

Defektabschätzung

- Malgorzata Wojciechowska
 - Institut für Informatik
 - FU Berlin
 - 03.04.2006
-

1. "Quantitative Analysis of Faults and Failures in a Complex Software System"

Niclas Ohlsson, PhD
Dep. of Computer and
Information Science
at Linköping University
in Sweden



Norman E. Fenton
Professor of Computer Science
Computer Science Department,
Faculty of Informatics and
Mathematical Sciences
Queen Mary University of London

LOC	Release n	Release n+1
<1000	23	26
1001-2000	58	85
2001-3000	37	73
3001-4000	15	38
4001-5000	6	16
5001-6000	0	6
>6000	1	2
Total	140	246

- Die Defektdaten wurden während vier unterschiedlicher Phasen gesammelt:
 - Funktionstest (FT) } Testdefekte
 - Systemtest (ST) } Testdefekte
 - Die ersten 26 Wochen einer Anzahl lokaler Tests (SI) } Betriebsdefekte
 - Das erste Jahr des Betriebs (OP) } Betriebsdefekte

Zusammenfassung der entdeckten Anzahl von Defekten in jeder Testphase:

Release	pre-release faults		post-release faults	
	Function test	System test	Site test	Operation
n (sample size 140 modules)	916	682	19	52
n+1 (sample size 246 modules)	2292	1008	238	108

2. "The Top Ten List: Dynamic Fault Prediction"

Richard C. "Ric" Holt
Professor,
Software Engineering,
School of Computer Science,
University of Waterloo



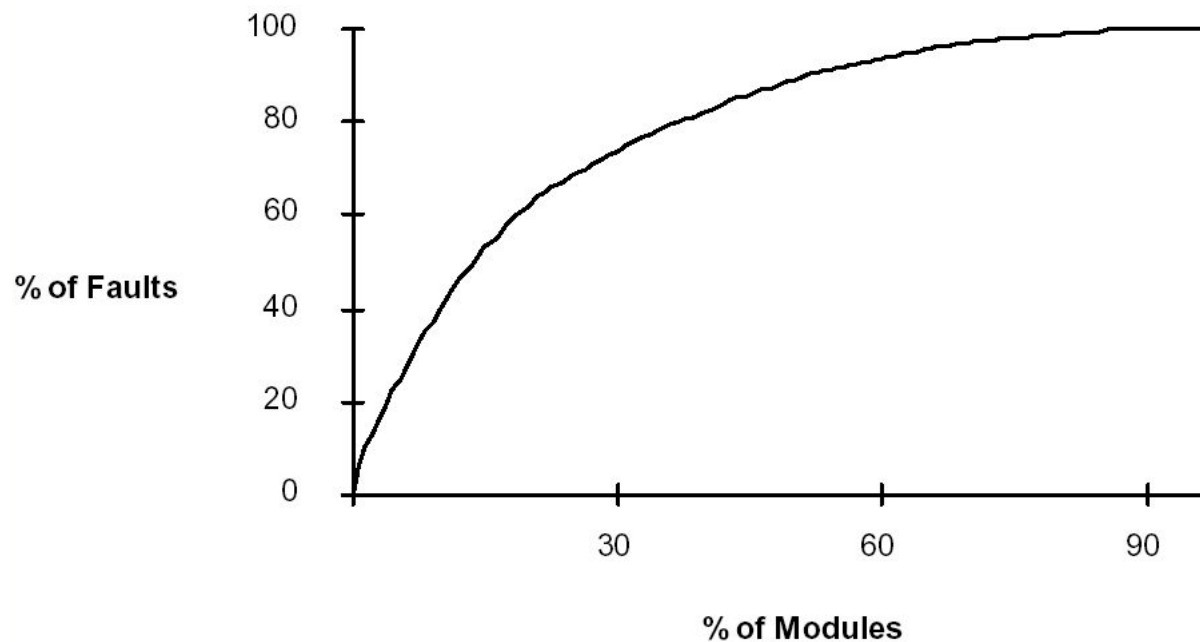
Ahmed E. Hassan
PhD Thesis,
School of Computer Science,
Faculty of Mathematics,
University of Waterloo

