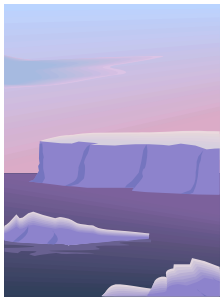
Defect-Flow Models

Part 4: Industrial Case Studies



Dr. Adam Trendowicz
adam.trendowicz@iese.fraunhofer.de

Ove Armbrust
ove.armbrust@iese.fraunhofer.de

Martin Kowalczyk
martin.kowalczyk@*iese.fraunhofer.de*

Michael Kläs
michael.klaes@iese.fraunhofer.de

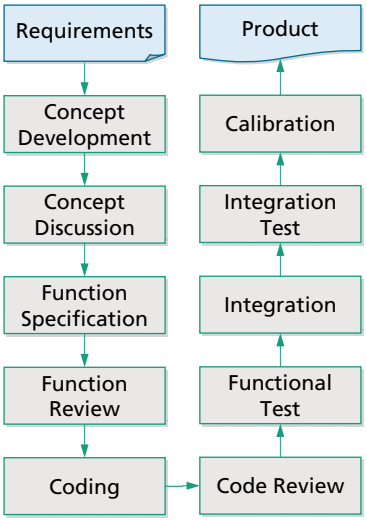
Fraunhofer IESE

Fraunhofer IESE

Part 4: Industrial Case Studies

- **Bosch GS**
- Allianz
- IBS
- Summary

Bosch GS – Context



- General Facts
  - Size: 900 employees responsible for software development
  - Domain: automotive
  - Business unit: Gasoline Systems (GS)
  - Products: electronic control units for gasoline engines
- Improvement Initiatives
  - CMMI-based
  - Measurement acknowledged as a key factor for successfully identifying and introducing software process improvements
- Waterfall-like software development lifecycle

Bosch GS – Motivation

Problems

- Rapid increase in functionality of software systems
- Rapid increase in complexity of software systems
- Decreased product quality
- Increased rework costs

Needs

- Effective quality assurance strategies required
- Effective measurement-based (quantitative) approaches needed
- Baselines need to be set up

Critical Questions

- What type and how many defects are injected per development activity/phase?
- What type and how many defects are found per QA activity/phase?
- Which QA activity is the most appropriate for finding defects of a particular type and/or injected in a particular development activity?

Bosch GS – Objectives

Goal: Maintain and improve customer satisfaction through high-quality products

- Continuously optimize the quality of the developed software
- Continuous process improvement and systematic quality management

Strategy: Support the derivation of improvement activities by providing transparency

- Establish measurement instruments for the QA process
  - Quantitative feedback to development process, QA activities, and products
  - Identification of defect injection and detection, defect slippage trends, and causes of defects
  - Information about effectiveness of QA activities
- Drive improvement activities
  - Derivation and prioritization of improvement activities
  - Evaluation of process changes and effectiveness of improvement initiatives
  - Provision of data for statistical process control

Process overview – Three cooperation projects

Basic Model Definition

- Develop defect-flow model by defining the defect attributes *injection* and *detection*
- Estimate/collect defect-flow data
- Determine qualitative QA strategy
  - Listing of defect classes that are either easy or hard to detect by each defect detection phase

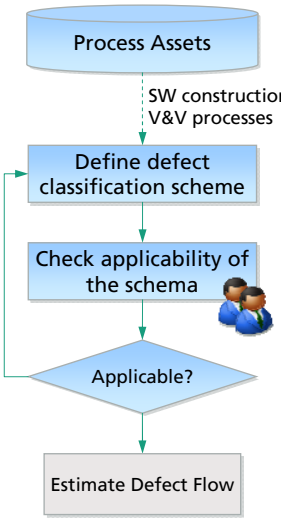
Detailed Model Definition

- Determine defect attributes and attribute values appropriate for the purpose of
  - Reducing defect injection rate
  - Increasing defect detection rate

