

Learning from past mistakes - Using reported faults to improve test efficiency

Lars-Ola Damm

Ericsson and Blekinge Institute of Technology



Outline



- Background and motivation
- Fault analysis and test strategies
- Success factors and considerations
- Summary



Background (1/2)

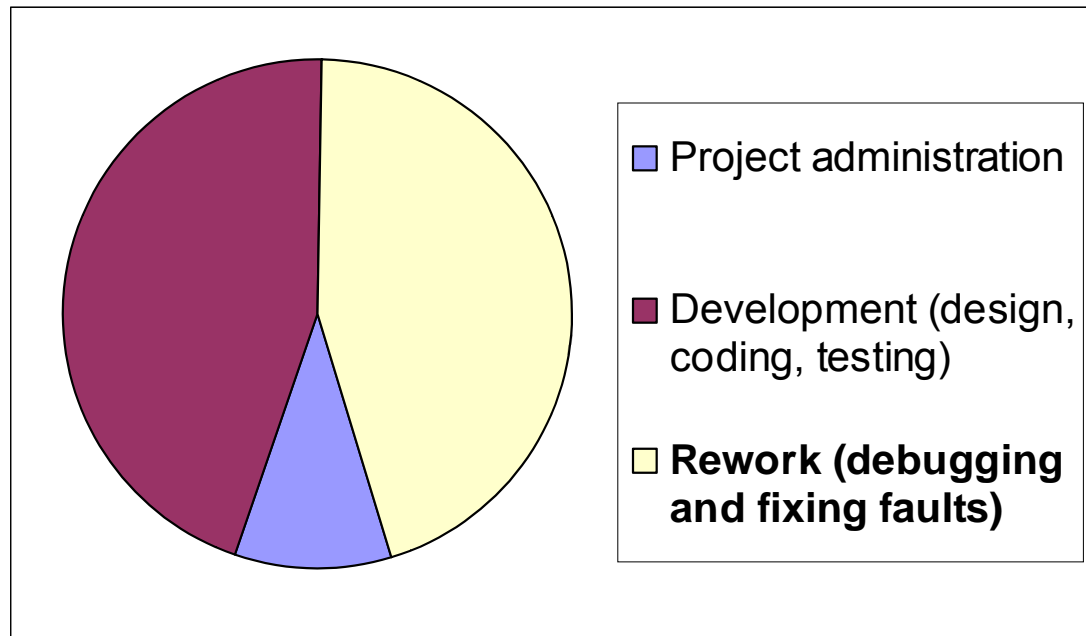
Who am I?

- Conduct research as an industrial PhD student
 - Part of the BESQ project
 - Duration
 - Started the research 1/1 2003
 - PhD defense planned for May 2007
 - Environment
 - Multimedia products for mobile networks
 - Size of projects: ~30-3000 man-months
 - Dev. process: incremental/iterative, going towards agile

Background (2/2)

Research focus

A typical software development project



- Research title:
 - “Early and cost-effective software fault detection”

Why measure? (1/2)

Quantification

Input to decisions

- “950 installation faults were found in the last project. This cost us about 2 MSEK and two weeks longer lead time. If you give me 2 MSEK to re-build the installation system, I need 2 MSEK for the implementation”
installation system, every future project will cost 1MSEK less”

