

Testautomatisierung Gute Qualität fällt nicht vom Himmel

Prof. Dr. Karin Vosseberg

Maximilian Azimi, Jens-Rainer Felske, Sebastian Lauber, Jan-Henrich Mattfeld, Pascal Schneider, Krischan Stapelfeldt, Timm Suhl, Nils Techau

SWM 2015, 18.03.2015



Motivation

Motivation

TDD

Veranstaltungskonzept

Ergebnisse

- Testautomatisierung alleine führt noch nicht automatisch zu einer besseren Qualität des Softwareprodukts
- Einbeziehen systematischer Testverfahren
- Zusammenwachsen der Rollen Entwickler und Tester
- Wie kann ich Studierende vorbereiten auf Anforderungen wie
 - Testgetriebene Softwareentwicklung
 - Testautomatisierung
 - Systematisches Testen
 - Symbiose zwischen Requirements Engineering und Testmanagement





Was zeichnet agiles Testen aus?

Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

»Testen in agilen Projekten unterscheidet sich vom klassisches Testen in erster Linia dadurch dass disselben Teste viel häufiger ausgeführt wer dieselben Tests viel häufiger indert (Ref. Automat Automatisierung üher als in klassischen

Tests als ausführbare Spezifikationen

Wie in klassischen Projekten auch unterscheidet man in adien Projekten zwischen der technischen Spezifikati

Spezifikation der Anwendungsdomännen

Akzeptanztests

Testgetriebene Entwicklung

Testgetriebene Entwicklung (Test-Driven Development) bedeutet, dass Tests vor dem Produktivcode geschrieben werden, um so die Softwareentwicklung zu steuern. So entsteht Qualitätssoftware mit sehr

Qualitätssoftware mit sehr hoher Testabdeckung

ww.it-agile.de/wissen/praktiken/agiles-testen/ 16.03.2015

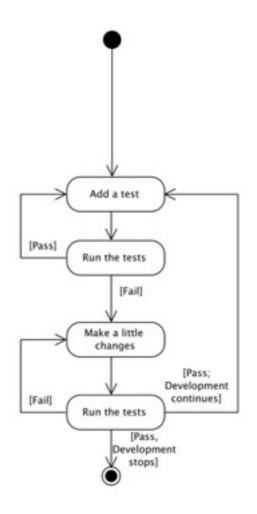
TDD

Motivation

TDD

Veranstaltungskonzept

Ergebnisse





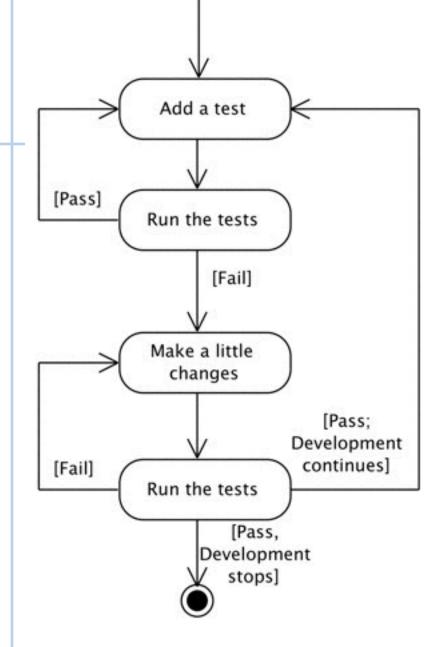


Motivation

TDD

Veranstaltungskonzept

Ergebnisse







Motivation

Maximum von 3 Zahlen

Maximum von 1 Zahl

kleine Iterationen

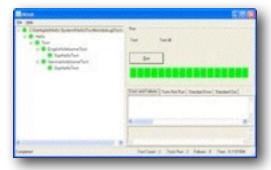
Testfall 7 = maximum(7)

TDD

Veranstaltungskonzept

Ergebnisse

```
public int maximum (int x)
{
  int max=0;
  max=x;
  return max;
}
```







