

# Tests

Schwarz, Christian (s68419)

4. Juli 2014

# Überblick

Aufgabenbereich

Entstehung eines Testberichtes

Statistik

Reflexion

Sicherheitshinweise

# Aufgabenbereich

- ▶ Testen der aktuellen Version
- ▶ Prüfung auf Sicherheit

# Entstehung eines Testberichtes

- ▶ Testfälle durchgehen und Ergebnis festhalten
- ▶ Testdokumentation erzeugen
  - ▶ Testberichte nach LaTeX übersetzen
  - ▶ (Testberichte parsen)
  - ▶ (Diagramm erzeugen)
  - ▶ Testdokumentation zusammensetzen
  - ▶ PDF erzeugen

# Testbericht in Rohform

```
1# Testbericht vom 30.06.2014
2
3## Testumgebung
4
5* Compiler: integrierter Compiler von Visual Studio 2012 (S355)
6* Testumgebung: Win7 (S355)
7
8## allgemeine Anmerkungen
9
10* Mailadressen mit Leerzeichen werden angenommen
11
12## Programm für den Dozenten
13
14### Beleg anlegen
15
16* leerer name           -      ok (Fehlermeldung)
17* langer name           -      fail (10 Zeichen gehen nicht durch, sollten aber laut Fehlermeldung)
18* leeres Passwort       -      ok (Fehlermeldung)
19* langes Passwort       -      ok (Fehlermeldung mit Längenangabe)
20
21* bei Nichtänderung der Belegkennung werden für weitere Belege nicht mehr alle verfügbaren Themen angeboten
22
23### Gruppe anlegen
24
25* noch kein Beleg vorhanden -      ok (Fehlermeldung)
26* leeres Passwort       -      ok (Fehlermeldung)
27* langes Passwort       -      ok (Fehlermeldung)
28
29### Tabelle bearbeiten
30
31...
```



# Endprodukt

## 1 Testbericht vom 02.06.2014

### 1.1 Testumgebung

- Compiler: keiner (vorkompiliertes Binary benutzt)
- Testumgebung: wine 1.7.19 auf Debian 3.2.54

### 1.2 allgemeine Anmerkungen

- Mailadressen mit Großbuchstaben werden nicht angenommen

### 1.3 Programm für den Dozenten

#### 1.3.1 Beleg anlegen

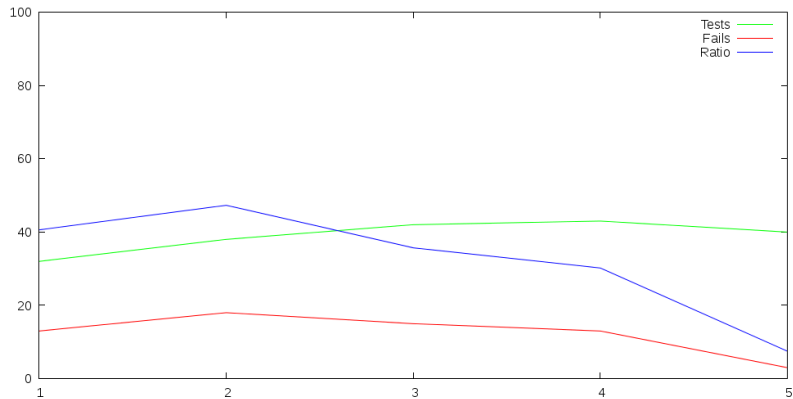
leerer name	ok (Fehlermeldung)
langer name	wird abgeschnitten (sollte Fehler auslösen)
normaler name	ok

# Zahlen

durchgeführte Tests:	5
durchschnittliche Anzahl Testfälle:	39
durchschnittliche Fehlerrate:	32.26%



# Diagramm



# Was habe ich gelernt?

- ▶ Ruby-Kenntnisse verbessert
- ▶ .NET und wine zum laufen gebracht
- ▶ Techniken zum umformatieren von Text erlernt
- ▶ L<sup>A</sup>T<sub>E</sub>X-Kenntnisse vertieft

## Was hat sich bewährt, was nicht?

hat sich bewährt:

- ▶ Ruby-Script zur automatischen Generierung der Dokumentation
- ▶ Verwendung von  $\text{\LaTeX}$  zur Dokumentation der Tests
- ▶ Nutzung von GNUplot zur Erzeugung von Diagrammen

hat sich nicht bewährt:

- ▶ Nutzung von wine zum Testen von .NET-Anwendungen

# Was würde ich wieder so machen, was nicht?

würde ich wieder so machen:

- ▶ Verwendung von  $\text{\LaTeX}$  zur Dokumentation der Tests
- ▶ Nutzung von GNUplot zur Erzeugung von Diagrammen

würde ich anders machen:

- ▶ Testingframework nutzen
- ▶ Implementation mit Unit-Tests begleiten

# Sicherheitshinweise

- ▶ Zugangsdaten werden im Klartext übertragen
- ▶ keine Nutzerverwaltung innerhalb der Datenbank