Application Name	Application Type	Start Date	Subsys. Count	Faults	Prog. Lang.
NetBSD	OS	21 March 1993	393	2451	C
FreeBSD	OS	12 June 1993	182	3264	C
OpenBSD	OS	18 Oct 1995	401	1015	C
Postgres	DBMS	9 July 1996	104	1401	C
KDE	Windowing System	13 April 1997	167	6665	C++
Koffice	Productivity Suite	18 April 1998	259	5223	C++

- Hypothesen in Bezug auf das Pareto-Prinzip über Defekt - und Versagensverteilung

Das Pareto-Prinzip

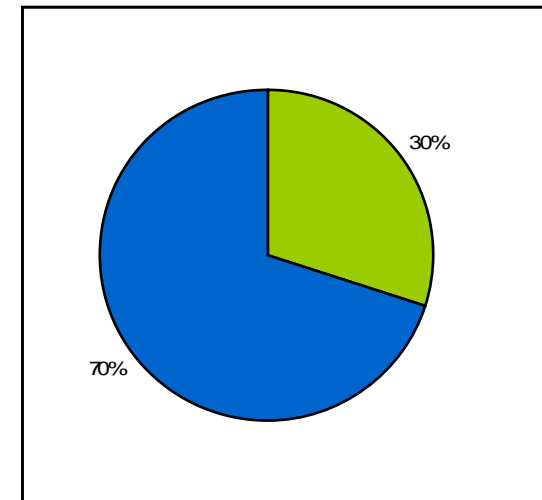
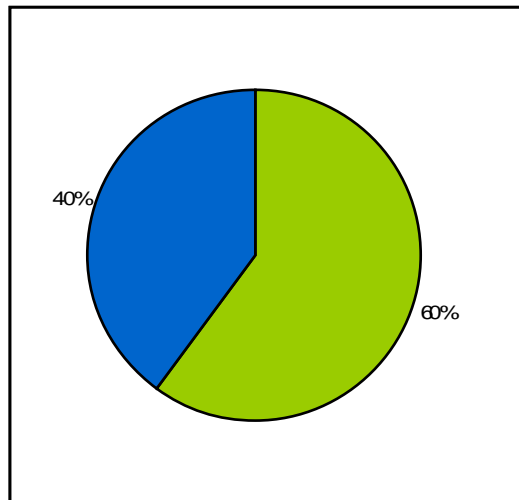
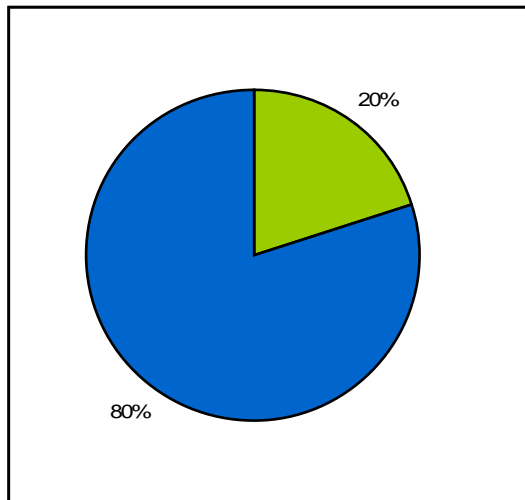
- **Hypothese 1a:** Die überwiegende Mehrheit der Defekte, die während der Vor-Veröffentlichungstests entdeckt werden, konzentriert sich in einigen wenigen Modulen.