Pilot Application

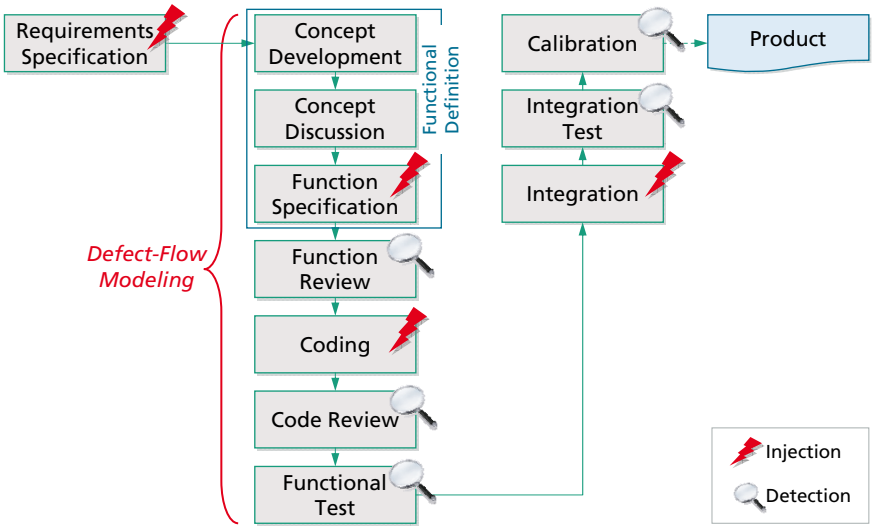
- Implement the defect classification scheme
- Collect data according to the scheme
- Focus on educating the developers in using the classification scheme
- Assess goodness of the scheme

Basic model definition – Developing defect-flow model



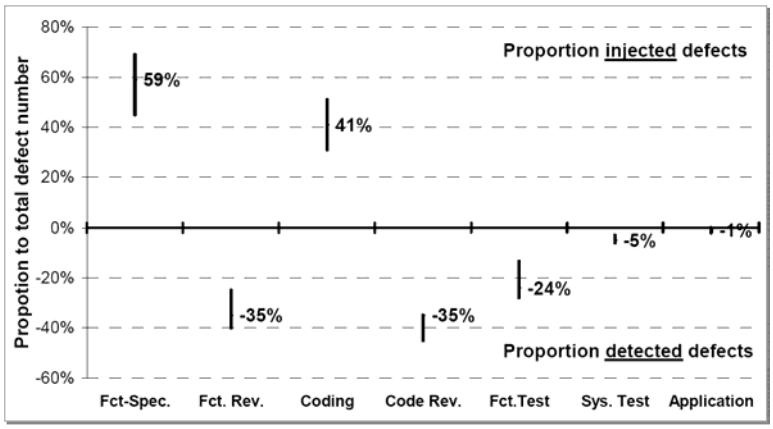
- Define defect classification scheme consisting of two attributes
  - *Injection*: In which phase was the defect injected?
  - *Detection*: In which activity was the defect detected?
- Use organizational process description
  - *Injection* attribute values are determined by constructive activities
  - *Detection* attribute values are determined by V&V (QA) processes
- Additionally, identify those phases from which no defect data can be collected
  - Typical example: debugging activity performed by the authors themselves
- Check the applicability of the resulting scheme with the developers

Defect injection and detection activities



Estimated defect flow

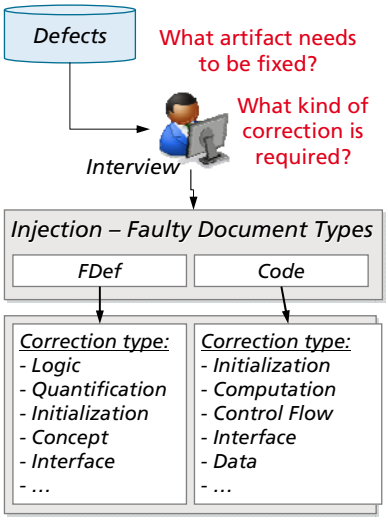
- Improvement potential: Analysis of defect flows allowed for identifying one defect class that was introduced early in the process but detected rather late (high cost of removal).