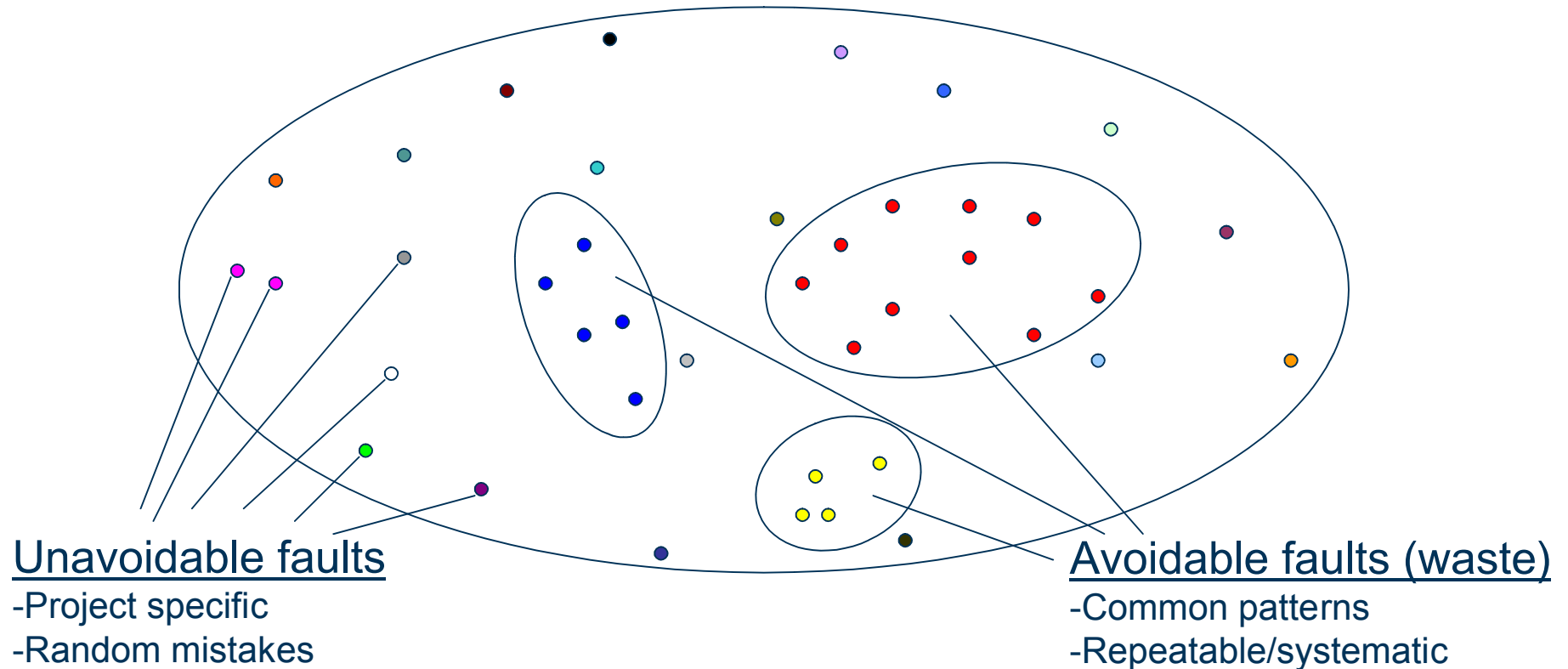


Product owner

Will we benefit enough from doing this investment?

Why measure? (2/2)

