veritas
iustitia
libertas
Freie Universität Berlin

Seminar

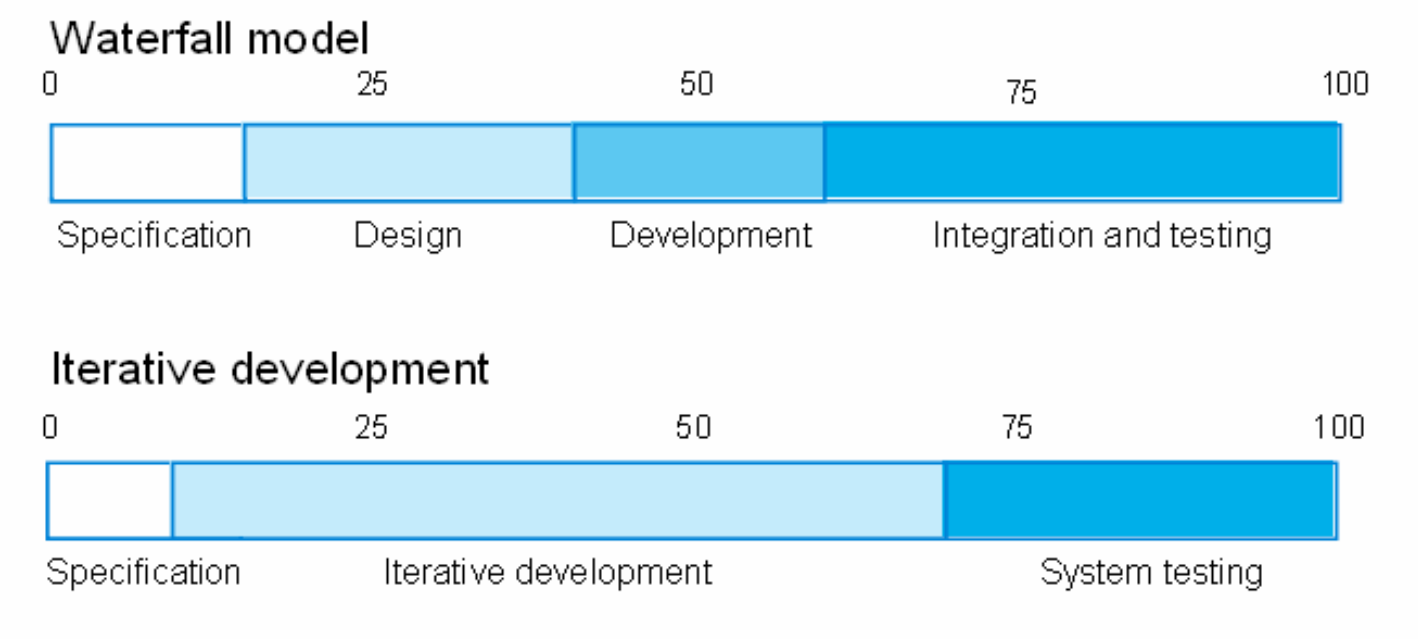
Ursachen und Vermeidung von Fehlern in der Softwareentwicklung (Teil 2)

Sebastian Jekutsch
Freie Universität Berlin, Institut für Informatik
AG Software Engineering