Estimated defect flow – Interpretation

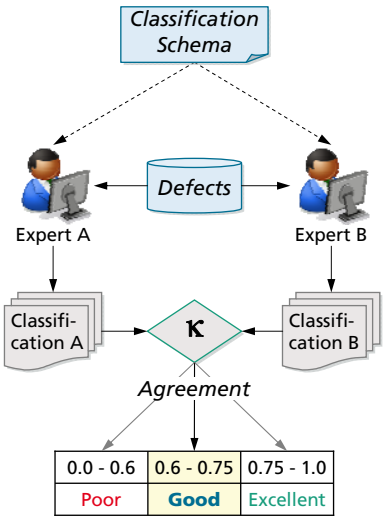
- For one V&V process, both defect counts and injection classification are available
- For inspection and testing processes, measurement data regarding defect counts are available
  - Injection classification was obtained through expert opinion
  - Developers were asked to estimate % of defects injected in the function specification and % of defects injected in software coding
- For one testing process, no systematic storage of defect numbers or classification was available
  - Total number of defects was estimated by testers as #tests multiplied by #defects per test
- The majority of defects are detected before the system test
- Distinction between function specification and software coding too coarsely grained to identify what type of defects should be detected by which defect detection activity

Detailed model definition



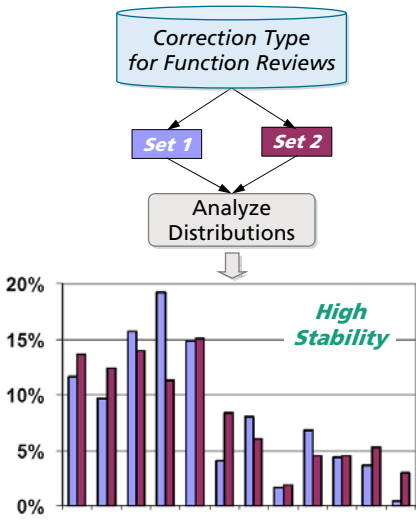
- Detailed classification uses HP scheme as reference and is based on experts
  - Developers relevant for particular injection phases and document types are involved
- Detailed model definition
  - For each injected defect, expert determines injection activity based upon the type of faulty document that needs to be fixed
  - For each document type, expert identifies types of corrections for removing the defect
- Defect classification for software documents was already available
- Correction types of functional specification needed to be defined

Validating defect-flow model – Reliability



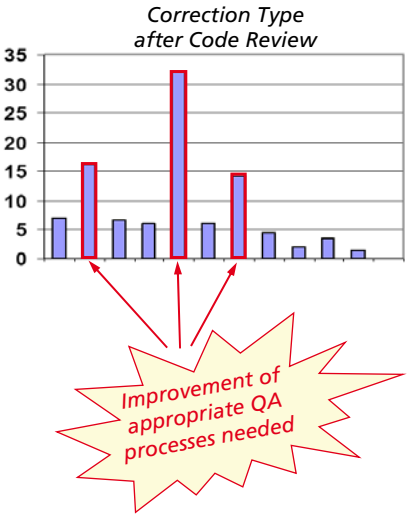
- Do different developers classify the same defects in the same way?
- Procedure
  - Let two developers classify the same set of defects
  - Compute Kappa coefficient to verify the level of agreement between the alternative classifications
- Result for FDEF correction type
  - Kappa = **0.65**
- Conclusions
  - good agreement
  - defect model reliable

Validating defect-flow model – Stability



- Is *correction type* distributed consistently for each instantiation of the same QA activity?
- Analyze defect distribution from the function review
  - Divide the set of function reviews into two groups of the same size
  - Determine the *correction type* distribution
  - Analyze differences in the distribution
- Results
  - ≤ 3% difference for 9 attributes
  - 8% difference for 1 attribute
- Conclusions
  - Distribution stable
  - Impact of potential process changes should be reflected reliably by changes in distribution of correction types

Validating defect-flow model – Improvement potentials



- What are the improvement potentials with respect to development and QA activities?
- Analysis of correction types for defects detected after code review
  - 3 particular types of defects frequently slip through the function review processes
- Conclusions
  - Early quality assurance processes require improvements
  - Improvement of function review process was initiated

