

# Seminar

## **Ursachen und Vermeidung von Fehlern in der Softwareentwicklung**

Prof. Dr. Lutz Prechelt, Sebastian Jekutsch  
Freie Universität Berlin, Institut für Informatik  
<http://www.inf.fu-berlin.de/inst/ag-se/>

- Motivation
- Themen
- Organisation
- Kontakt

# Aufwände für Testen und Debugging

- Grob 60% sind Entwicklungskosten, 40% sind Testkosten

## Waterfall model



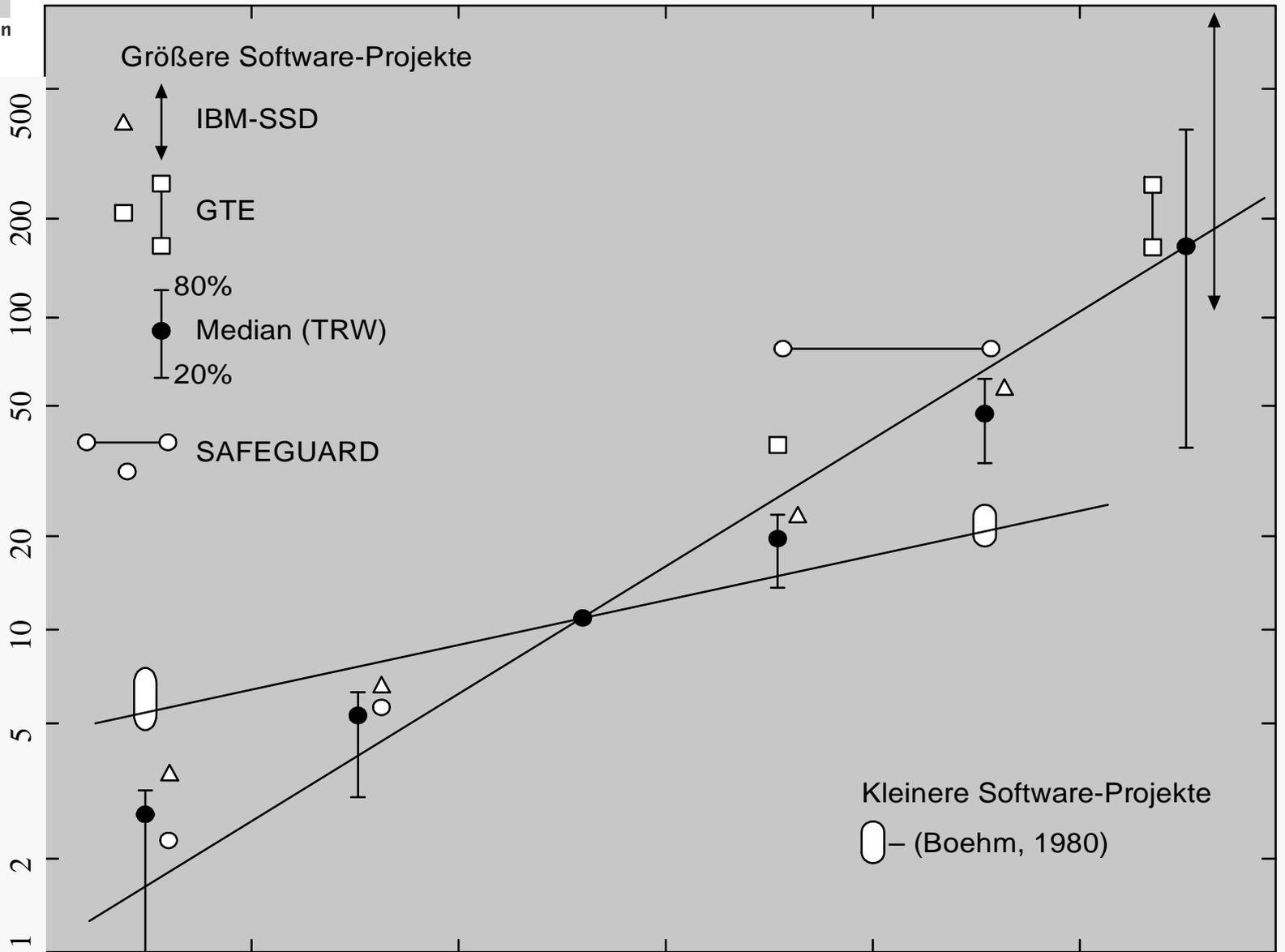
## Iterative development



© Sommerville 2004

# Späte Defektentdeckung kostet besonders viel

Relative Kosten für Fehler finden und beheben



Phase: Defintion    Entwurf    Implement.    Tests    Abnahme    Betrieb

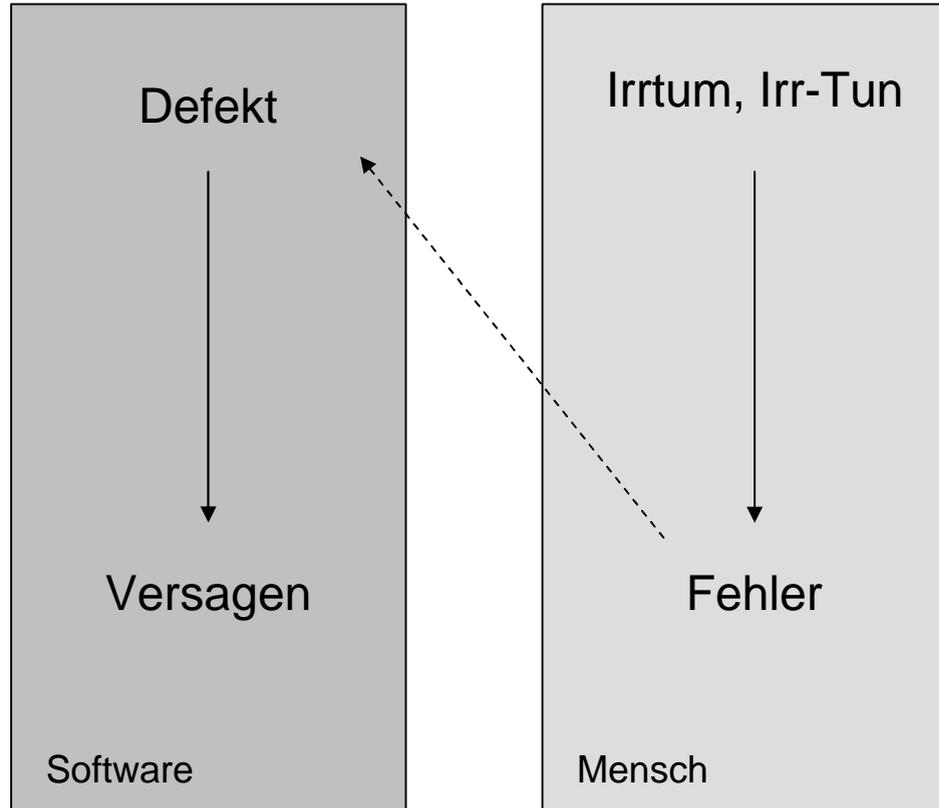
- Testen = absichtliches Verursachen von Versagen
- Debugging = Suchen nach ursächlichen Defekten
- Erst Defekte einbauen,  
dann Defekte suchen,  
dann Defekte wieder ausbauen.

⇒ alles Symptombekämpfung

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

*Ursache / Zustand*

*Wirkung / Ereignis*

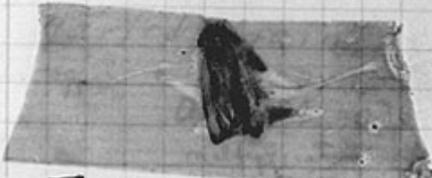


# Begriff „Bug“?

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

92

9/9

0800 Antam started  
 1000 " stopped - antam ✓  
 13<sup>00</sup> (032) MP-MC ~~1.982647000~~ { 1.2700 9.037 847 025  
 (033) PRO 2 2.130476415 ~~2.130476415~~ } 9.037 846 995 conduct  
 conduct 2.130676415 4.615925059(-2)  
 Relays 6-2 in 033 failed special speed test  
 in Relay " 10,000 test.  
 Relays changed  
 1100 Started Cosine Tape (Sine check)  
 1525 Started Multi-Adder Test.  
 1545  Relay #70 Panel F  
 (moth) in relay.  
 First actual case of bug being found.  
~~1630~~ 1630 Antam started.  
 1700 closed down.

