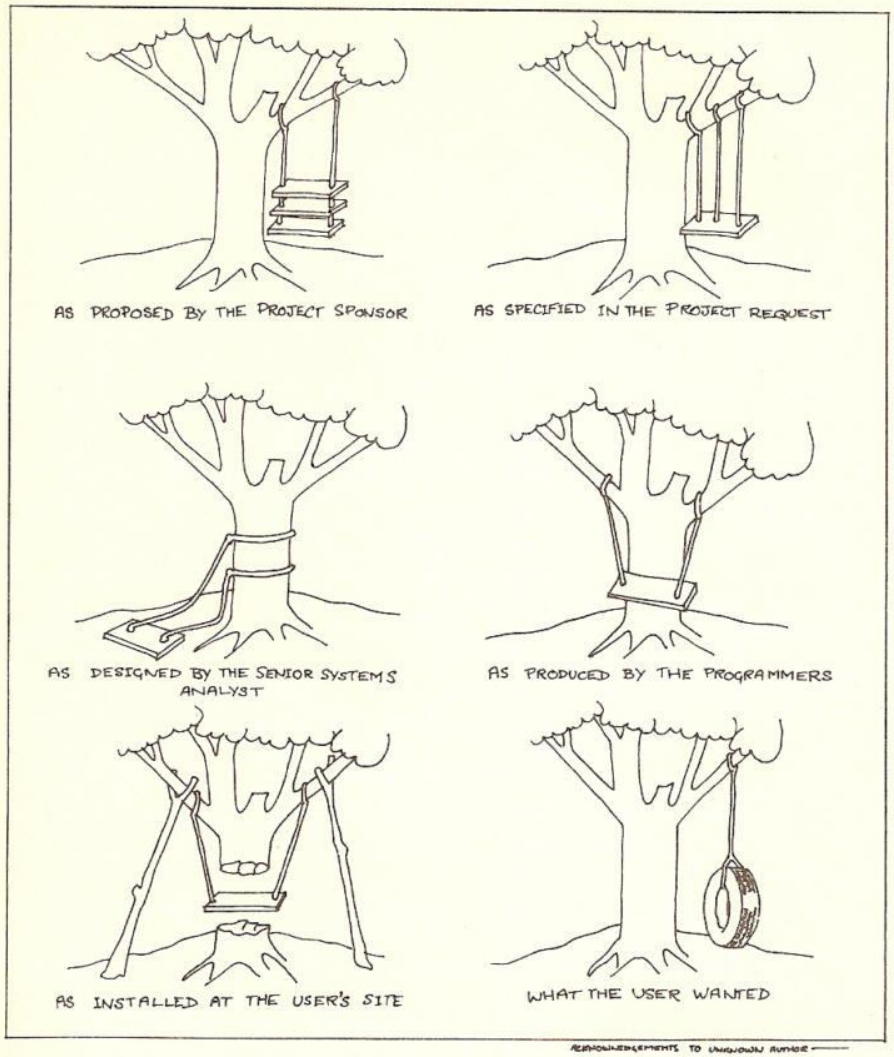
Разработка ПО через тестирование поведения средствами Cucumber

Динерштейн Леонид, Флэтсофт

56% всех ошибок — результат неверного понимания или объяснения требований.

