

Reliability Evaluation of Web-Based Software Applications

Kartika Maramis

Inhalt

1. Motivation
2. Typische Fehler im Web
3. Testzuverlässigkeit im Web
4. Schritte zur eingeführten Methodik
5. Case Study: SOGU
 - Bewertung nach idealem Fall
 - Bewertung nach realem Fall
6. Fazit

Motivation

- Massivzugriff von Benutzer
- Ursprünglichen Error schwer zu finden
- Integration von vielen unterschiedlichen Softwarekomponenten
- Vielfältigkeit des Bezugssystems von Entwicklern

Typische Fehler im Web

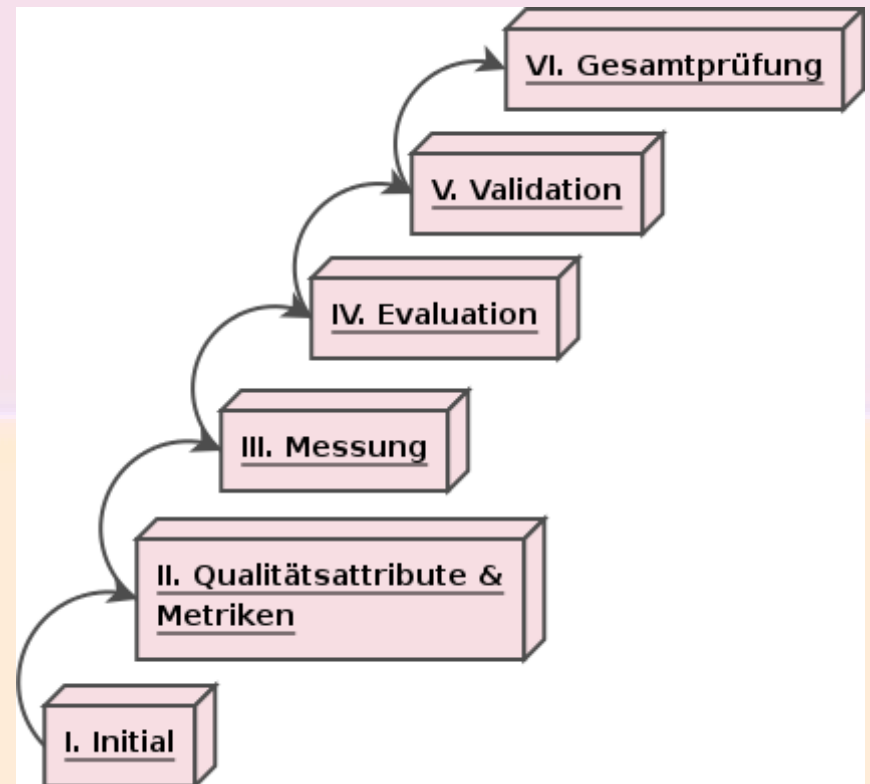
- Host/Netzwerk Fehler (Zentralcomputer)
 - Meistens kommt aus Middleware- oder Webservice-Ebene
- Browser Fehler (Client)
 - Verschiedene Technologien des Web-Browsers
- **Quellen-/Inhaltsfehler (Server)**
 - Serverseitig
 - Keine Übereinstimmung mit Benutzeranforderung

Testzuverlässigkeit im Web

- Annäherung an Zeitdomänen
 - Stochastischer Prozess
 - Softwarezuverlässigkeitswachstumsmodell
 - „Markow-Kette“
- Annäherung an Eingabedomäne
 - Wiederholtes Fehlermuster
 - Eingabe von Navigationsmuster der Webanwendung

Schritte zur eingeführten Methodik

- I. Strukturelle Tester, Systemleistung, Verteilungsparameter festlegen
- II. Qualitätsattribute & Metriken festlegen
- III. Identifiziert & analysiert abweichende Module
- IV. Bewertung durch gewähltem statischen Verfahren
- V. Resultat prüfen & Relevanz nachvollziehen
- VI. Vorausberechnen von Verhaltensweise der getesteten Software