Fault measurements



- Fault analysis is about finding and quantifying the avoidable faults

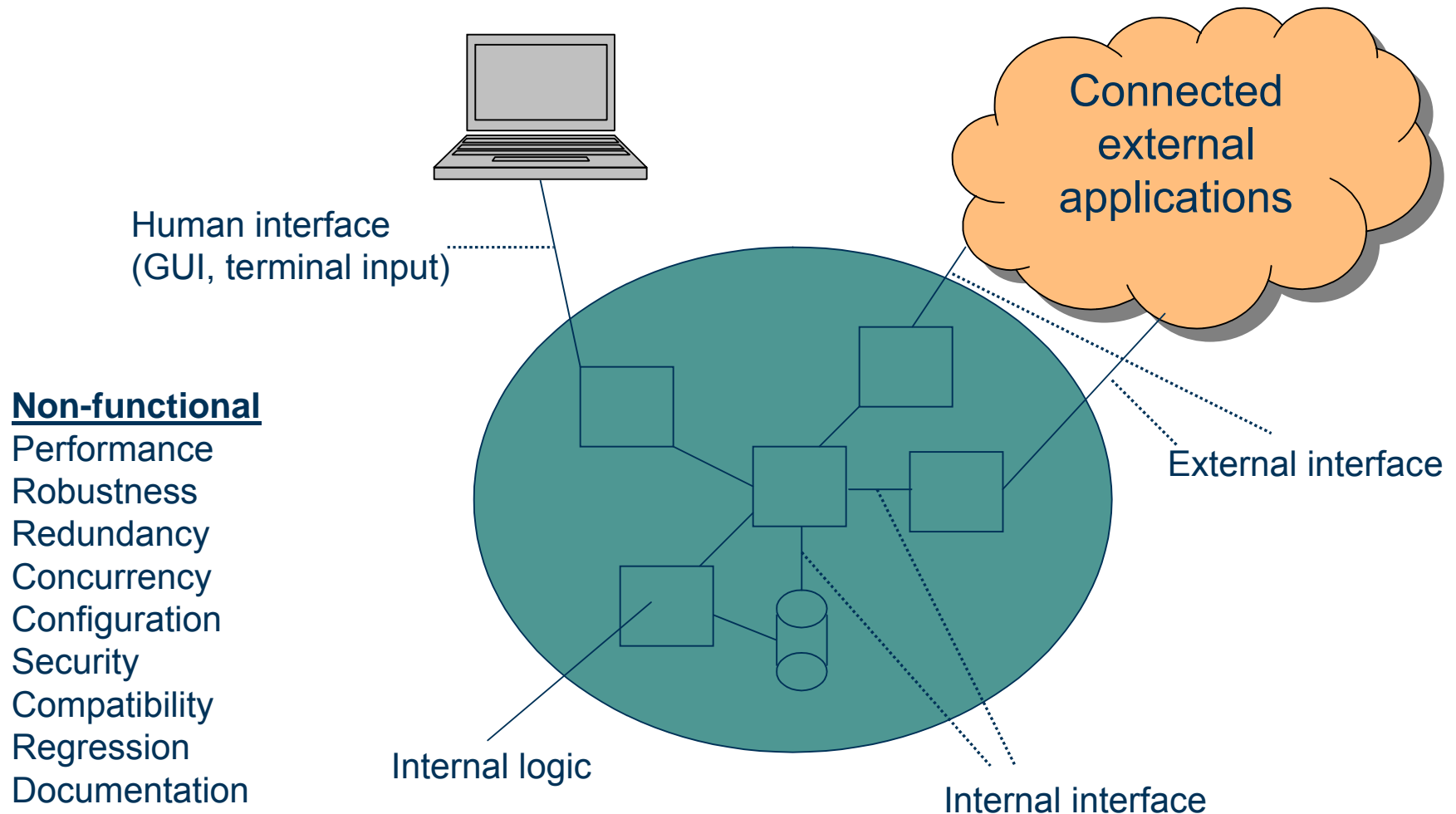
Fault Analysis and Testing (1/3)



What does fault analysis have to do with testing?

Fault Analysis and Testing (2/3)

Possible test areas (example)



Fault Analysis and Testing (3/3)



Test strategy (example)

Test area	Unit Test	Integration/application Test	System Test
Internal logic	X		
Internal interface	X	X	
External interface		X	
Human interface (GUI, terminal)	X	X	
Load/performance			X
Robustness	X		X
Redundancy (e.g. backup, restore)			X
Security	X	X	X
Compatibility			X
Configuration/installation		X	X
Regression	X	X	X
Documentation		X	X