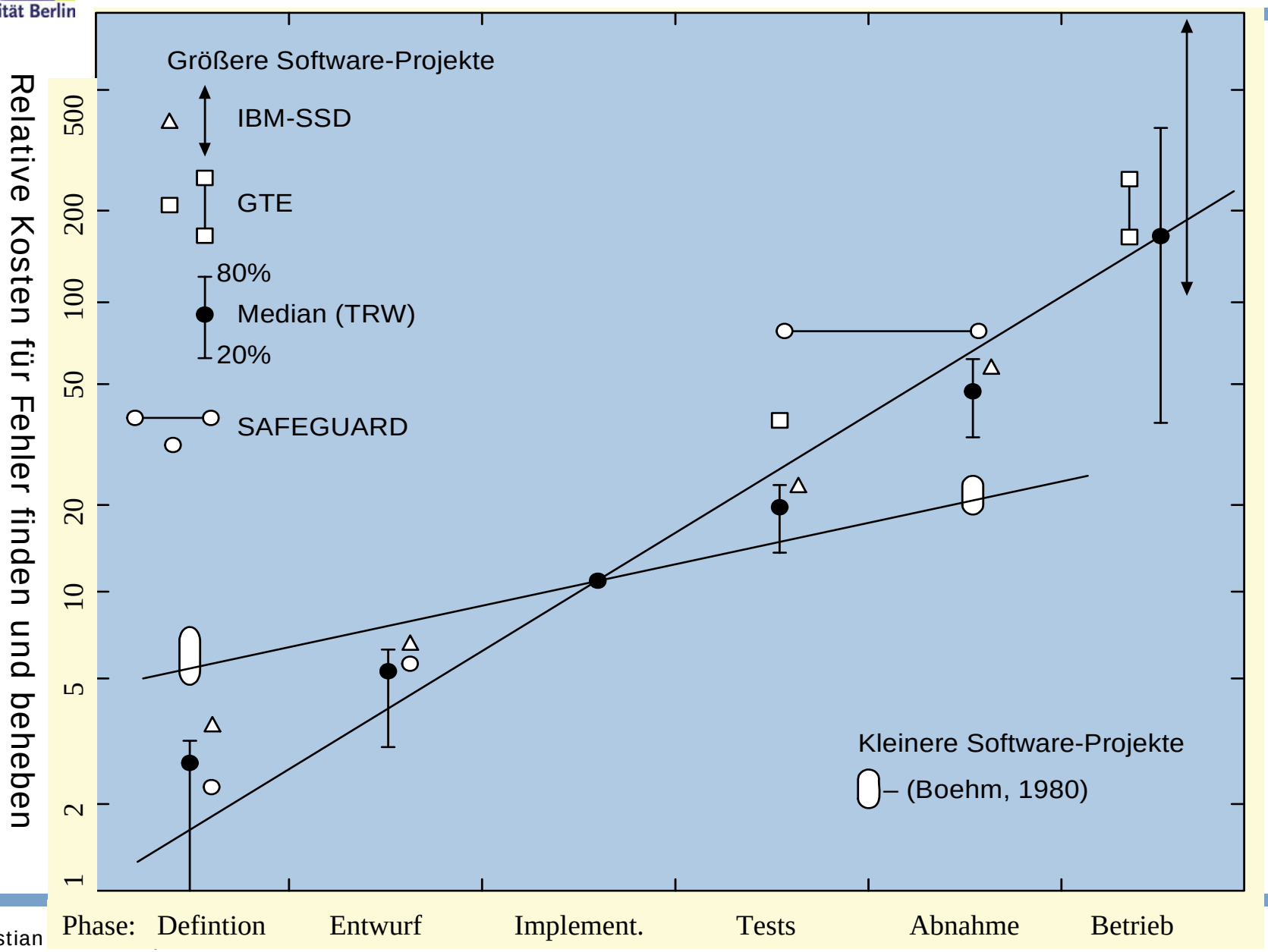
- Motivation
- Themen
- Organisation

Aufwände für Testen und Debugging

- Grob 60% sind Entwicklungskosten, 40% sind Testkosten



Späte Defektentdeckung kostet besonders viel



Testen und Debugging in der Software-QS

- QS in der Softwareentwicklung ist vor allem:
  - Testen = absichtliches Verursachen von Versagen
  - Debugging = Suchen nach ursächlichen Defekten
  - Durchsicht = Suchen nach Defekten ohne Testen
- Das Prinzip scheint also zu sein:
  Erst Defekte einbauen,
  dann Defekte suchen,
  dann Defekte wieder ausbauen.

$\Rightarrow$ alles nur Symptombekämpfung!

- Warum baut man Defekte überhaupt ein?
  Keiner weiß es wirklich.
- Wie kann man verhindern, dass es passiert?
  Es gibt tausend Ideen.

Ursachen: Keiner weiß es wirklich?

