

Testen

Prof. Sven Apel

Universität des Saarlandes



Teil I

Einführung



Software is eating the world!

(Andreessen, The Wall Street Journal, 2011)

Software arbeitet nicht immer korrekt



SOFTWARE **FAIL** WATCH

5TH EDITION

LOSSES FROM SOFTWARE FAILURES (USD)

1,715,430,778,504

ONETRILLIONSEVENHUNDREDFIFTEENBILLIONFOURHUNDREDTHIRTYMILLIONSEVENHUNDREDSEVENTY-EIGHTTHOUSANDFIVEHUNDREDFOUR

Fehler in der Praxis

Industrie:

- 30-85 Fehler per 1000 LOC;

- 0,5-3 Fehler per 1000 LOC werden nicht erkannt vor Auslieferung.

- 1% der Software-Fehler rufen 50% der Crashes hervor

Snapshot von Mozilla's Bugzilla Bug Datenbank

- Gesamte Historie von Mozilla; alle Produkte und Versionen

- 60.866 offene Bug reports

- 109.756 zusätzliche reports markiert als Duplikate

Snapshot von Mozilla's Talkback Crash Reporter

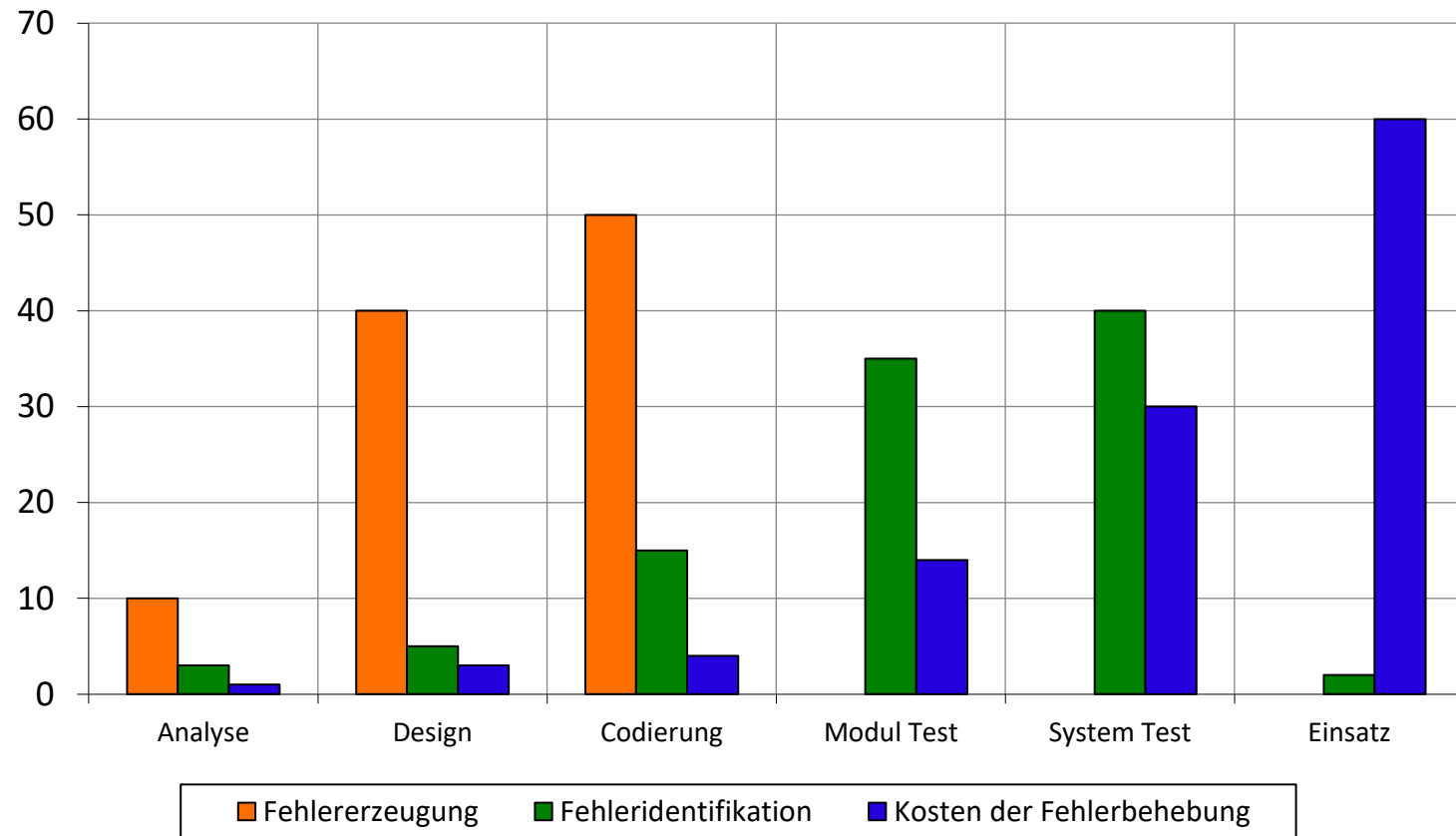
- Firefox 2.0.0.4 der letzten 10 Jahre

- 101.812 eindeutige Nutzer

- 183.066 Crash reports

- 6.736.697 Stunden von user-driven "testing"

Kosten durch Fehler



Ziele von Testen

Sicherstellen, dass...

... das Programm *nicht abstürzt*

... keine *neuen Fehler* auftreten

... *alle Anforderungen* erfüllt sind

... es *keine Seiteneffekte* gibt

Ziele von Testen

"Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence"

Edsger Dijkstra

Fehler finden → ein *erfolgreicher Test findet Fehler*

Vertrauen in Software herstellen

Herausforderungen

Man muss annehmen, dass ein Programm fehlerhaft ist; nicht, dass es korrekt ist

Testen ist *teuer*

Effektivität von Tests *schwer zu messen*

Unvollständige, nicht-formalisierte und sich *ändernde* Anforderungen

Integrationstest in *komplexen Umgebungen*

Steigende Anzahl von Versionen und Varianten

00:00,2

arte

Big-boys.com

K



Testen im SoPra

Sie können Ihren Schein nur bekommen,
wenn Ihre Software *möglichst korrekt* läuft

Sie müssen Tests schreiben. *Viele Tests!*

In der Gruppe sowie *allein*