Reported faults can tell you if the test strategy works as intended or not →



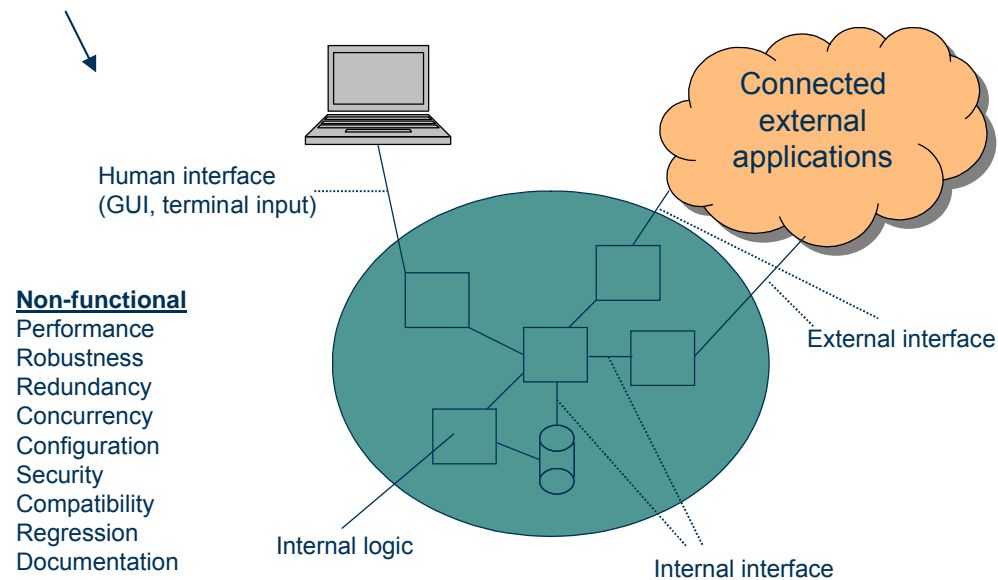
How to use fault analysis to evaluate the test strategy – Two examples

Test strategy evaluation

Example 1 – Fault trigger measurements (1/2)

- A fault trigger determines which type of test activity that *should* identify a fault
 - For example, robustness faults should belong to the trigger ‘robustness’

