Freie Universität Berlin

Systemanalyse und -design, Behaviour Driven Development

Steve Dierker
dierker.steve@fu-berlin.de

CSW
Corporate Semantic Web

Prof. Dr. Adrian Paschke
Email: paschke@inf.fu-berlin.de
Arbeitsgruppe - Corporate Semantic Web
Fachbereich Mathematik und Informatik, Institut für Informatik
Freie Universität Berlin
http://www.inf.fu-berlin.de/groups/ag-csw

Gliederung

Freie Universität Berlin

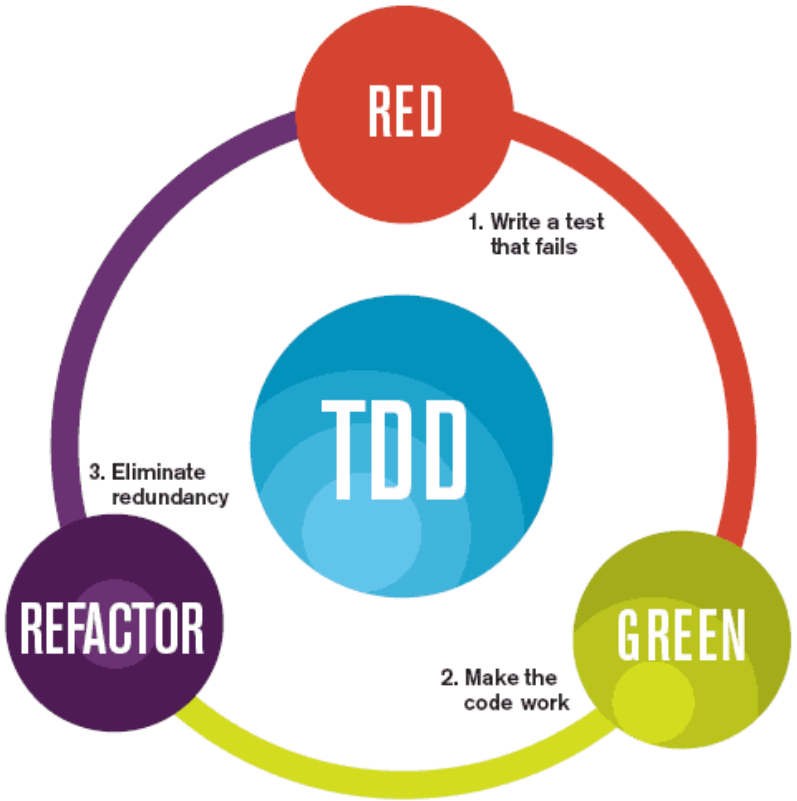
- Was ist TDD?
  - Begriffsklärung
  - Workflow
  - Vorteile
  - Probleme
- Was ist BDD?
  - Workflow
  - Demo
  - Vergleich zu TDD
- Andere Sprachen, andere Tools

Was ist TDD? - Begriffsklärung

Freie Universität Berlin

- TDD heisst Test Driven Development
- began 1998 mit Extreme Programming
- ’entdeckt’ durch Kent Beck
- es ist ein Test-First Softwareprozess
- es gibt auch Test-Last Prozesse

Was ist TDD? - Workflow



Was ist TDD? - Workflow

Freie Universität Berlin

- ein Zyklus sollte nicht länger als 15 Minuten dauern
- Debuggen schneller als neu implementieren?
- Features und Tests nicht zu groß wählen
- nur ein Feature pro Test
- Refactor!

Was ist TDD? - Vorteile

Freie Universität Berlin

- es wird sich ein Emergentes Design bilden
- kompakt und minimalistisch
- kann sich dynamisch den Anforderungen anpassen
- ermöglicht sofortiges Feedback
- immer begrenzt lauffähige Software

Was ist TDD? - Probleme

Freie Universität Berlin

- kein White-Box-Testing!
- keine Hinweise mit welchen Tests anzufangen ist
- was soll ueberhaupt getest werden?
- ersetz NICHT einen ausführlichen Testzyklus am Ende der Entwicklung
- Anforderungen müssen von Programmierer_innen in Test übetragen werden
- kein Einheitliches Namensschema für Tests

Was ist TDD? - Probleme

Listing 1: Ruby UnitTest

```
1  class CalculatorTest < Test::Unit::TestCase
2
3    def test_addition
4      calculator = Calculator.new
5      calculator.push 3
6      calculator.push 4
7      assert_equal(calculator.add, 7)
8    end
9
10   def test_division
11     calculator = Calculator.new
12     calculator.push 3
13     calculator.push 2
14     assert_equal(calculator.divide, 1.5)
15   end
```

