Tests & Test-driven development

Antonia Bücklers

Gliederung

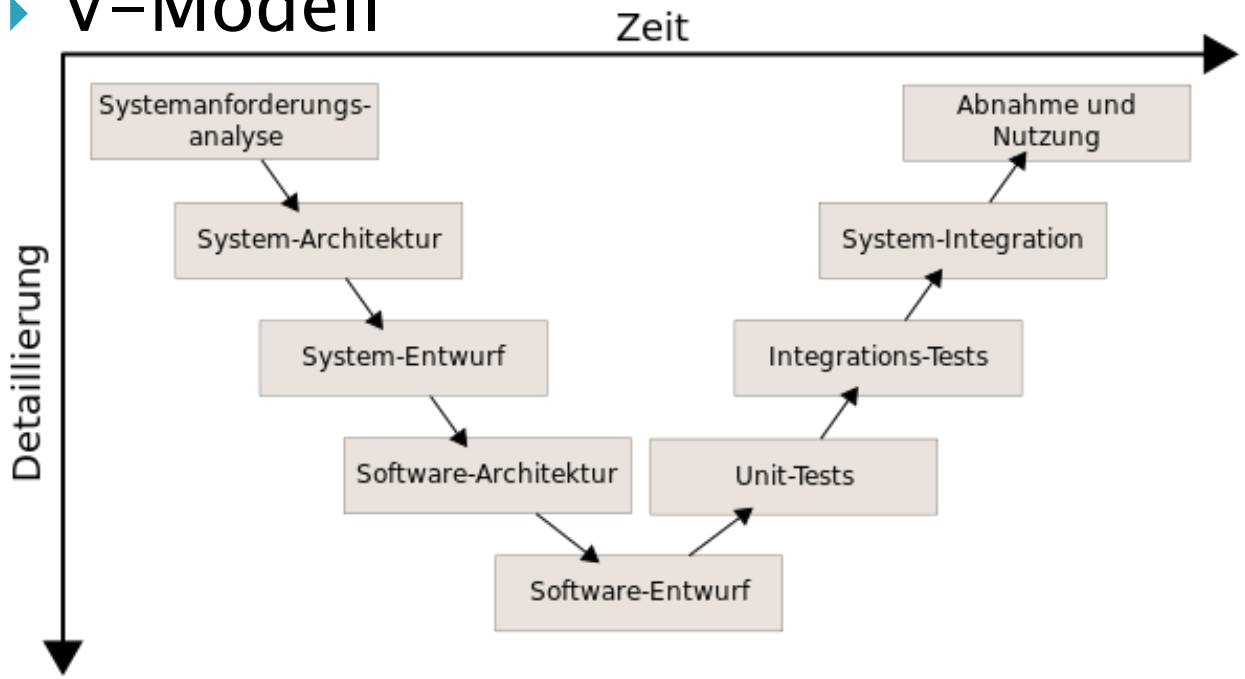
- Softwaretests
- Testgetriebene Entwicklung (TDD) vs wissenschaftliche Methode
- TDD Case Study
- Zusammenfassung

Softwaretest

- prüft und bewertet Software auf Erfüllung der spezifischen Anforderungen
- misst Qualität

Stufen

- Einordnung durch Entwicklungsstand des Systems
- V-Modell



Softwaretest.Wikipedia – Die frie Enzyklopädie.
http://de.wikipedia.org/wiki/Softwaretest

Stufen

- Komponententests / Unit – Tests

→ testen der technischen Lauffähigkeit

→ korrekte fachliche Ergebnisse

- Integrationstests

→ testen der Schnittstellen

- Systemtests

→ testen des gesamten Systems gegen Anforderungen

- Abnahmetests

→ testen durch Kunden

Test Klassifikation

- Prüftechnik (statisch/dynamisch)
- Testkriterium(Funktionale, Schnittstellen, Stress…)
- Zeitpunkt der Durchführung
- Testintensität
- Informationsstand
- …

Box Tests

- White Box Tests:
- Kenntnisse über innere Funktionsweise

- Black Box Tests:
- funktionsorientierte Tests
- nach außen sichtbares Verhalten wird getestet

Black Box vs. White Box

Vorteile Black Box:
- bessere Verifikation des Gesamtsystems
- semantischen Eigenschaften bei geeigneter Spezifikation
- Portabilität

Nachteile Black Box:
- größerer organisatorischer Aufwand
- einige Funktionen nur durch Zufall getestet
- Testsequenzen einer unzureichenden Spezifikation sind unbrauchbar

Black Box vs. White Box

Vorteile White Box:

- Teilkomponenten und innere Funktionsweise werden getestet
- organisatorischer Aufwand gering
- Automatisierung möglich

Nachteile White Box:

- Spezifikation wird nicht geprüft
- um Fehler herum testen möglich

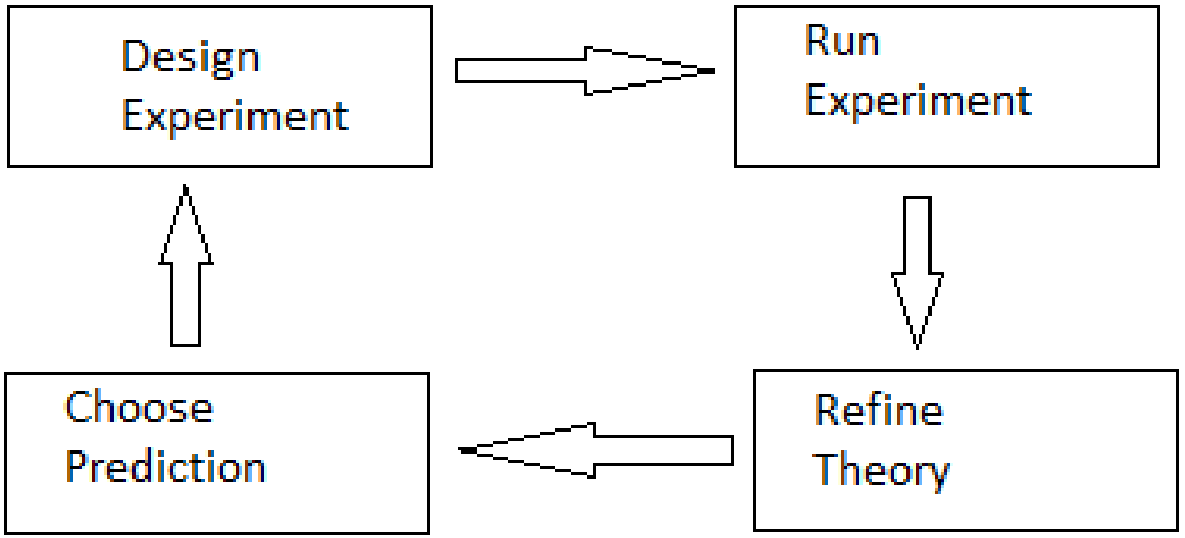
Grey Box Test

- verbinden Vorteile von Black und White Box Tests
- Test vor Programm
- Black Box:
  - Unkenntnis über das Interne
- White Box:
  - geschrieben von Entwicklern des Programms
- Testgetriebene Entwicklung

Testgetriebene Entwicklung (TDD)

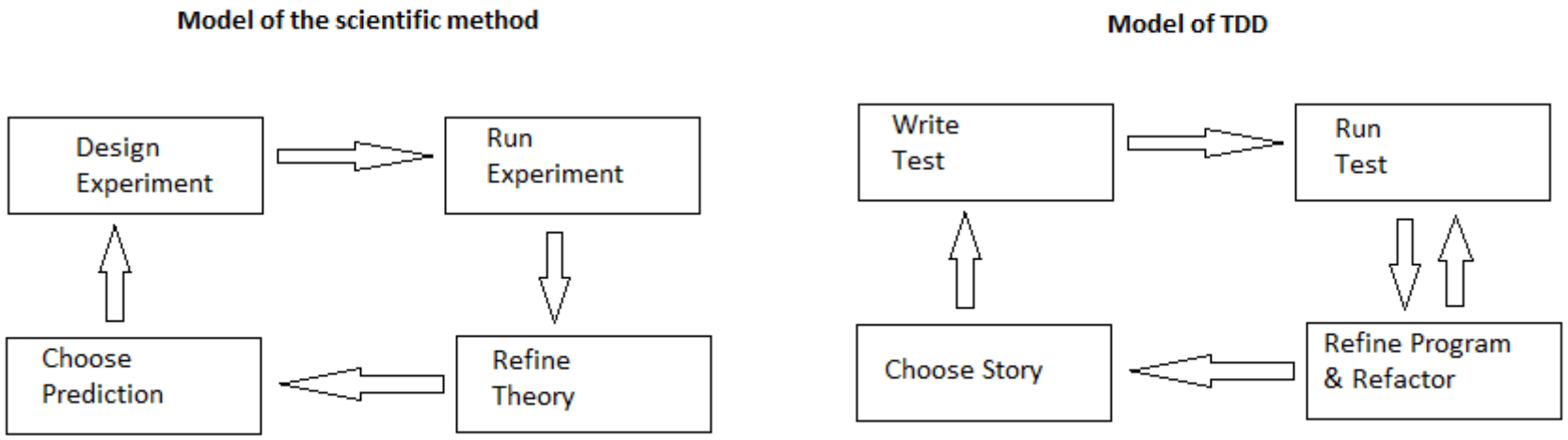
- Definition:
- Softwaretests konsequent vor den zu testenden Komponenten
- Hintergrund:
- oft nicht gewünschte oder erforderliche Testabdeckung

Modell der wissenschaftlichen Methode



Test Driven Development and the Scientific Method. Rick Mudridge

Wissenschaftliche Methode/TDD



Test Driven Development and the Scientific Method. Rick Mudridge

Kritik an TDD

- Konsequenz ist erforderlich
- Übung notwendig
- Einbüße in der Produktivität
- Keine Ersatz für andere Testarten
- andere Testarten eignen sich teilweise besser für bestimmte Testbereiche
- keine Garantie für Fehlerfreiheit
- keine Test Suite
- alle Tests müssen eigenständig entwickelt werden