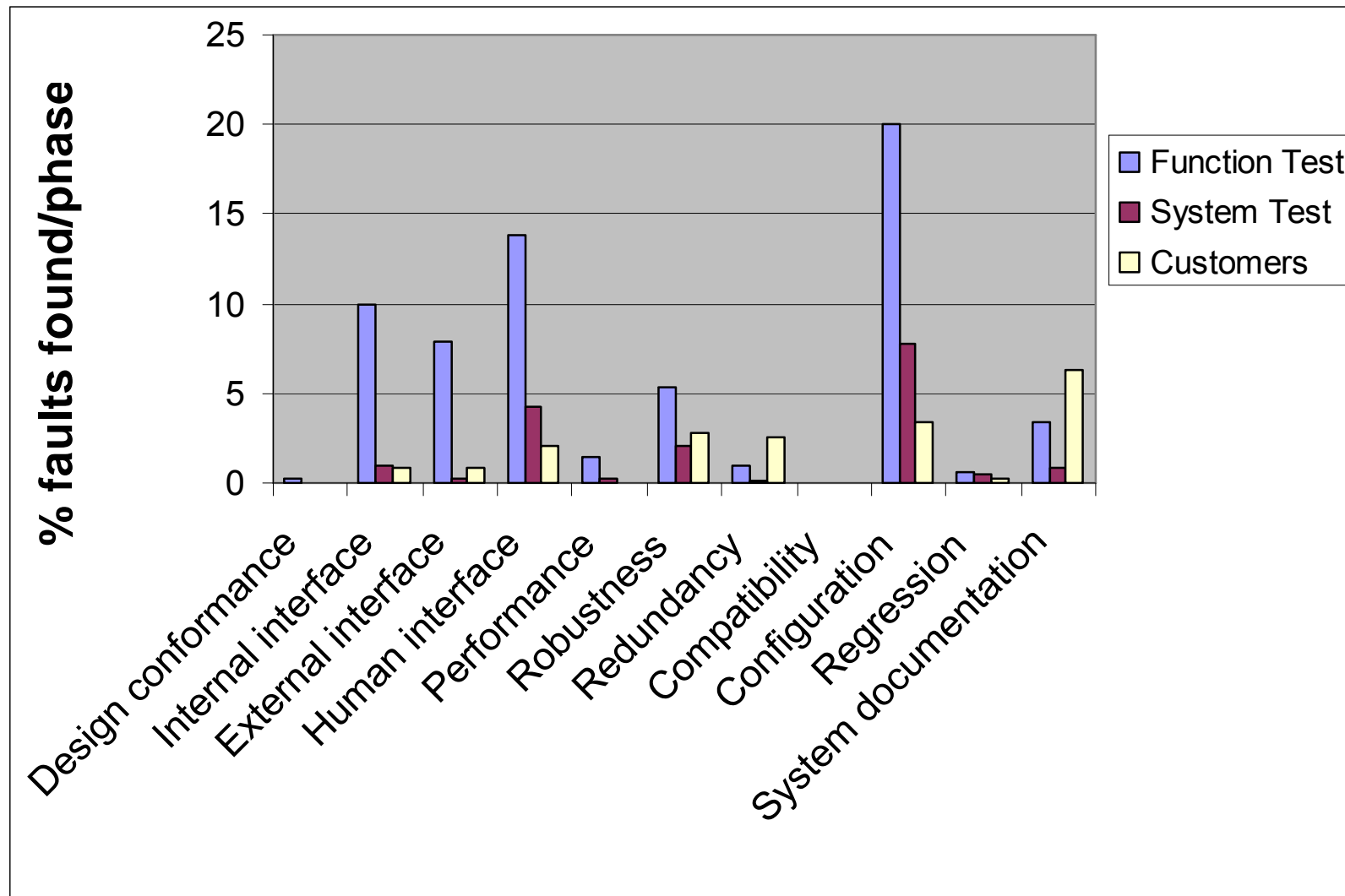
Test area \approx fault trigger





Test strategy evaluation

Example 1 – Fault trigger measurements (2/2)

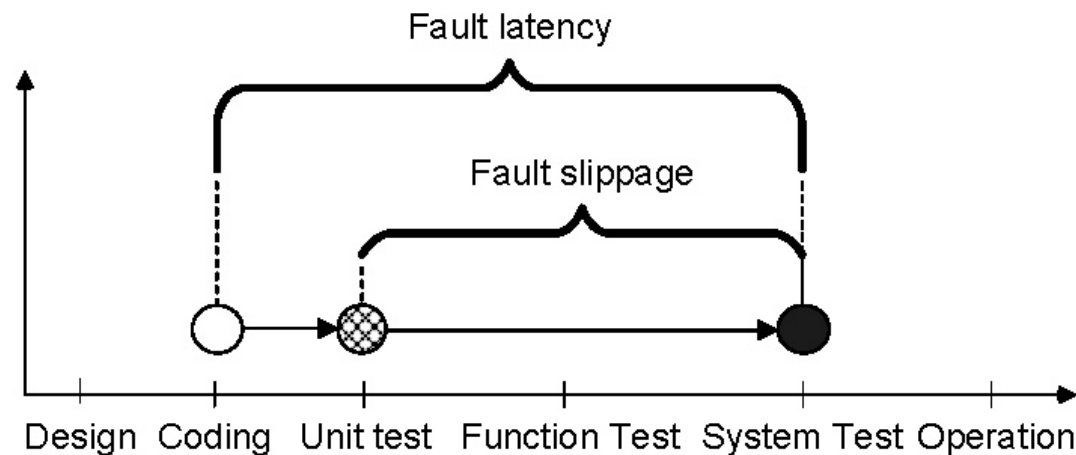
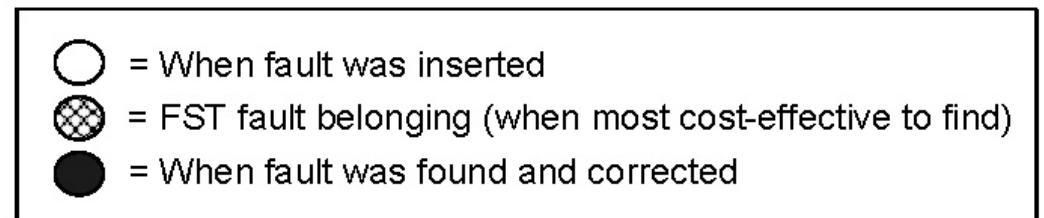


Test strategy evaluation

Example 2 – Faults-slip-through measurements (1/2)