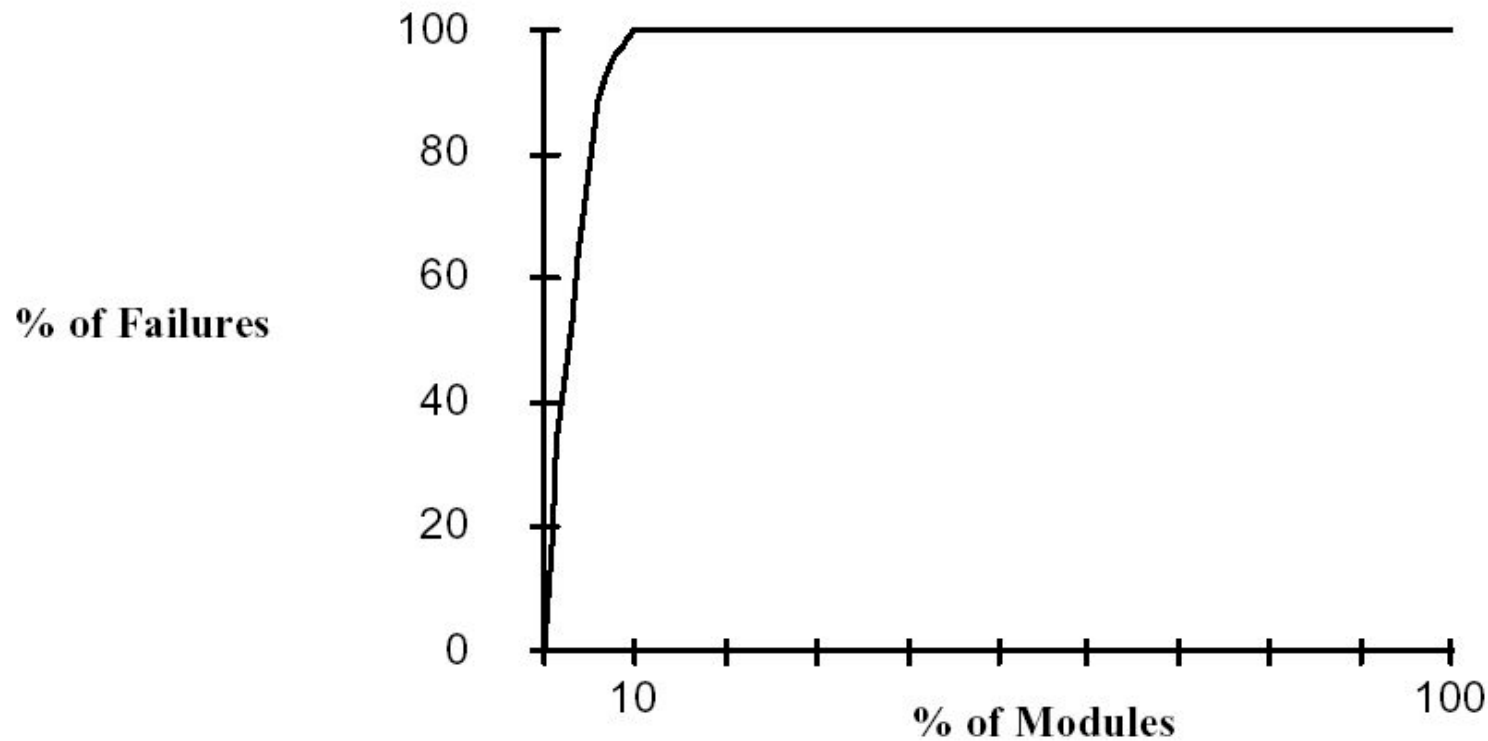
Das Pareto-Prinzip

- **Hypothese 1b:** Die Module, in denen sich die Defekte, die während der Vor-Veröffentlichungstests entdeckt werden, konzentrieren, machen gleichzeitig den größten Teil des gesamten Codes aus.

20% der **Module** beinhalten 60% der **Defekte** und machen 30% der **Systemgröße** aus



- **Hypothese 2a:** Eine kleine Anzahl von Modulen enthält die meisten Betriebsdefekte (gemeint sind Versagen wie man sie in Phasen SI und OP beobachtet hat).



- **Hypothese 2b:** Eine kleine Anzahl von Modulen enthält die meisten Betriebsdefekte, weil diese Module den größten Teil des Codes ausmachen.

