

SI-SE 2002

Testen in der Softwareentwicklung

Fachtagung mit Tutorien
Zürich
7. und 8. März 2002

Prof. Dr. Andreas Spillner, Hochschule Bremen
Prof. Dr. Martin Glinz, Universität Zürich

NASA - Mariner 1: Codierfehler

Der NASA ging eine auf dem Weg zur Venus
befindliche Raumsonde Mariner 1 verloren

4 Minuten nach dem Start durch Selbstzerstörung, 22.6.1962

DO 10 I=1,3 (Definition einer Schleife)

...
10 CONTINUE

fehlerhaft codiert wurde

**DO10I=1.3 (Zuweisung des Werts 1.3
an Variable DO10I)**

...
10 CONTINUE

NASA - Mariner 1

Nur ein FORTRAN-Problem? Nein!

C FOR (i=1; i<=3; ++i);
 f(i);

(1 mal mit i=4)

Pascal FOR i=1 TO 3 DO;
 f(i);

(1 mal mit i=?)

Modula 2 FOR i=1 TO 3 DO;
 f(i)
 END;

(richtig)

H. Klaeren:
Probleme des Software-Engineering.
Informatik Spektrum (1994) 17: 21-28

© Prof. Dr. A. Spillner 2002

3

Gemini V: Spezifikationsfehler

Die bemannte Raumkapsel *Gemini V* verfehlte ihren Landeplatz um 160 Kilometer

Beim Programm für die Landung hatte man die Rotation der Erde um die Sonne nicht berücksichtigt!

© Prof. Dr. A. Spillner 2002

4

Äquatorüberflug: Dokumentationsfehler

US Air Force, Programm zur Raketensteuerung:

Aus Speicherplatzmangel wurde beim Flug einer Rakete über den Äquator die Flugkoordinaten nicht neu berechnet, sondern nur das Vorzeichen geändert

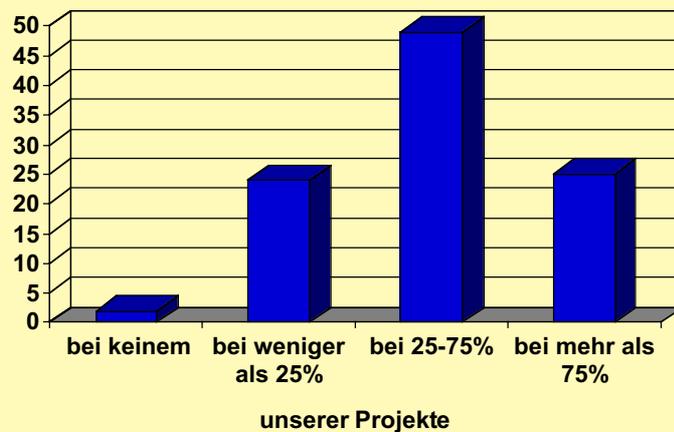
Dadurch drehte sich die Rakete beim Überflug des Äquators zwar um die eigene Achse, was allerdings niemanden störte