Was ist TDD? - Schlussfolgerung

Freie Universität Berlin

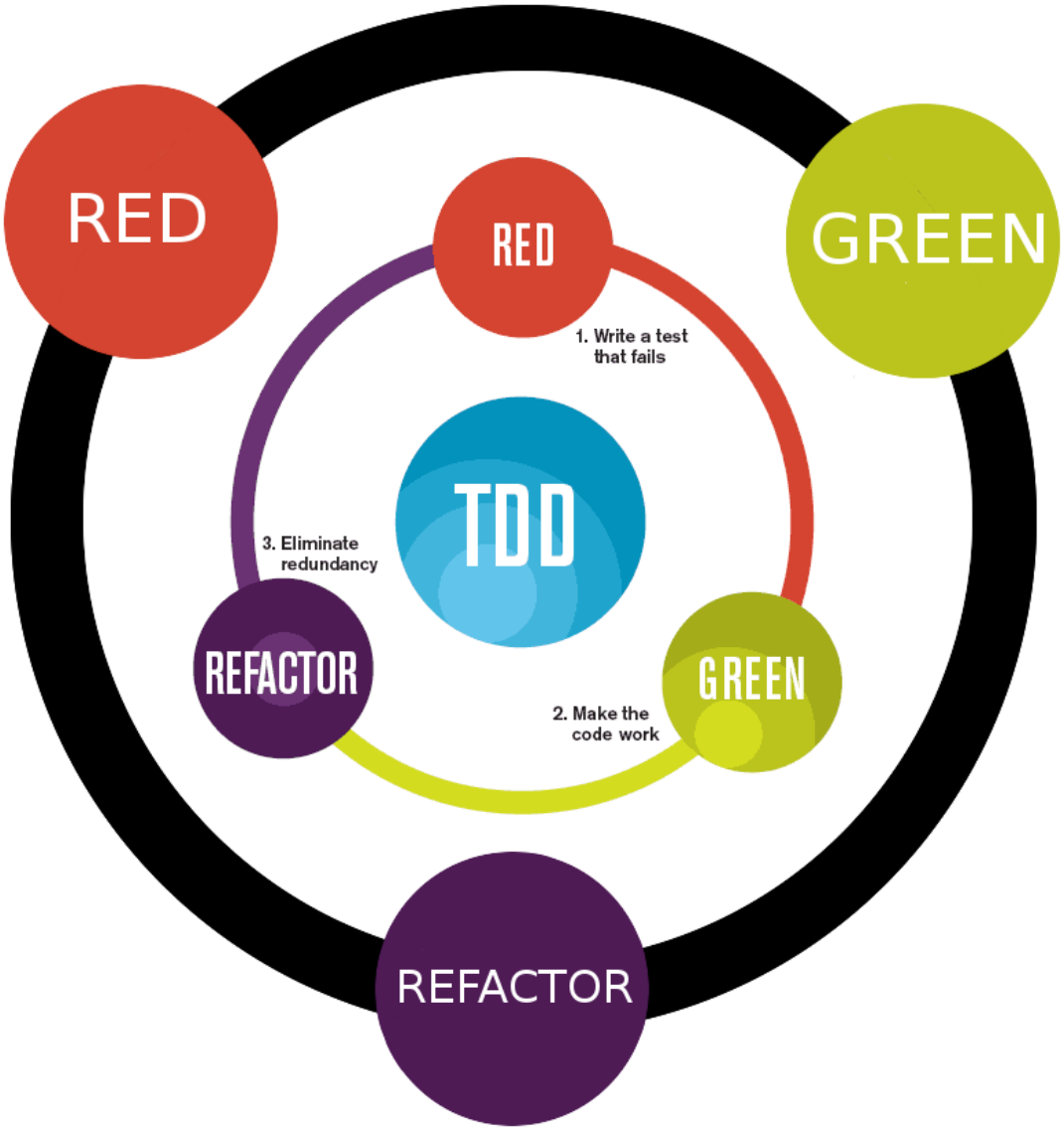
- Gute Idee, schwer zu vermitteln und zu meistern

Was ist BDD? - Workflow

Freie Universität Berlin

- Erweiterung von TDD
- Stakeholder und Entwickler_innen entwickeln Tests gemeinsam
  $\rightarrow$ *Gherkin*
- Stakeholder und Enwickler_innen benutzen die selben Tools
- ein weiterer Zyklus wird hinzugefügt
- But let's dive in!

Was ist BDD? - Workflow



Listing 2: Cucumber Gherkin

```
1  # language: en
2  Feature: Addition
3     In order to avoid silly mistakes
4     As a math idiot
5     I want to be told the sum of two numbers
6
7     Scenario Outline: Add two numbers
8        Given I have entered <input_1> into the calculator
9        And I have entered <input_2> into the calculator
10       When I press <button>
11       Then the result should be <output> on the screen
12
13    Examples:
14       | input_1 | input_2 | button | output |
15       | 20 | 30 | add | 50 |
16       | 2 | 5 | add | 7 |
17       | 0 | 40 | add | 40 |
```

Was ist BDD? - DEMO

Freie Universität Berlin

- Demo!

Was ist BDD? - Vergleich zu TDD

Freie Universität Berlin

- Stakeholder wird direkt an der Entwicklung beteiligt
- Fokus auf Verhalten der Software, allein durch Wortwahl
- Namenskonventionen
- automatisierte Dokumentation

Was ist BDD? - Vergleich zu TDD

Freie Universität Berlin

- bis jetzt keine Studien zu BDD
- Studien zu TDD:
  - erhöht Produktivität der Enwickler_innen
  - erhöht Qualität der Software
  - verbessert Design
  - erhöht Erweiterbarkeit der Software
  - Schwierigkeiten in der Umsetzung

Andere Sprachen, andere Tools

Freie Universität Berlin

- Java $\rightarrow$ JBehave
- C $\rightarrow$ CBehave
- Ruby $\rightarrow$ Cucumber, RSpec
- PHP $\rightarrow$ behat
- Python $\rightarrow$ Lettuce

Quellen

- Dan North, Introducing BDD
- Scott Bellware, Behaviour-Driven Development
- Chelimsky, David and Dennis, Zach and Astels, Dave and Hellesoy, Aslak and Helmpamk, Bryan and North, Dan, The Rspec Book
- Diogo Osorio, Test Driven Development using PHPUnit
- Gherkin, https://github.com/cucumber/cucumber/wiki/Gherkin
- Cucumber, http://cukes.info/

Quellen

- RSpec, http://rspec.info/
- Dan North, Introducing RBehave
- Kaufmann, Reid and Janzen, David, Implications of Test-Driven-Development
- Nagappan, Nachiappan and Maximilien, E. Michael and Bhat, Thirumalesh and Williams, Laurie, Realizing quality improvement through test driven development: results and experiences of four industrial teams
- David Janzen, Software Architecture Improvement