Antithese:

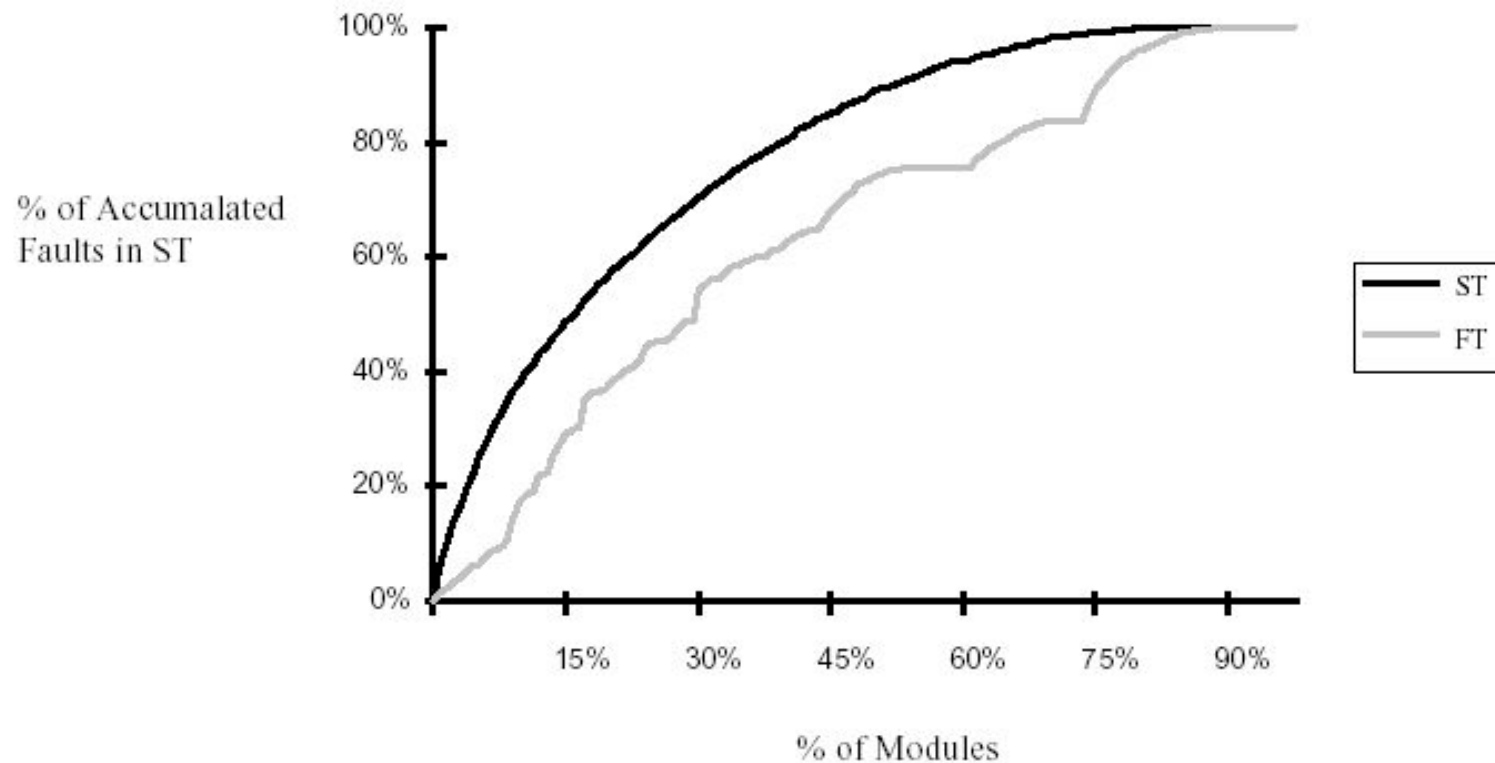
Die meisten Betriebsdefekte werden durch Defekte in kleinen Teilen des Codes verursacht.

Ergebnis:

100% von Betriebsdefekten befanden sich in Modulen mit nur 12% der Systemgröße

- Hypothesen in Bezug auf den Gebrauch von früheren Defektdaten dienen der Vorhersage von Defekten und Versagen (im Modullevel):

- **Hypothese 3:** Ein häufigeres Auftreten von Defekten in Funktionstests (FT) ist verknüpft mit einem häufigeren Auftreten von Defekten in Systemtests (ST).