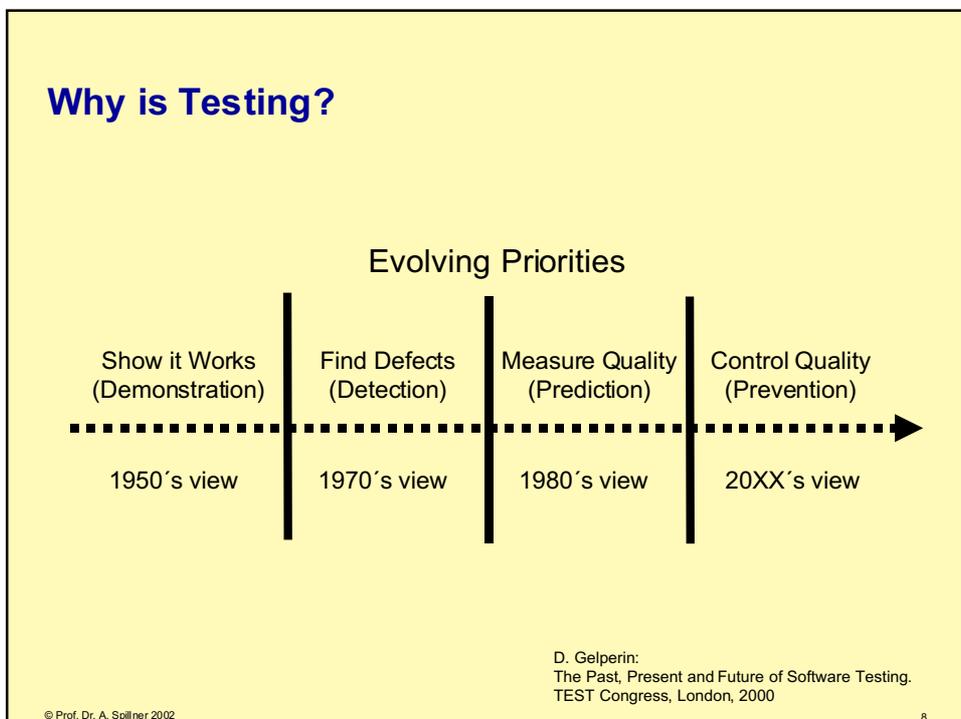
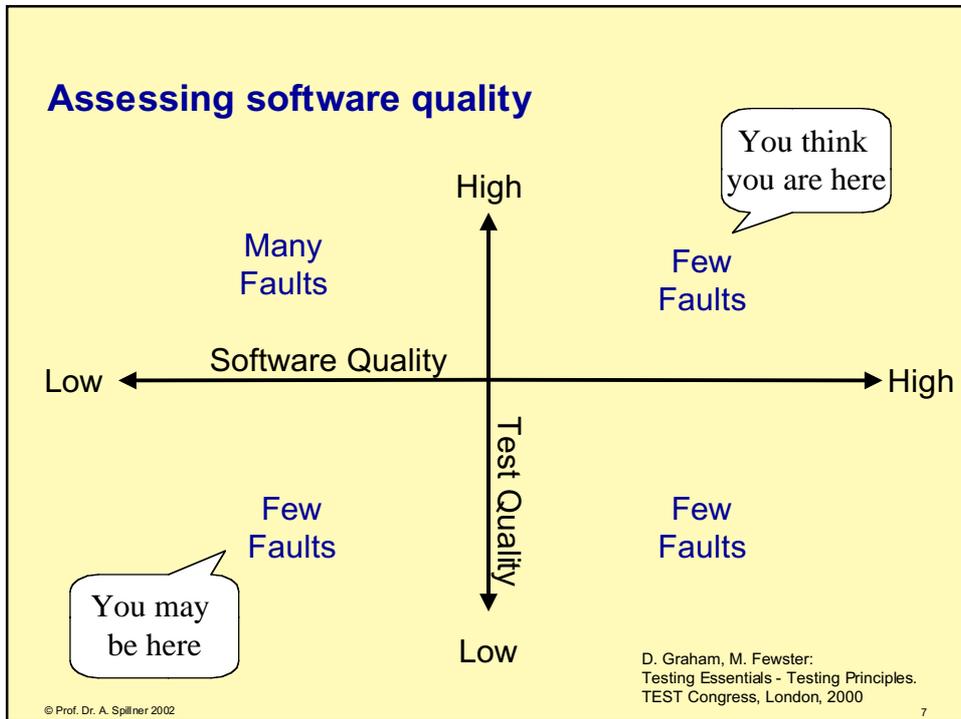
Programm wurde unverändert in den Autopiloten des Jäger F-18 übernommen

Bei Überflug des Äquators drehte sich die Maschine auf den Kopf!

Qualität muss oft zurückstehen

Wie oft ist Qualität ein Zugeständnis an die Geschwindigkeit, wenn Sie Geschäfts- und IT-Projekte entwickeln?





Testziele

- **Nachweis der Qualität**
durch Ausführen von Software
 - **Verifikation**
der Korrektheit auf Grundlage der Spezifikation
"Das System **richtig entwickelt**"
 - **Validierung**
des System relativ zu den Kundenanforderungen
"Das **richtige System** entwickelt"
- **Fehler finden**
bevor sie Schaden in Produktion anrichten
- **Regressionstest**
als Schutz vor unbeabsichtigten Veränderungen

Test - Aufwand

Erfahrungswerte (großes erfolgreiches Softwarehaus)

- Für Großprojekte gilt:
Implementierung + Test 40%, Integration 8%
des Gesamtaufwandes
- Beispielprojekt A:
gesamter **Testaufwand 1,2 mal Implementierungs-Aufwand**, davon
mehr als **2/3 im Modultest**
- Beispielprojekt B:
Systemtest (51,9%) +
Konzept (26,7%) + Implementierung (21,4%) = 100%

Test - Aufwand

Erfahrungswerte (großes erfolgreiches Softwarehaus)

- testintensive Projekte
(z.B. Jahr 2000 Umstellungen, Reengineering Projekte)
Testaufwand bis zu 80%

Erfahrung bei einem Kunden

- 20%-40% des Prozesses für Test
(Prozess = Analyse bis Übergabe an die Produktion)
- 1h pro fachlichen Testfall
(Erstellen, Ausführen, Dokumentation)

Vortragsthemen

- Testen im Softwareentwicklungsprozess
- To Test or Not To Test -
Wissensmanagement im Testprozess
- Testen und UML (Unified Modelling Language)
- Erfahrung und Tools: die Erfolgskriterien für e-Tests
- Testen im E-Business-Projekt -
geht das überhaupt bei Time-To-Market?
- Verbesserungen des Entwicklungsprozesses
durch minimale Testaktivitäten
- Automatisierung - die Lösung aller Testprobleme?
(Podium)

Ausstellung

6 Firmen

Consulting, Schulung, Werkzeuge

- cmg
- ELCA
- Imbus
method park Software AG
- SQS AG
- radview Software