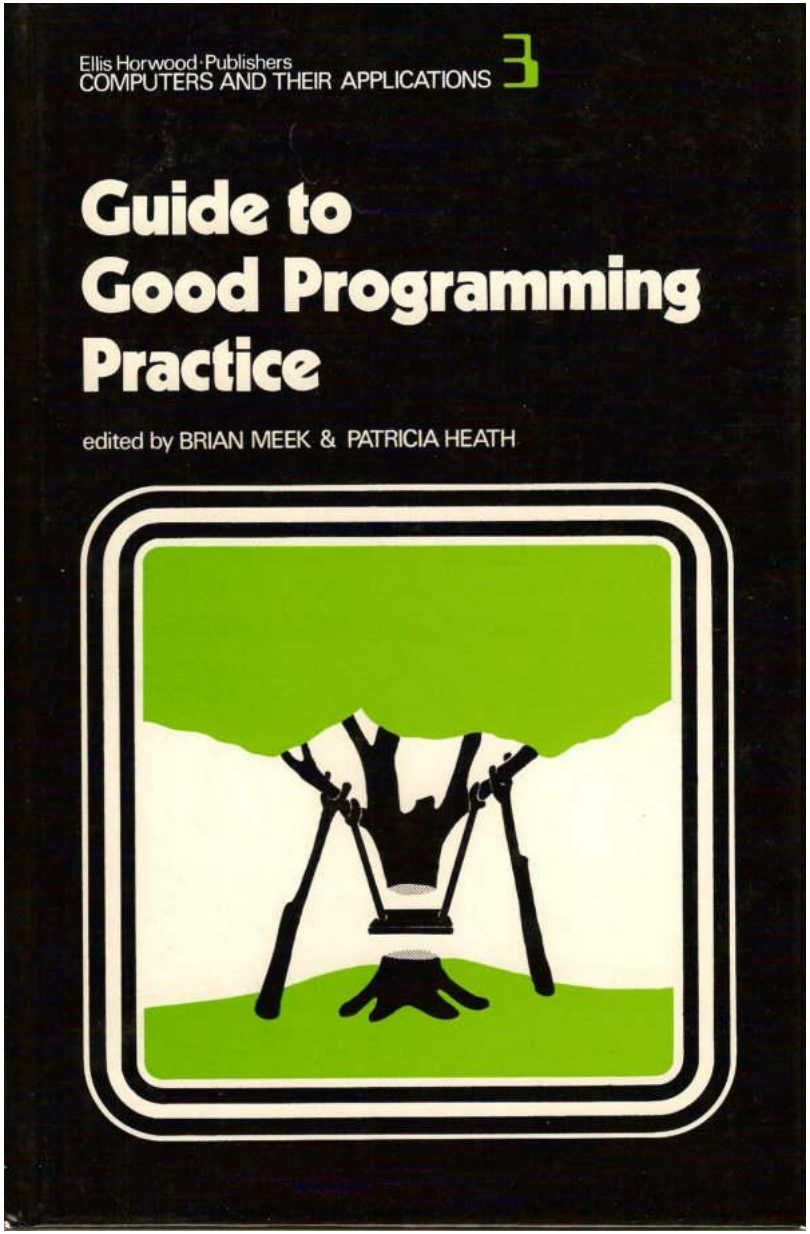
(CHAOS Report)



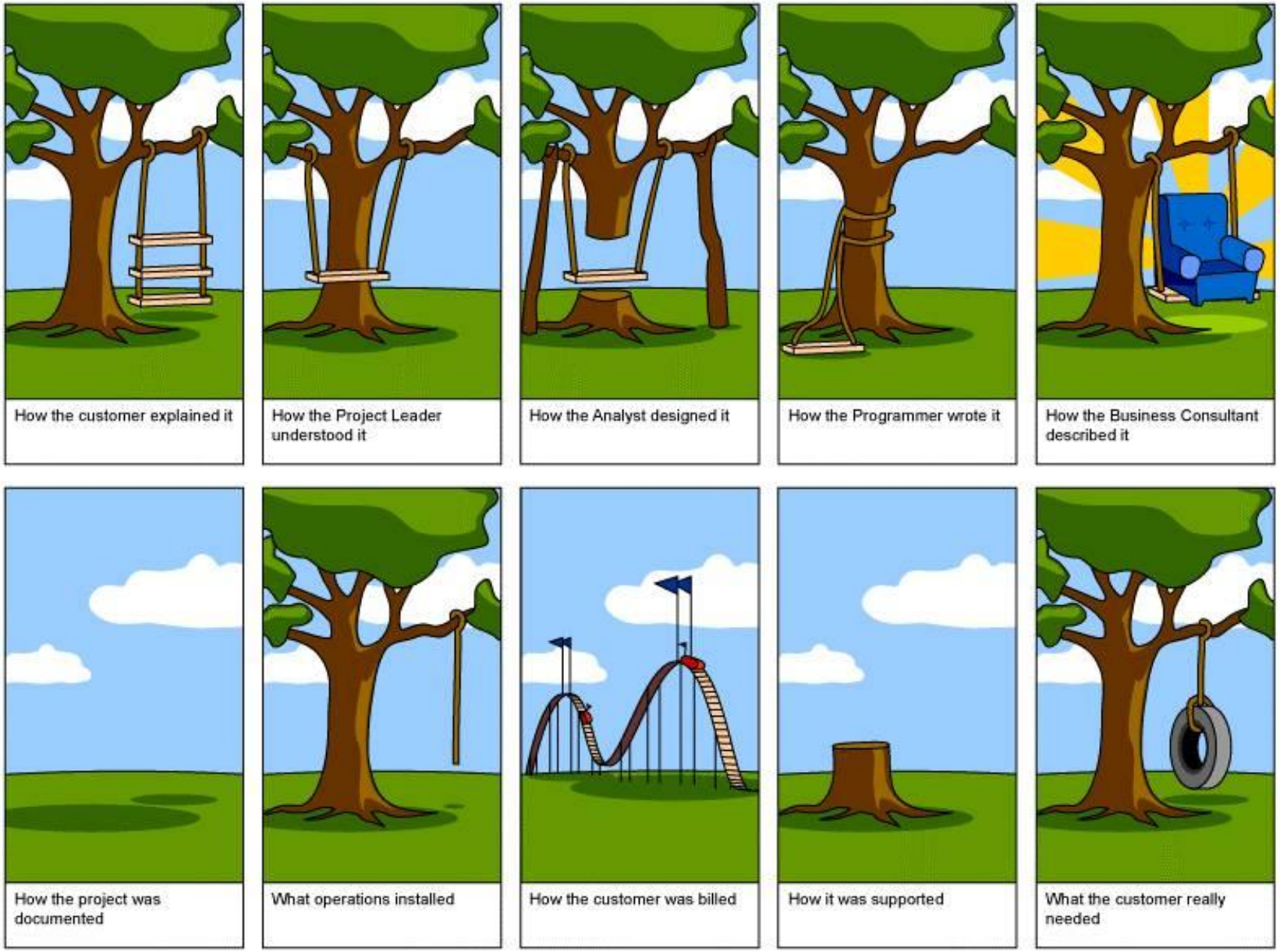
From the *University of London Computer Centre Newsletter* No. 53, March 1973