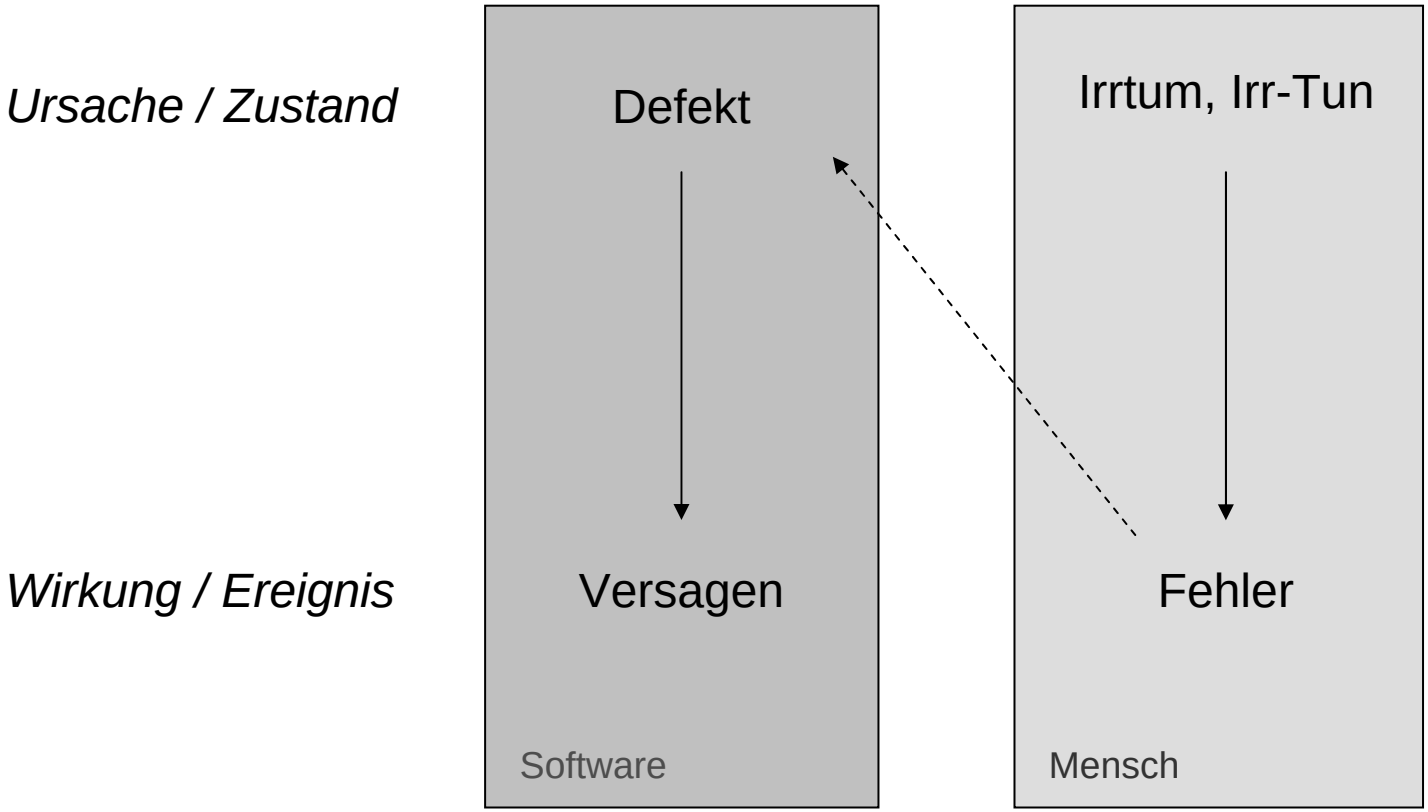
- Psychologie
  - Menschliches Versagen
  - Denk- und Merkschwächen
- Arbeitswissenschaft
  - Stress, Druck, etc.
  - Arbeitsumgebung
- Informatik
  - Programmkomplexität

- Weitere Ideen?

Vermeidung: Tausend Ideen!

- Prozessverbesserung
  - Persönliches Lernen aus Defekten
  - Ursachenforschung gesamtheitlich
  - Formale Methoden
- Hilfe für den Programmierer
  - Programmiersprachensemantik
  - Programmiertipps
  - Erinnerungs- und Lernhilfen
- Weitere Ideen?

