может справится с такой задачей, она требует определенной сноровки и навыков

Когда плохо

- Если вы работаете с трудными заказчиками
- Если такая задача будет слишком сложна для ваших коллег

Тестирование JS

- https://github.com/skoczen/polytester
- https://mochajs.org/
- http://jasmine.github.io/edge/introduction.html