Relay  
2145  
Relay 3376

# 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?

Nun zu Organisatorischem...

- JedeR bekommt ein Thema (~2 Papers + Recherche)
- Vortrag darüber etwa eine Stunde
- Ausarbeitung passender Länge (10-20 Seiten)
- Zwei Begutachtungen des Schriftlichen
- Zwei Begutachtungen der Vorträge

⇒ benoteter Schein

- Warnung:  
Ein Seminar ist aufwändiger als man zunächst glaubt!

- Heute: *Termin*festlegung, Themenauswahl
- Demnächst: Ausgabe der Quellen, zufällige Vergabe der Begutachtungsaufgaben
- bis Februar: Besprechung der Gliederung mit dem Seminarleiter
- 2 Wochen vor *Termin*: Abgabe an Gutachter
- bis *Termin*: Begutachtungen
- *Termin*: Blockseminar findet statt
- April: Abgabe Ausarbeitung, Scheinvergabe

- Seminar am Block innerhalb einer Woche
- Mögliche Wochen:
  - 7. März – 12. März
  - (14. März – 19. März)
  - 4. April – 8. April
  - Zwei halbe Wochen
- Eckdaten:
  - WiSe-Ende: 19. Februar
  - Klausur SWT: 22. Februar
  - SoSe-Beginn: 11. April
  - Karfreitag/Ostern: 25./27./28. März

# Kalender (mit Schulferien Berlin)

# 2005

	Januar					Februar					März					April					Mai					Juni					
KW	1	2	3	4	5	5	6	7	8	9	9	10	11	12	13	13	14	15	16	17	17	18	19	20	21	22	22	23	24	25	26
Mo	3	10	17	24	31	7	14	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27					
Di	4	11	18	25	1	8	15	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28					
Mi	5	12	19	26	2	9	16	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29					
Do	6	13	20	27	3	10	17	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30					
Fr	7	14	21	28	4	11	18	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24						
Sa	1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25					
So	2	9	16	23	30	6	13	20	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26					
	Juli					August					September					Oktober					November					Dezember					
KW	26	27	28	29	30	31	32	33	34	35	35	36	37	38	39	39	40	41	42	43	44	44	45	46	47	48	48	49	50	51	52
Mo	4	11	18	25	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	5	12	19	26					
Di	5	12	19	26	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27					
Mi	6	13	20	27	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9	16	23	30	7	14	21	28					
Do	7	14	21	28	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	17	24	1	8	15	22	29					
Fr	1	8	15	22	29	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30				
Sa	2	9	16	23	30	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	3	10	17	24	31				
So	3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25					

Schulferien Feiertage in Berlin (ohne Gewähr)

© www.agentur-lindner.de

- (Open-Source)
- (Agile Methoden)
- Ausgewählte Beiträge zum Software Engineering
  - Ausgewählte Anwendungen der Softwarearchäologie
  - Software aus Komponenten
  - Beobachtung des Kodierprozesses
  - Dokumentation von Komponenten und Bibliotheken
  - Kognitive Modelle des Programmierens
  - Agile Methoden
  - ... beliebige andere Themen
- jeweils Donnerstag 16-18 Uhr im Raum 046  
(morgen fällt allerdings aus)

1. Wahre Geschichten
2. Historie über Defekte
3. Defektklassifizierungen
4. Defektabschätzung 1
5. Defektabschätzung 2
6. Defektabschätzung 3
7. Denkfallen
8. Menschliches Versagen
9. Arbeitsumgebung
10. PSP
11. Root Cause Analysis 1
12. Root Cause Analysis 2
13. ODC
14. Cleanroom
15. Programmiersprachen
16. Semi-formale Methoden
17. Programmiertipps
18. Anti-Patterns
19. Archäologie 1
20. Archäologie 2

# Viel Spaß!

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