■ Definition

- A fault-slip-through is a fault that should have been found in an earlier phase
- The test strategy defines when which faults should be found

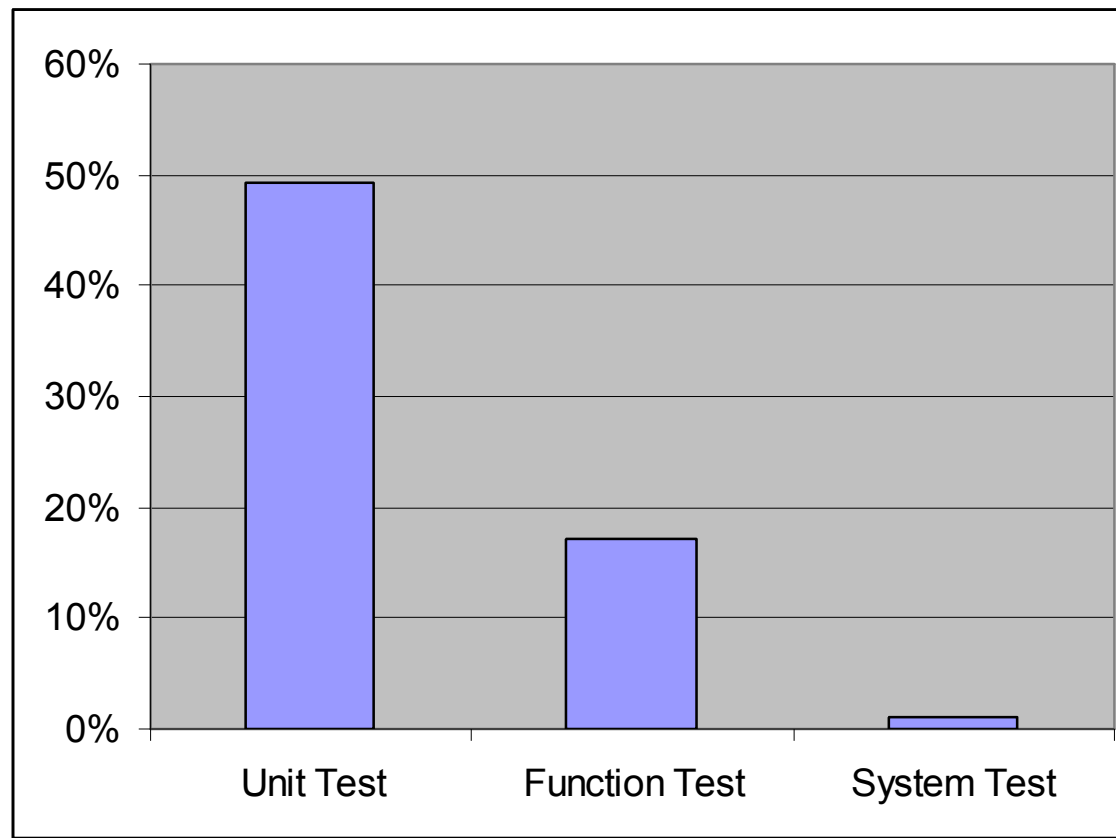




Test strategy evaluation

Example 2 – Faults-slip-through measurements (2/2)

% fault slippage from different phases





What else can we learn from fault analysis?

- Many faults in legacy/third-party software?
- Many rejected fault reports?
- No. faults/component
- Progress/project control (trends of no. reported faults, open fault reports)
- ...

Considerations/success factors



- Considerations
 - Measurements are useless without a context
 - Connect measurements to business goals
 - Never use measurements to evaluate individuals
 - Avoid rewards connected to fault measurements

- Primary success factors
 - Having a culture of continuous improvements
 - The impact the measurements have on decision making
 - Requires visibility/transparency

Summary



- Learn from history (faults)
- Quantify to motivate investments
- Fault analysis can be used for test strategy improvement



Questions?

lars-ola.damm@ericsson.com

Visibility/tool support



Reported faults/week