Motivation

TDD

Veranstaltungskonzept

Ergebnisse

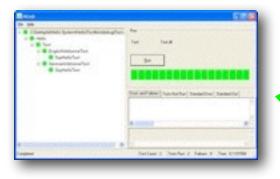
Fazit

Maximum von 2 Zahlen

Testfälle

7 = maximum(7,5)

7 = maximum(5,7)







Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

```
    Maximum von 3 Zahlen
```

max=z;

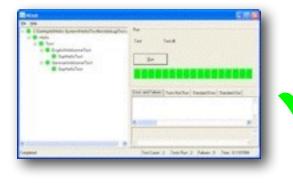
return max;

Testfälle

7 = maximum(7,5,4)

7 = maximum(5,7,4)

7 = maximum(4,5,7)







Motivation

TDD

Veranstaltungskonzept

Ergebnisse

```
    100% Entscheidungsüberdeckung

                                          Testfälle
                                          7 = maximum(7,5,4)
                                         7 = maximum(5,7,4)
  public int maximum (int x=7/5/4,
                                         7 = maximum(4,5,7)
                         int y=5/7/5,
                         int z=4/4/7)
     int max=0;
                        int max=0;
                                              int max=0;
     if (7>4)
                        if (5>4)
                                              if (4>7)
        max=7;
                            max=5;
                                                 max=x;
     if (5>7)
                                              if (5>4)
                        if (7>5)
        max=y;
                            max=7;
                                                 max=5;
     if (4>5)
                        if (4>7)
                                              if (7>5)
        max=z;
                                                 max=7;
                            max=z;
     return max=7;
                        return max=7;
                                             return max=7;
```



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

```
systematische

    Maximum von 3 Zahlen

                                           Testfälle
 public int maximum (int x,
                                           x > y > z
                           int y,
                                           x > z > y
                                                      Reihenfolge der
                           int z)
                                           y > x > z
                                                      beiden kleineren
                                           y > z > x
                                                      Werte
    int max=0;
                                           Z > X > V
    if (x>z)
                                           z > y > x
                                           X = V > Z
        max=x;
    if (y>x)
                                           x = y < z
                                                       zwei oder drei
                                           X = Z > A
        max=y;
                                           X = Z < A
                                                       gleiche Werte
    if (z>y)
                                           y = z > x
        max=z;
                                           y = z < x
    return max;
                                           x = y = z
```



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

```
    Maximum von 3 Zahlen
```

```
int y,
int z)

int max=0;
if (x>z)
    max=x;
if (y>x)
    max=y;
if (z>y)
    max=z;
return max;
```

public int maximum (int x,

systematische Testfälle

```
y > x > z
y > z > x
Z > X > V
z > y > x
X = V > Z
x = y < z
X = Z > A
X = Z < A
y = z > x
y = z < x
x = y = z
```

Testfälle

```
7 = maximum(7,5,4)
7 = maximum(5,7,4)
```

$$7 = maximum(4,5,7)$$



systematische Maximum von 3 Zahlen Motivation **Testfälle TDD** public int maximum (int x=7, x > y > zint y=4, (x > z > y 5 = maximum(7,4,5) int z=5) Veranstaltungsy > x > zkonzept y > z > xint max=0; z > x > yErgebnisse if (7>5) z > y > xmax=7;x = y > zif (4>7)x = y < zFazit X = Z > Amax=y;X = Z < Yif (5>4)y = z > xmax=5;y = z < xreturn max=5; x = y = z



systematische Maximum von 3 Zahlen Motivation **Testfälle TDD** public int maximum (int x=7, x > y > zint y=7, x > z > yint z=7) Veranstaltungsy > x > zkonzept y > z > xint max=0; z > x > yif (7>7) Ergebnisse z > y > xx = y > z $\max=x;$ if (7>7) x = y < zFazit X = Z > Amax=y;X = Z < Aif (7>7) y = z > xmax=z;y = z < xreturn max=0; x = y = z 0 = maximum(7,7,7)