Case Study: SOGU

| | |
|-----------------------|---|
| Plattform | : RedHat |
| WebServer | : Apache |
| DBMS | : MySQL |
| Interfacesprache | : PHP |
| LOC | : 1200 |
| Entwicklungsdauer | : 6 Monate |
| Datenzugriff | : Prof.(rwxr), Stud.(rwxr), Pub.(---r) |
| Normalverteilung | : $\mu = 3$ Sek., $\sigma = 1$ Sek. |
| Exponentialverteilung | : $\mu = 4$ Sek. |
| Qualitätsattribute | : Zuverlässigkeitsgrad des fehlerfreien Ablaufs |
| Metrik | : <i>Defect density</i> |

Case Study: SOGU (Fort.)

$$f(x) = \alpha \beta^{-\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} \quad (1)$$

$$\frac{\sum_{i=1}^n X_i^\alpha \ln X_i}{\sum_{i=1}^n X_i^\alpha} - \frac{1}{\alpha} = \frac{\sum_{i=1}^n \ln X_i}{n} \quad (2)$$

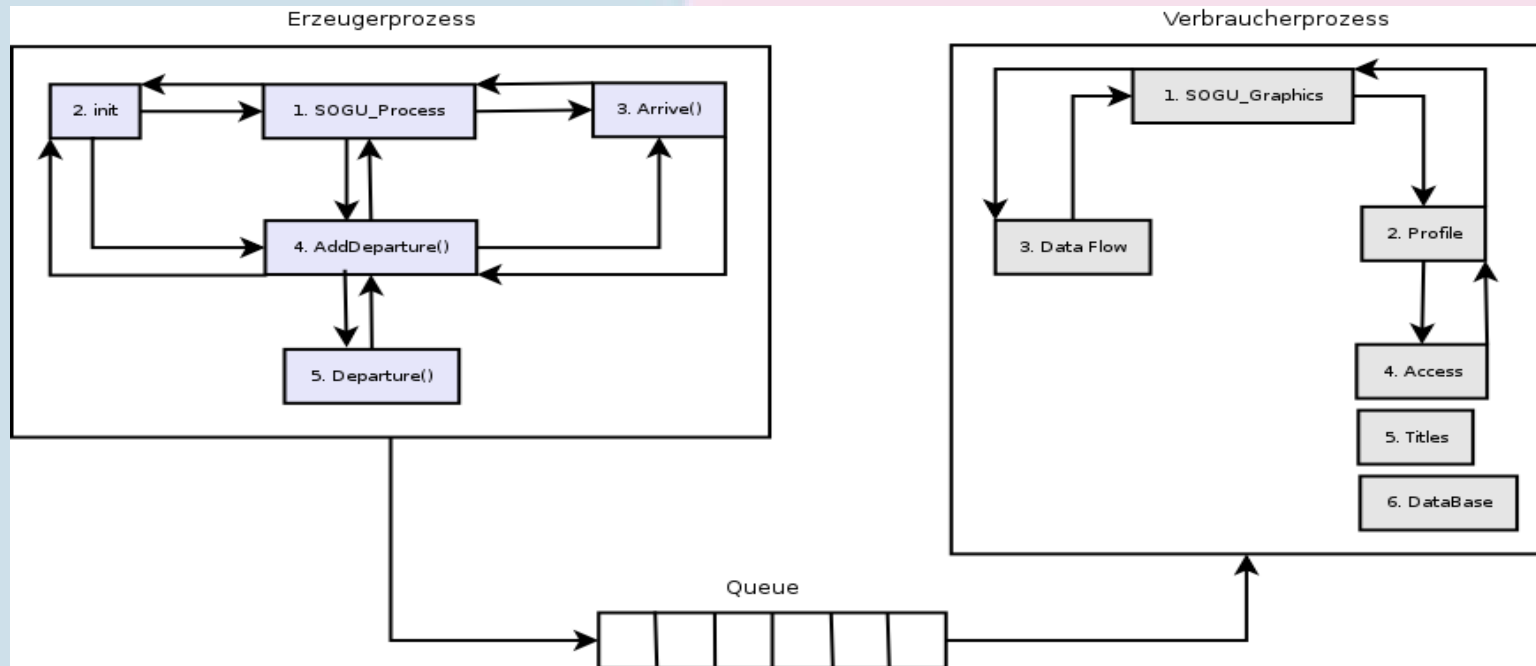
$$\beta = \left(\frac{\sum_{i=1}^n X_i^\alpha}{n} \right)^{1/\alpha} \quad (3)$$

Case Study (Fort.)