1973 г.

SQA KAZAN 2011 DAYS



1980 г.


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Наше время

SQA KAZAN 2011 DAYS

1970 г. — сегодняшний день

... который создал новые языки для разных людей, из-за чего они перестали понимать друг друга, не могли продолжать строительство ...

SQA KAZAN 2011 DAYS

1. Средство для автоматизированного тестирования
2. Позволяет описывать поведение системы на естественном языке
3. Является основным инструментом в Behaviour Driven Development (BDD)

SQA KAZAN 2011 DAYS

Cucumber

behaviour driven development with elegance and joy

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)

SQA KAZAN 2011 DAYS

Cucumber

behaviour driven development with elegance and joy

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования

SQA KAZAN 2011 DAYS

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования
3. Запустите тесты и убедитесь, что они не проходит

SQA KAZAN 2011 DAYS

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования
3. Запустите тесты и убедитесь, что они не проходит
4. Напишите код, который реализует поведение, описанное в тестах

SQA KAZAN 2011 DAYS

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования
3. Запустите тесты и убедитесь, что они не проходит
4. Напишите код, который реализует поведение, описанное в тестах
5. Запустите тесты снова и убедитесь, что некоторые тесты начали проходить

SQA KAZAN 2011 DAYS

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования
3. Запустите тесты и убедитесь, что они не проходит
4. Напишите код, который реализует поведение, описанное в тестах
5. Запустите тесты снова и убедитесь, что некоторые тесты начали проходить
6. Повторите 2-5 шаги, пока все тесты не начнут проходить

SQA KAZAN 2011 DAYS

1. Опишите поведение системы на естественном языке (Напишите сценарий поведения)
2. Опишите шаги сценария на языке программирования
3. Запустите тесты и убедитесь, что они не проходит
4. Напишите код, который реализует поведение, описанное в тестах
5. Запустите тесты снова и убедитесь, что некоторые тесты начали проходить
6. Повторите 2-5 шаги, пока все тесты не начнут проходить
7. Повторите 1-6 шаги, пока не закончатся деньги у заказчика

40 естественных языков, включая русский

SQA KAZAN 2011 DAYS

Функционал

Опишите поведение системы на естественном языке

 # language: ru
Функционал: Сложение чисел
Чтобы не складывать в уме
Все, у кого с этим туго
Хотят автоматическое сложение целых чисел

Сценарий: Сложение двух целых чисел
Допустим я ввожу число 50
И затем ввожу число 70
Если я нажимаю "+"
То результатом должно быть число 120

SQA KAZAN 2011 DAYS

Функционал

Опишите шаги сценария на языке программирования

```
Допустим /ввожу число (\d+)/ do |число|
  calc.push число.to_i
end

Если /нажимаю "(.*)"/ do |операция|
  calc.send операция
End

То /результатом должно быть число (\d+)/ do |результат|
  calc.result.should == результат.to_f
End
```