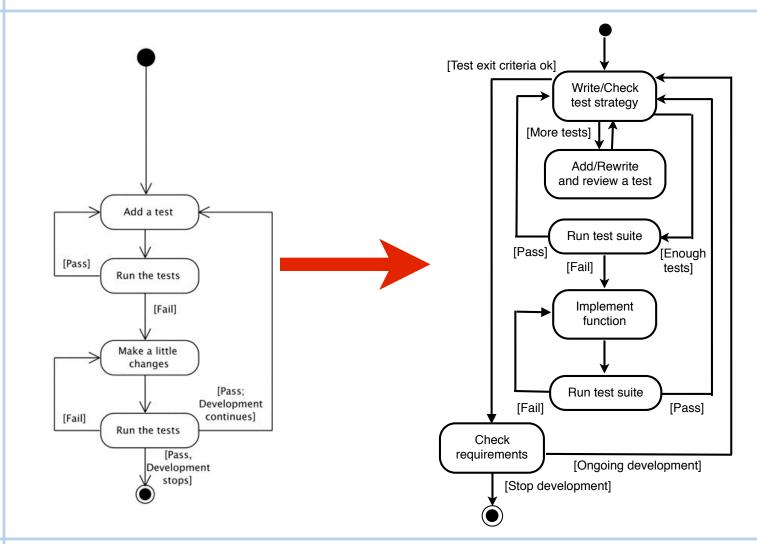
TDD++

Motivation

TDD

Veranstaltungskonzept

Ergebnisse







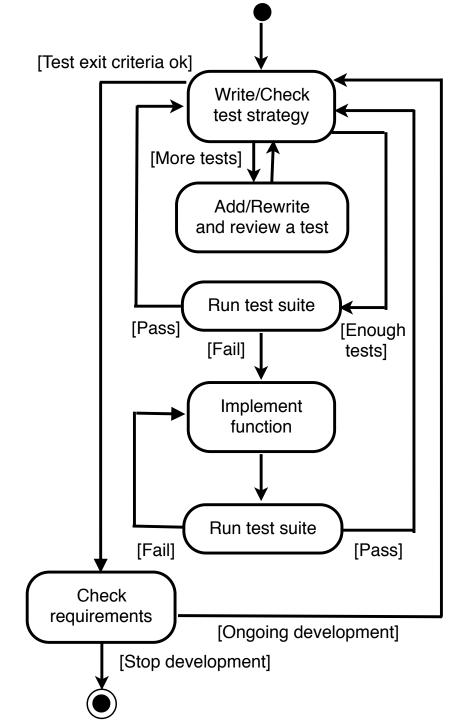
TDD++

Motivation

TDD

Veranstaltungskonzept

Ergebnisse





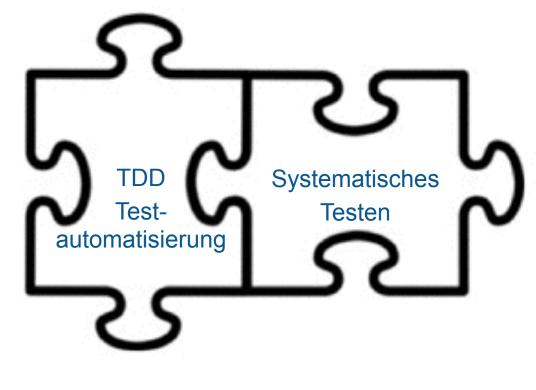
Veranstaltungskonzept

Motivation

TDD

Veranstaltungskonzept

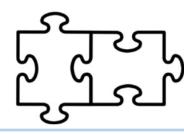
Ergebnisse







Veranstaltungskonzept



Motivation

TDD

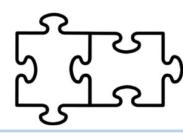
Veranstaltungskonzept

Ergebnisse

TDD	Systematisches
Testautomatisierung	Testen
Entwicklung einer Gebäudeverwaltung	ISTQB Certified Tester Foundation Level
 Excel-Sheet mit Anforderungen	 Grundlagen des
und Akzeptanzkriterien	Softwaretestens
 Erstellen von Testfällen mindestens Systemtest und	Testen im
Unittest Testentwurfsverfahren	Softwarelebenszyklus Statischer Test Testentwurfsverfahren
TestautomatisierungContinous IntegrationJubula, JUnit	Black-boxWhite-boxErfahrungsbasiert
• Jacoco •	TestmanagementTestwerkzeuge



Rahmenbedingungen



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

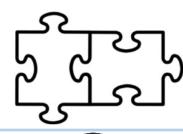
1 Semester, 4 SWS, 6 CP

- Vorlesung "Certified Tester" 8 Blöcke
- 7-8 Workshops zur Unterstützung des Projekts
- Abgabe des Projekts am Ende des Semesters
- Teams von 4-5 Personen
- Workshops
 - Frank Düsterbeck, HEC GmbH: QS in agilen Projekten Björn Seebeck, HEC GmbH: Aufbau einer Testpipeline
 - Alexandra Schladebeck, Bredex GmbH: Jubula
 - ATDD und TDD
 - Continuous Integration
 - Björn Seebeck, HEC GmbH: Unterstützung im Aufbau der eigenen Testpipeline