„Bewertung nach idealem Fall(1)“

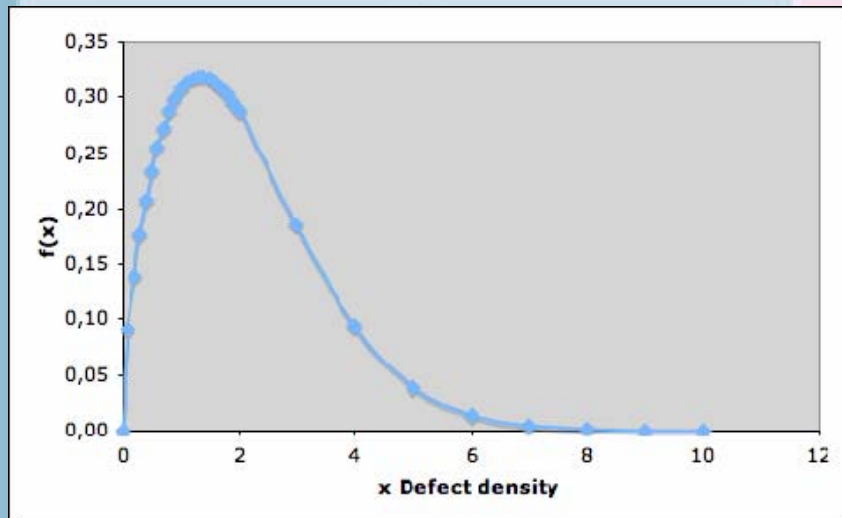
- keine Überlastung und Überschneidungsprobleme
- Systemleistung:

500 MHz (Intel) + 70 GB + 500 MB = 100 gleichzeitiger Benutzer



Case Study (Fort.)

„Bewertung nach idealem Fall(2)“



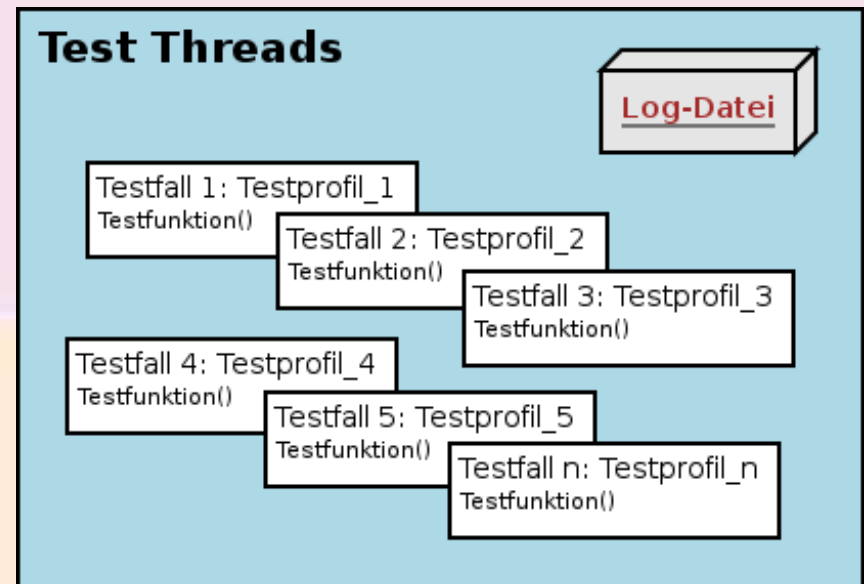
» $\alpha=1,63$ $\beta=2,4$

- Weibull-Verteilung ist in dem Fall geeignet um Grad des fehlerfreien Ablaufs zu modellieren
- Geringe Werte der *defect density* finden unter höherer Frequenz statt

Case Study (Fort.)