Zentrale Begriffe



Und was ist dann ein „Bug“?

Photo # NH 96566-KN First Computer "Bug", 1945

92

9/9

0800 Arctan started
1000 " stopped - arctan ✓ { 1.2700 9.037 847 025
9.037 846 795 correct
13"uc (032) MP - MC ~~1.986 000 000~~ ~~2.130 476 415 (-3)~~ 4.615 925 059 (-2)
(033) PRO 2 2.130 476 415
correct 2.130 676 415
Relays 6-2 in 033 failed special speed test
in relay " 11.00 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multy Adder Test.
1545  Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1700 closed down.

Relay 2145
Relay 3370

Nun zu Organisatorischem…

Seminartätigkeiten

- JedeR bekommt ein Thema (~2 Papers + Recherche)
- Vortrag darüber etwa 45 Minuten (max. 50 min.)
  - Diskussion / Arbeit danach etwa 30 Minuten
  - Video vom Vortrag kann gemacht werden
- Ausarbeitung passender Länge (10-20 Seiten)
- Zwei Begutachtungen der Folien und der Ausarbeitung

$\Rightarrow$ benoteter Schein

- Tipps und Formales unter *http://projects.mi.fu-berlin.de/w/bin/view/SE/SeminarRegeln*

- Warnung: Ein Seminar ist aufwändiger als die meisten glauben!

Seminarablauf

- Heute: Festlegung des *Termin*s des Blocks und Themenauswahl
- Demnächst: Vergabe der Begutachtungsaufgaben und genaue Zeitplanvorgabe
- bis Ende Januar: Besprechung der Gliederung mit dem Seminarleiter anhand vorläufiger Folien
- 1 Monat vor *Termin*: Abgabe an Gutachter
- 2 Wochen vor *Termin*: Abgabe Begutachtungen
- bis *Termin*: Abgabe Ausarbeitung
- *Termin*: Blockseminar findet statt
- Mitte April: Scheinvergabe

- *Unbedingt in Mailingliste eintragen!*

Terminfestlegung

Februar

	Mo	Di	Mi	Do	Fr	Sa	So
5			1	2	3	4	5
6	6	7	8	9	10	11	12
7	13	14	15	16	17	18	19
8	20	21	22	23	24	25	26
9	27	28					

März

	Mo	Di	Mi	Do	Fr	Sa	So
9			1	2	3	4	5
10	6	7	8	9	10	11	12
11	13	14	15	16	17	18	19
12	20	21	22	23	24	25	26
13	27	28	29	30	31		

April

	Mo	Di	Mi	Do	Fr	Sa	So
13						1	2
14	3	4	5	6	7	8	9
15	10	11	12	13	14	15	16
16	17	18	19	20	21	22	23
17	24	25	26	27	28	29	30

- Scheine bis 1. April nötig?
- Klausuren?
- Abstimmung (Woche): (8), 9, 12, 13, 14
  - Drei Vorträge pro Tag
  - Voraussichtliche Uhrzeiten:
    9:00-10:15, 10:30-11:45, 13:00-14:15 Uhr

Seminare der AG Software Engineering

- Open-Source
- Agile Methoden
- Ausgewählte Beiträge zum Software Engineering
  - Beinahe beliebige Themen zum Software Engineering
  - 15 Termine beliebig verteilt
  - 1 Vortrag + Ausarbeitung
  - jeweils Donnerstag 16-18 Uhr im Raum 046

Themen

1. (Einstieg)
2. Defektuntersuchungen
3. COQUALMO
4. Defektabschätzung allg.
5. Modulgröße u.ä.
6. Objektbasierte Maße
7. Historiemaße
8. Prozessmaße
9. Mikroprozessbetrachtung
10. Stressfaktoren
11. Paarprogrammierung
12. Persönlicher Softwareprozess
13. Root Cause Analysis
14. Programmiertipps
15. Statische Analyse
16. Analyse v. Defektbehebungen
17. Analyse v. Änderungen

Viel Spaß!

Sebastian Jekutsch
Raum 008
Tel 030 / 838-75239
http://www.inf.fu-berlin.de/~jekutsch