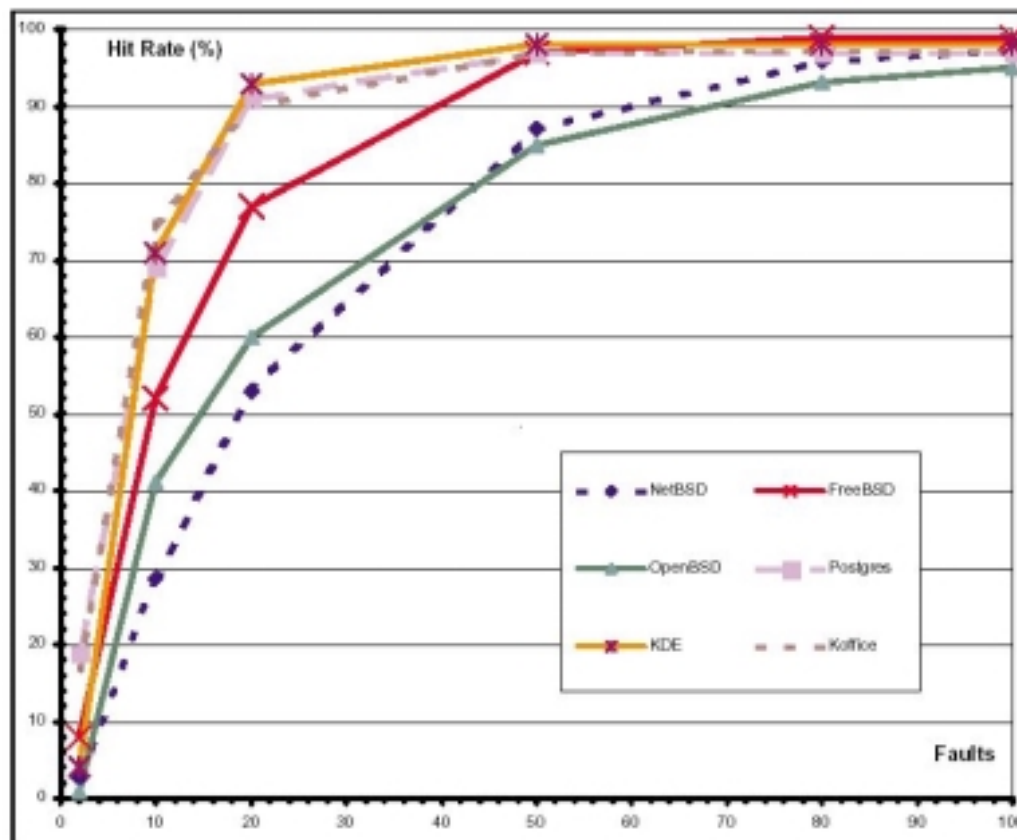


- **Hypothese 4:** Ein häufigeres Auftreten von Defekten während der Vor-Veröffentlichungstests ist verknüpft mit einem häufigeren Versagen im Betrieb.

- Hypothesen in Bezug auf Häufigkeit und Zeitspanne, in der Veränderung der Subsysteme vorgenommen werden:

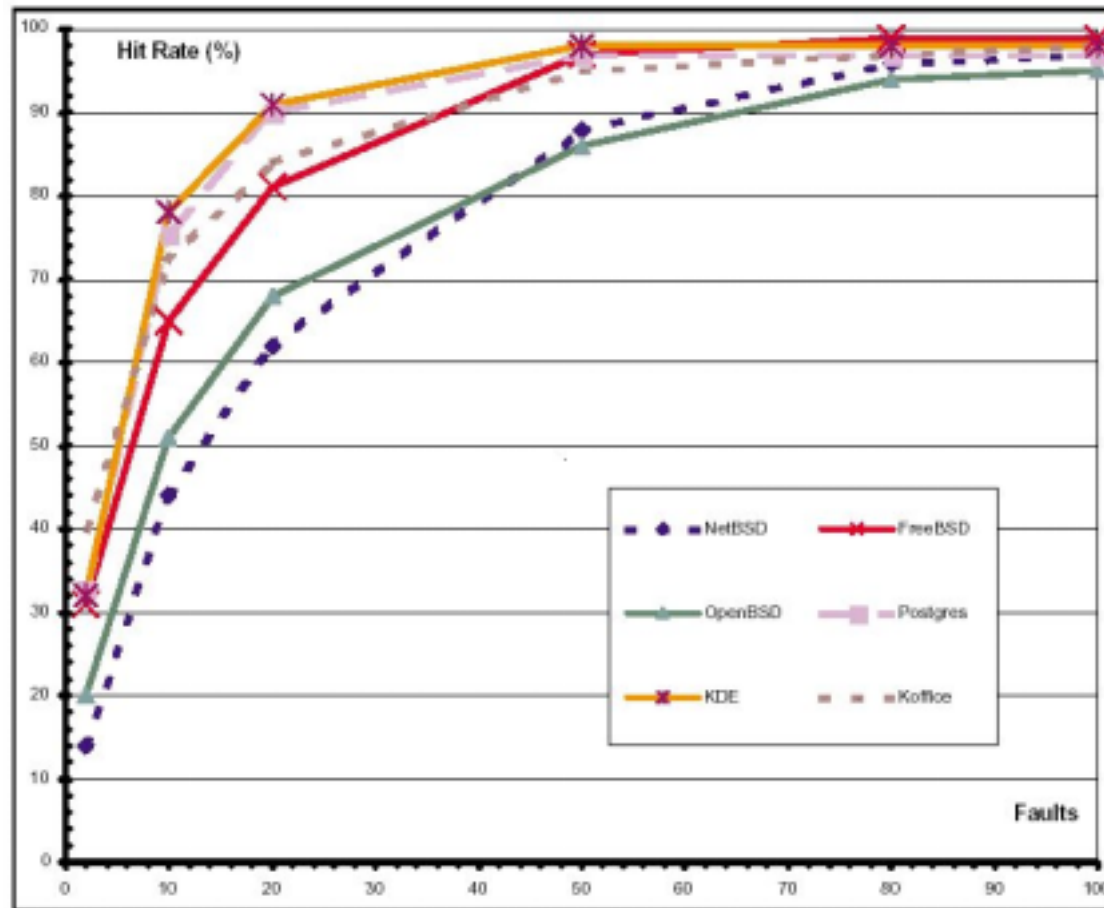
Häufigkeit und Zeitspanne der Veränderung der Subsysteme

- **Hypothese 5:** Je öfter die Subsysteme verändert wurden, desto höher ist die Wahrscheinlichkeit, dass sie Defekte beinhalten.



Häufigkeit und Zeitspanne der Veränderung der Subsysteme

- **Hypothese 6:** Erst vor kurzem veränderte Subsysteme weisen eine erhöhte Defektanfälligkeit auf.



- Hypothesen über Größenmaße in der Defektvorhersage

- **Hypothese 7:** Aus einfachen Größenmaße (LOC) des Moduls lässt sich eine Vorhersage über die Defektanfälligkeit ableiten.

- **Hypothese 8:** Das Komplexitätsmaß ist ein besserer Indikator für defekt- und versagensanfällige Module als ein einfaches Größenmaß.

- Hypothesen zur Beziehung zwischen Defektdichte, Qualität und Benchmarking-Daten:

- **Hypothese 9:** Es ist davon auszugehen, dass die Defektdichte in den Testphasen und der nachfolgender Betriebsphase bei Hauptversionen eines Softwaresystems nahezu konstant bleibt.

	FT	ST	SI	OP
Rel n	3.49	2.60	0.07	0.20
Rel n+1	4.15	1.82	0.43	0.20

- **Hypothese 10:** Softwaresysteme, die unter vergleichbaren Bedingungen produziert werden, weisen in der Regel eine etwa gleiche Defektdichte sowohl in der Test- als auch in der Betriebsphase auf.

	Pre-release	Post-release	All
Rel n	6.09	0.27	6.36
Rel n+1	5.97	0.63	6.60

Vielen Dank!