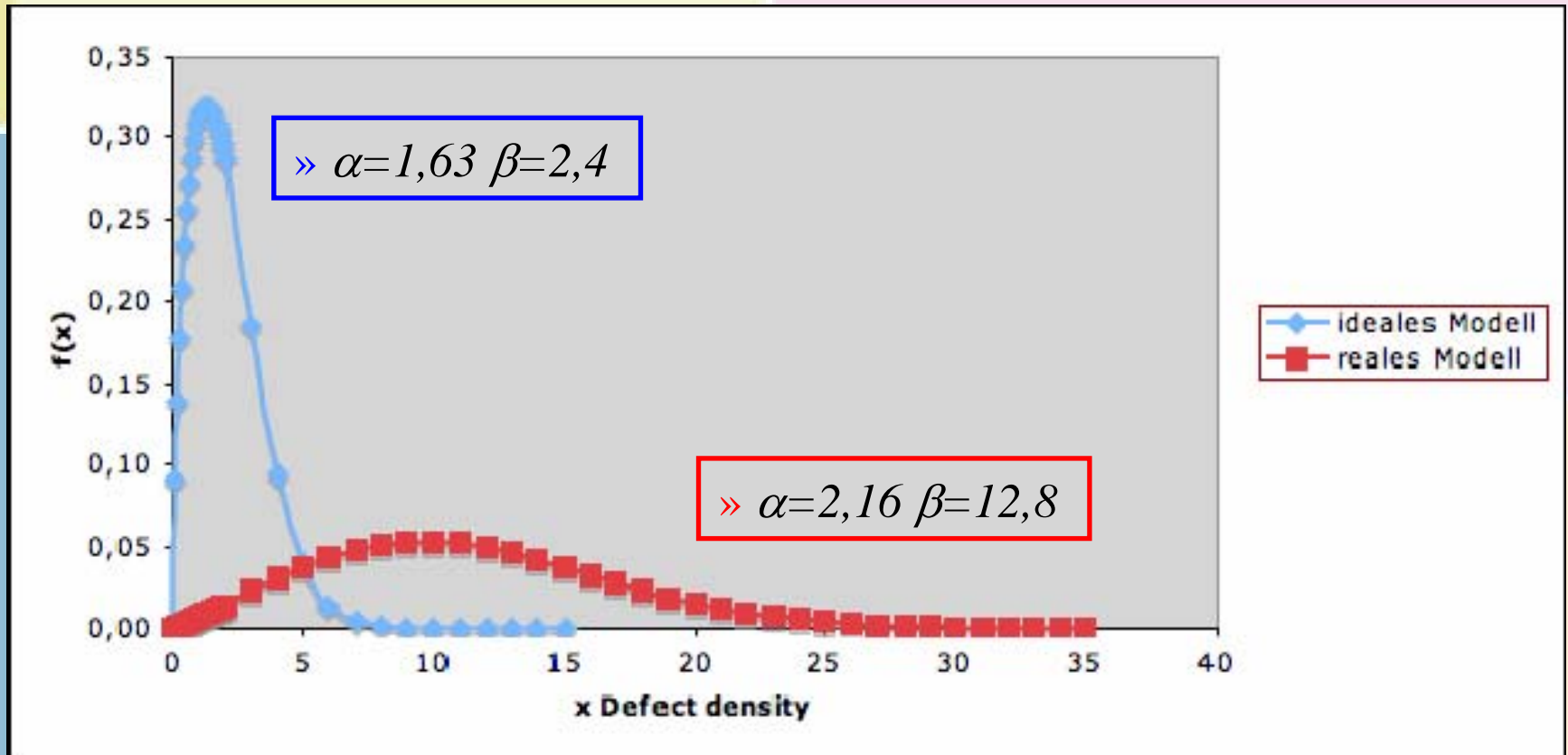
„Bewertung nach realem Fall(1)“

- Testfall zufällig ausgewählt
- Testprofil:
 - bezeichnet Navigationspfad
 - schaut an, ob der Prüfer prüft
- Log-Datei beinhaltet Ereignisse
- Log-Datei gelesen von generellem Abtaster und beim Fehler:
 - generiert Error-Log-Datei
 - erhöht *defect density*



Case Study (Fort.)

„Bewertung nach realem Fall(2)“



Fazit

- Implementierung von statistischer Modellierung
- Struktureller Tester um *defect density* und mittleren Fehlerfall einer Software zu analysieren
- Qualitätsattribute nach idealem und realem Fall vergleichen
- Entwickelt eine effektive Methodik zum Abschätzen der Zuverlässigkeit der Web-basierten Anwendungssysteme

Quelle: Leticia Davila-Nicanor, Pedro Mejia-Alvarez: *Reliability Evaluation of Web-Based Software Applications*,