Bosch GS – Summary

Benefits

- Quantitative management and control of the quality assurance process in the context of CMMi Level 4
- Ability to generate quantitative evidence
  - Status of the quality assurance process
  - Impact of process-spanning changes
- Ability to present quantitative evidence to management
- Involvement of developers and management in the pilot application helped to obtain support and sponsorship across all hierarchy levels for the introduction of DFM technology

Further steps

- Extend defect classification and defect-flow model for early and late phases of software development (4th cooperation project with Bosch GS)

Bosch GS – Lessons learned

- Facilitate positive attitude by talking about findings instead of defects
- Classify and measure defects using GQM paradigm
- Consider "correction type" as a central element of defect classification
- Provide appropriate tool support for data collection, analysis, and interpretation
- Motivate management and technical stakeholders
  - Estimate the current defect flow within the organization using available data and expert estimates
  - Provide initial defect classification and measurement results as soon as possible
  - Use appropriate visualization means
- Drawing reasonable conclusions from the defect-flow data requires stable processes

Part 4: Industrial Case Studies

- Bosch GS
- **Allianz**
- IBS
- Summary

Allianz – Context

Organization

- Allianz Life (part of Allianz)
  - 500 employees (350 application developers)
- Existing IT/software (1998)
  - 8500 software applications written in PL/I, C, Assembler
  - Running mainly on a Host environment (Client-Server architecture recently established)

Business Goal

- Improve customer satisfaction

Strategies

- Identifying and understanding the weaknesses of the current development processes by means of a measurement program
- Selecting and implementing appropriate techniques that are to overcome the detected weaknesses
- Evaluating the impact of the selected techniques by means of a measurement program

Allianz – Motivation and objectives

Problem

- 50% of defects identified in testing originate from early development phases
  - High rework costs (testing effort takes 30% of total development effort)
  - Poor communication and understanding between SW development departments

Solution

- Improve overall control of V&V processes
- Improve quality assurance activities in early stages of software development
  - Improving effectiveness of inspections by introducing Perspective-Based Reading (PBR) technique
  - Artifacts: requirements and design document
  - Use DFM for:
    - Characterizing current defect slippage (before introducing new technology)
    - Evaluating improvement in defect slippage after introducing new technology
- Improve communication between departments
  - Consistent defect measurement

Context of the pilot application

- Euro conversion project (300 person-months, several sub-projects)
  - Convert the amounts of insurance policies to the EURO currency
  - 250 applications (PL/I, C, Assembler)
  - 550 interfaces and 14 related information systems need to be adapted
  - Sub-project A
    - Effort: total of 38 person-months, 28 person-months for IT
    - Objective: adapt Euro conversion to investment subsidiary of Allianz Group
    - Team: 4 persons (each from IT and investment departments)
  - Sub-project B
    - Effort: total of 33 person-months, 22 person-months for IT
    - Objective: adapt Euro conversion to insurance subsidiary of Allianz Group
    - Team: 6 persons from IT department and 11 persons from insurance department

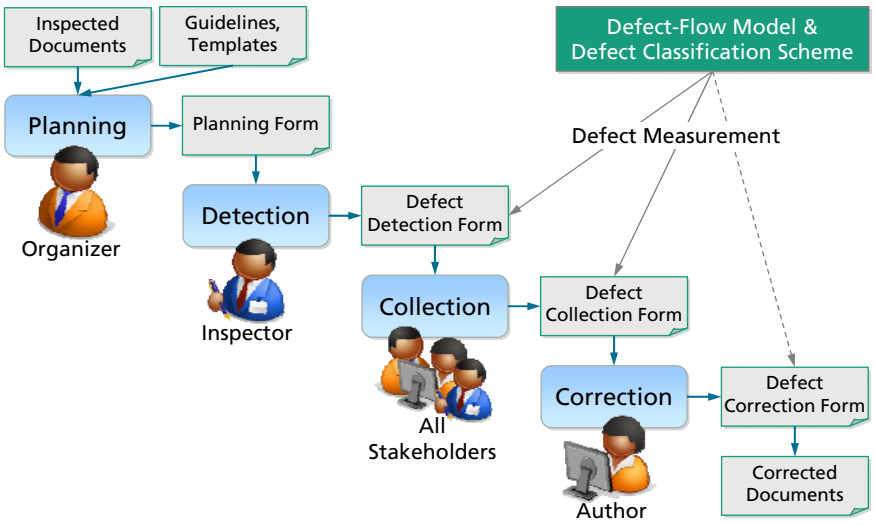
Use of DFM for perspective-based reading