- Zusatzqualifikation zum ISTQB Certified Tester Foundation Level
 - Prüfung durch iSQI





Ergebnisse



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

- 2 Teams (4 und 5 Studierende) haben eine Testpipeline erstellt
 - GitHub zur Verwaltung des Source Codes
 - Jenkins als Integrationsserver
 - Maven als Build-Management Werkzeug
 - Checkstyle und FindBugs zur statischen Analyse der Sourcen
 - Jubula Implementierung und Ausführung von Testfällen auf der Integrations- und Systemteststufe
 - JUnit Implementierung und Ausführung von Testfällen auf der Unitteststufe
 - JaCoCo zur Ermittlung der Codeabdeckung
 - Mantis zur Fehlerverfolgung
 - TestLink als Testmanagementwerkzeug : Verwaltung von Testfällen.







Test





Build-Auslöser

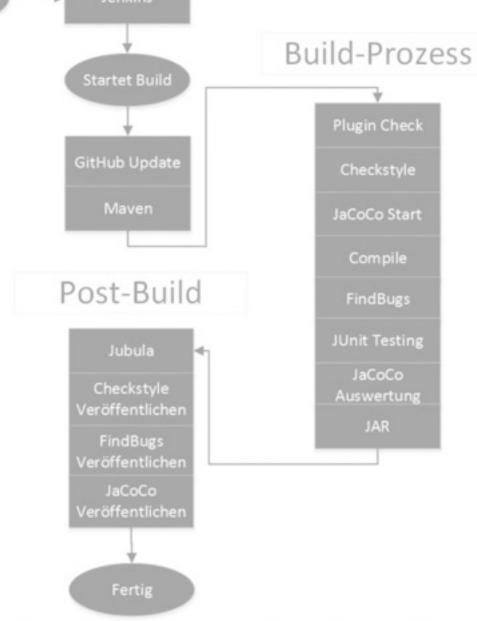
Build-Auslöser Jenkin

Motivation

TDD

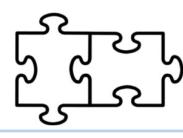
Veranstaltungskonzept

Ergebnisse





Ergebnisse



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

2 Teams (4 und 5 Studierende)
 haben (überwiegend) testgetrieben eine
 Gebäudeverwaltung entwickelt

- Review der Anforderungen
- Testfälle auf Basis von Testentwurfsverfahren entwickelt Äquivalenzklassen und Grenzwertanalysen Whitebox-Testentwurfsverfahren Zustandsbasiertes Testen Kontrollflussgraphen zur Prüfung ihrer Testfälle
- Testfälle auf unterschiedlichen Teststufen Systemtest, Integrationstest, Unittest
- Testendekriterien auf Basis der Anforderungen Positv- und Negativ-Testfälle
- Einfluss des Testens auf die Entwicklung reflektiert