Extreme Programmierung

- Lösen einer Programmieraufgabe steht im Vordergrund
- Anforderungen des Kunden zu Beginn unklar
- Annäherung an Anforderungen des Kunden
- aktive Teilnahme des Kunden am Produkt
- Anpassung an Anforderungen

→ nur Verwirklichung von dem, was für den Kunden einen Nutzen hat

Extreme Programmierung

- Kundennahe Entwicklung
- Weniger Fehler im Ergebnis
- Kompensation von Krankheitsausfällen

Case Study : Test-Driven Development as a Defect-Reduction Practice

mögliche Vorteile von TDD:

- Effiezienz
- Test Vorteile
- Reduzierung von Programmfehlern

Experiment mit Studienabsolventen

- 2 Gruppen:
  - TDD
  - Conrol group (automatisierte Tests nach Code)

- Kriterien:
  - Entwicklungszeit
  - Zuverlässigkeit
  - Verständlichkeit

Test-Driven Development as a Defect-Reduction Practice

Experiment mit Studienabsolventen

- Ergebnis:
  - kein Unterschied in Entwicklungszeit
  - TDD: weniger Zuverlässigkeit nach Implemetierungsphase
  - TDD: mehr Zuverlässigkeit nach Akzeptanztestphase

→ TDD nicht schneller und keine bessere Qualität

Experiment Professionelle Programmierer

- 2 Gruppen:
  - TDD
  - Control group
  - → Paar Programmierung

- Ergebnis
  - → TDD: 18% mehr erfolgreiche Black Box Tests
  - → TDD: einfacheres Design
  - → TDD: 16% mehr Zeit
  - → Control group: keine rentablen automtisierten Tests

→ Vergleich schwrierig

Test-Driven Development as a Defect-Reduction Practice

Experiment: IBM software engineers

	Legacy 7th Iteration	New 1st Release
Team Size (Developers)	5	9
Team Experience (Language and Domain)	Experienced	Some Inexperienced
Collocation	Collocated	Distributed
Code Size (KLOC) New; Base; Total	**6.6**; 49.9; 56.5	**64.6**; 9.0; 73.6
Language	Java/C++	Java
Unit Testing	Ad hoc	TDD
Technical Leadership	Shared resource	Dedicated coach

Test-Driven Development as a Defect-Reduction Practice. Laurie Williams, E.Michael [illegible]milien, Mladen Vouk.IEEE.2003

Experiment: IBM software engineers

- Ergebnis:
- TDD 40% weniger Programmfehler
- Funtional Verification Tests identisch

- altes Produkt muss auf zwei Plattformen laufen (Windows und Linux)
- neues System nur auf Linux
- altes Produkt funktioniert auf mehr Hardware Plattformen als neues → Testfälle müssen für jede Plattform wiederholt werden

Test-Driven Development as a Defect-Reduction Practice

Experiment: IBM software engineers – Zusammenfassung und Ausblick

- TDD nach UML design Prozess
- 40% weniger Programmfehler nachgewiesen
- minimale Auswirkungen auf Produktivität der Entwickler
- Automatisierte Tests sind wiederverwendbar und erweiterbar
- Tests sind Basis für Qualitätschecks und dienen als Vertrag zwischen allen Mitgliedern des Teams
- Regressionstests

Test-Driven Development as a Defect-Reduction Practice

Zusammenfassung

- Testen wichtig um Qualität zu sichern
- TDD:
  - Test vor zu testendem Programmcode
  - hohe Testabdeckung
  - Übung erforderlich
  - keine Garantie für Fehlerfreiheit
- Case Study:
  - 40% weniger Defects
  - Sehr geringe Auswirkungen auf Produktivität
- viel Hoffnung in TDD

Quellen

Literatur:

- Test Driven Development and the Scientific Method. Rick Mudridge. IEEE.2003
- Test-Driven Development as a Defect-Reduction Practice. Laurie Williams, E.Michael Maximilien, Mladen Vouk.IEEE.2003

- Black-Box-Test. Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Black-Box-Test
- Extreme Programmierung. Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Extreme_Programming
- Grey-Box-Test. Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Grey-Box-Test
- Softwaretest.Wikipedia – Die frie Enzyklopädie. http://de.wikipedia.org/wiki/Softwaretest
- Testautomatisierung.Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Testautomatisierung
- Testgetriebene Entwicklung. Frank Westphal. Extreme Programmer. Ruby on Rails Freelancer. Web 2.0 Technologist. 06.01.2002 http://www.frankwestphal.de/TestgetriebeneEntwicklung.html
- Testgetriebene Entwicklung. Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/Testgetriebene_Entwicklung
- White-Box-Test. Wikipedia – Die freie Enzyklopädie. http://de.wikipedia.org/wiki/White-Box-Test

Testautomatisierung

- Automatisierung von Aktivitäten im Test
- Nutzen: Feedback über Status der Software

Tests (nach Testkriterium)

- Stresstest
- Lasttest
- Fehlertest
- Crashtest
- Sicherheitstest

Phasenmodell

- Testplanung
- Testvorbereitung
- Testspezifikation
- Testdurchführung
- Testauswertung
- Testabschluss