Finding = Defect

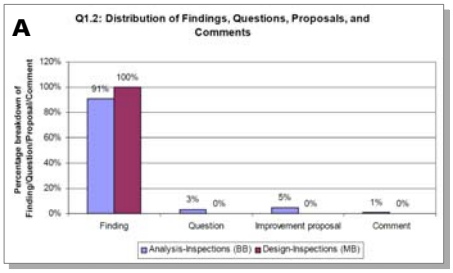
- In order to promote a positive atmosphere, the term "finding" was used to refer to issues raised during inspections instead of the term "defect"

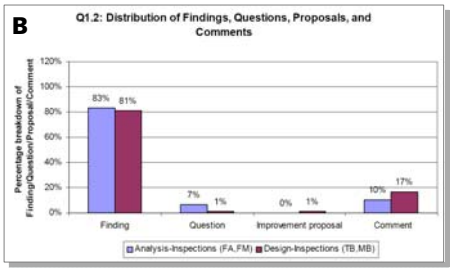
- Perspective-based Reading tells inspectors
  - What to look for during the inspection
  - How to inspect a software artifact under inspection for potential findings
- Classification of issues (inspection-specific; early development phases)
  - Class – type of issue
    - Finding, question (has to be investigated), improvement proposal, comment
  - Reference – documenting the finding refers to
    - Inspected product, other product, development process
  - Severity – importance of finding
    - Very critical, critical, interesting
  - Cause – the type of error that led to the finding (defect)
  - Impact – the negative consequences in case the finding has not been detected and fixed