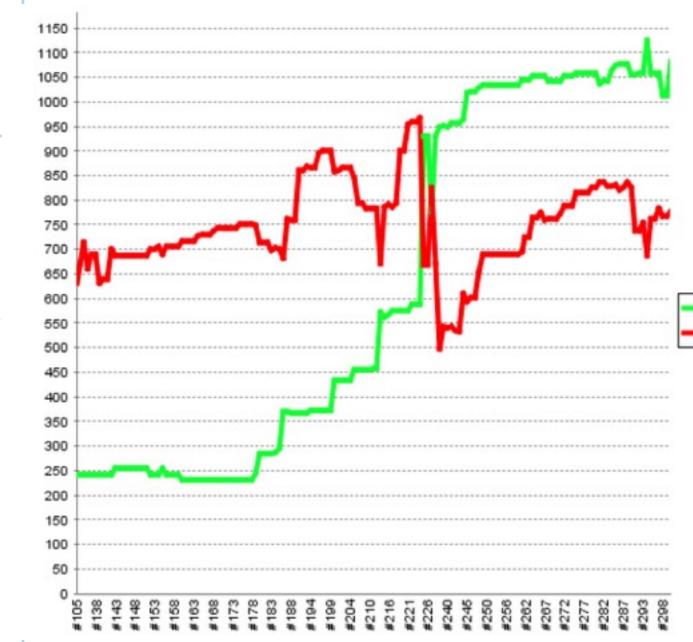
Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit

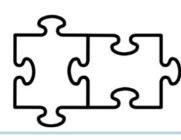


Zeilen abgedeckt

Zeilen nicht abgeded



Ergebnisse



Motivation

TDD

Veranstaltungskonzept

Ergebnisse

- 2 kleinere Teams (2 und 1 Studierende) mit einer abgespeckten Aufgabenstellung
 - Review der Anforderungen
 - Testkonzept entwickelt
 - Testfälle auf Basis von Testentwurfsverfahren entwickelt
 - Hauptfunktionalitäten der Gebäudeverwaltung entwickelt
 - Testfälle auf der Unittestebene mit JUNIT implementiert



Fazit

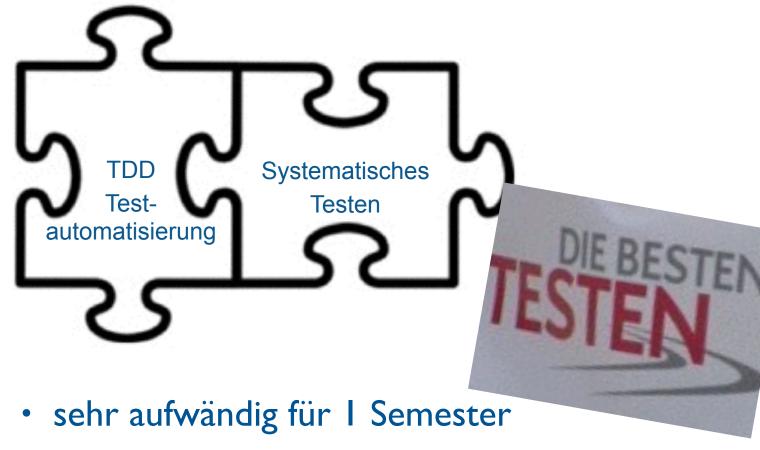
Motivation

TDD

Veranstaltungskonzept

Ergebnisse

Fazit



trotzdem sehr gute Ergebnisse



Fragen?

Motivation

TDD

Veranstaltungskonzept

Ergebnisse