SQA KAZAN 2011 DAYS

Gherkin

"en":

name: English
native: English
feature: Feature
background: Background
scenario: Scenario
scenario_outline: Scenario Outline|Scenario Template
examples: Examples|Scenarios
given: "*|*Given*"
when: *"*|When"*
then: *"*|Then"*
and: *"*|And"*
but: *"*|But"*

SQA KAZAN 2011 DAYS

Gherkin

"ru":

name: Russian
native: русский
feature: Функция|Функционал|Свойство
background: Предыстория|Контекст
scenario: Сценарий
scenario_outline: Структура сценария
examples: Примеры
given: *"*|Допустим|Дано|Пусть"*
when: *"*|Если|Когда"*
then: *"*|То|Тогда"*
and: *"*|И|К тому же"*
but: *"*|Но|А"*

SQA KAZAN 2011 DAYS

Формат

Feature: Title
  In order to [Business Value]
  As a [Role]
  I want to [Some action]

Scenario: Title
  Given [Context]
  When [Action]
  Then [Outcome]

SQA KAZAN 2011 DAYS

Формат

Scenario: Title

Given *[Context]*
And *[Context]*
When *[Action]*
And *[Action]*
Then *[Outcome]*
But *[Outcome]*

SQA KAZAN 2011 DAYS

features/user/manage_account/**update_account.feature**

Feature: Update account information

Background:
Given I am an authenticated user
And I am on the edit account page

Scenario: Update user information successfully
When I fill in "Full name" with "Super Man"
And fill in "Email" with "super@example.com"
And fill in "Password" with "654321"
And fill in "Password confirmation" with "654321"
And fill in "Current password" with "123456"
And press "Update"
Then I should see "You updated your account successfully."
And I should be on the home page

SQA KAZAN 2011 DAYS

features/step_definitions/**authentication_steps.rb**

« Given I am an authenticated user»

```
Given /^I am an authenticated user$/ do
  Given %Q{a confirmed user: "me" exists with id: "1", email: "me@example.com",
password: 123456, full_name: "Current User"}
  When %Q{I go to the sign in page}
  When %Q{fill in "Email" with "me@example.com"}
  When %Q{fill in "Password" with "123456"}
  When %Q{press "Sign in"}
end
```

SQA KAZAN 2011 DAYS

Feature: Crop avatar

Background:

Given a user "u1" exists with username: "John Doe", password: "secret", password_confirmation: "secret", email: "user@test.test"

And user with username: "John Doe" has an avatar: "spec/fixtures/files/avatar.png"

And I am signed in as a user using email: "user@test.test", password: "secret"

Scenario: Cropping user avatar via API

When I have query values start_x: "0", start_y: "10", end_x: "100", end_y: "60" namespaced with "coordinates"

And I send "POST" request to "/users/u1/crop_avatar.json"

Then response should be "success"

And user with username: "John Doe" should have avatar with width: "100" and height: "50"

And response should have message: "image.notices.resized"

SQA KAZAN 2011 DAYS

Зачем нужен **заказчику**?

Зачем нужен **менеджеру**?

Зачем нужен **разработчику**?

SQA KAZAN 2011 DAYS

Заказчик

1. Разговор на одном языке
2. Четкие требования
3. Метрика
4. Вовлеченность в работу

SQA KAZAN 2011 DAYS

Менеджер

1. Разговор на одном языке
2. Уверенность
3. Метрика

SQA KAZAN 2011 DAYS

Разработчик

1. Разговор на одном языке
2. Четкие требования
3. Уверенный рефакторинг

SQA KAZAN 2011 DAYS

Ошибки и заблуждения

1. Хлопотно (я могу тестировать Unit test, Rspec …)
2. Требования быстро меняются
3. Оформление часто меняется
4. Описывать поведение должен заказчик
5. Cucumber == BDD
6. Вы действительно используете Cucumber. (Chicken test)

SQA KAZAN 2011 DAYS

Плагины

Pickle
Capybara
Webmock
VCR
Spork
Parallel
Testjour
….

SQA KAZAN 2011 DAYS

1. Одними из самых неприятных ошибок, являются ошибки неверной реализации требований или даже отсутствие должной функциональности. Сложность этих ошибок заключается в том, что только заказчик может найти их.
2. Разработка программ через тестирование поведения (BDD) является продолжением идеи TDD
3. Главное отличие BDD и TDD заключается в том, что тестируется поведение системы, а не внутренне устройство классов и код.
4. Главной особенностью Cucumber является возможность описания поведения системы на естественном языке.

Вопросы?