Inspection process



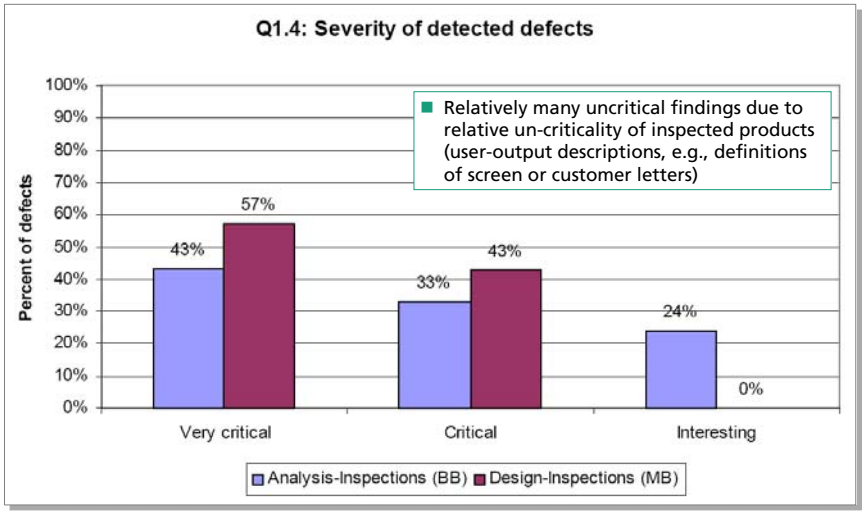
Type of issues



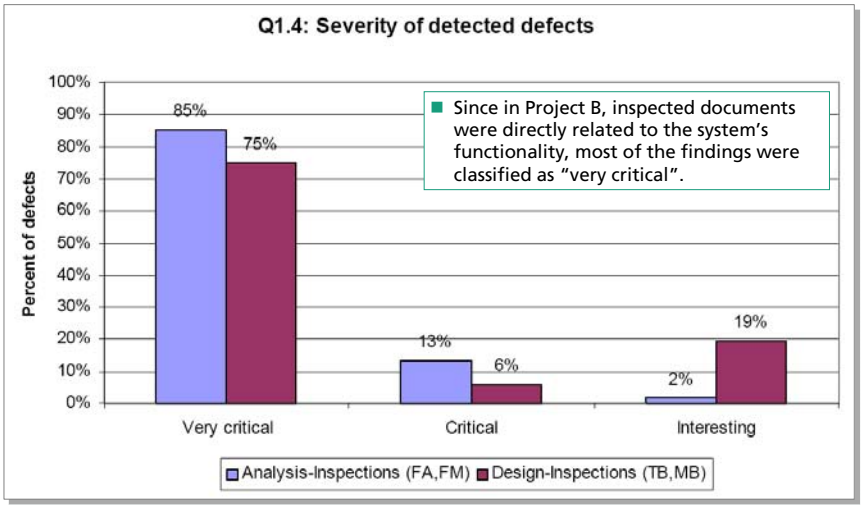


- Inspection could detect different types of issues, …
- …but inspection mainly detected defects

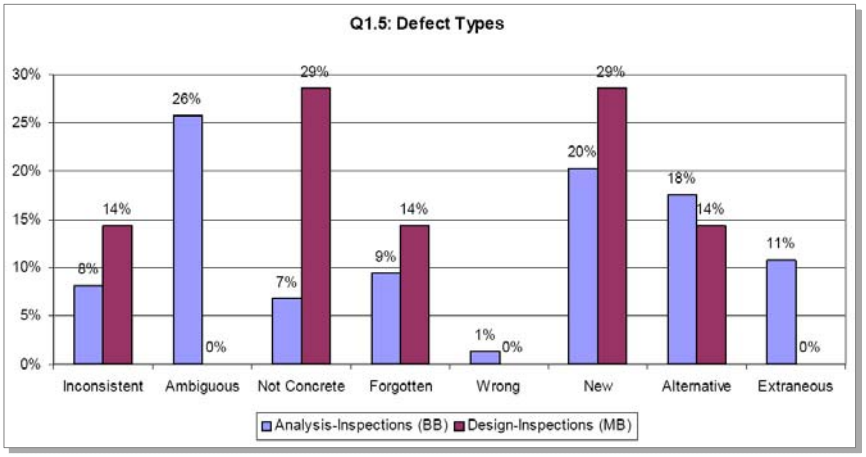
Severity of defects (Project A)



Severity of defects (Project B)

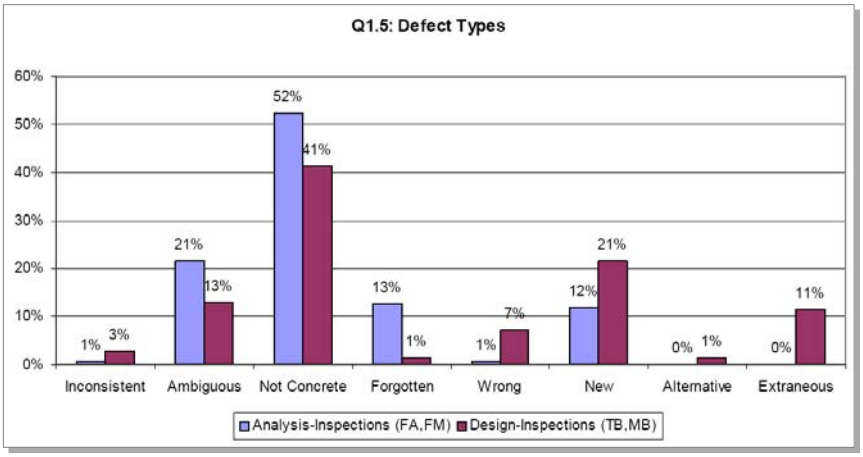


Defect types (Project A)



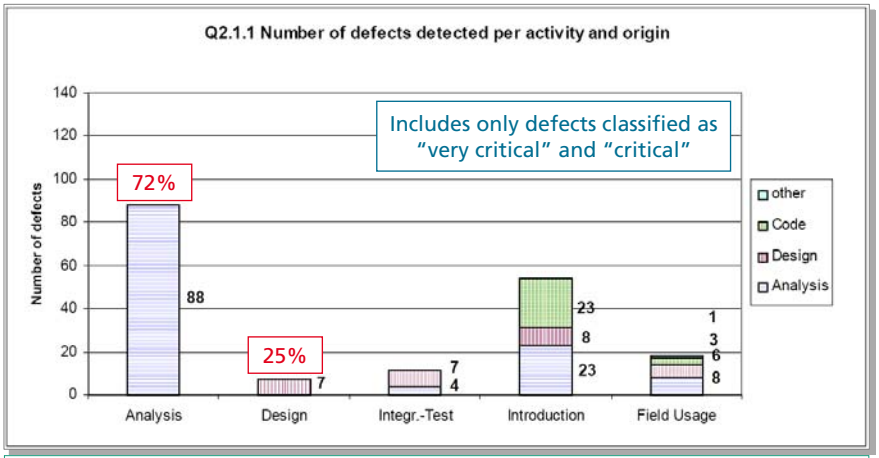
- Due to the focus on user-friendliness (of user screens and letter), the detected defects mainly considered the clarity aspects (defect type "ambiguous").

Defect types (Project B)



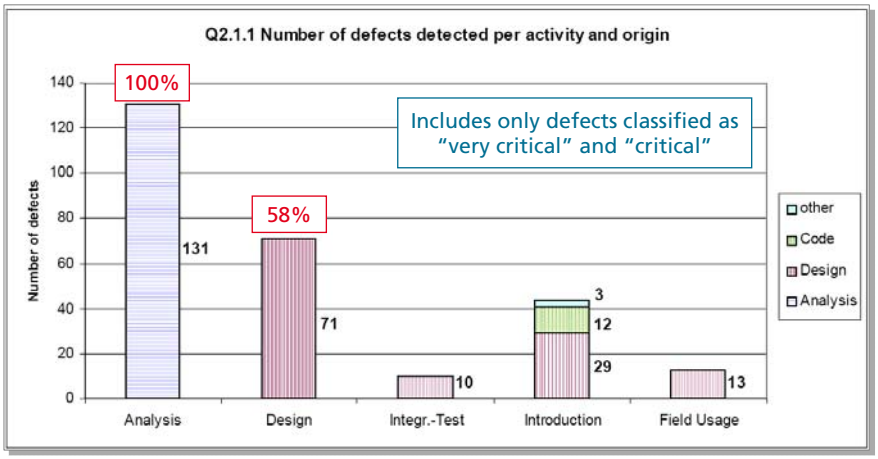
- Inspections were performed during the 1st iteration of iterative development, where a number of aspects were planned to be implemented during the 2nd iteration.

Defect slippage (Project A)



- 72% of all analysis defects were already found by inspections at the end of the analysis
- Due to high schedule pressure, only 25% of design defects were found at the end of design

Defect slippage (Project B)

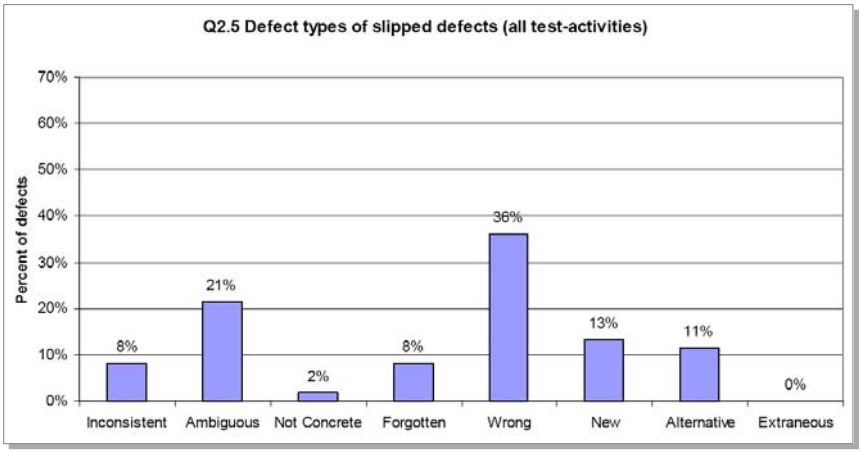


- But, analysis in Project B dealt solely with conceptual/strategic aspects of the software. Details were specified during design. It was thus actually not possible to trace defects back to analysis.

Severity of slipped defects

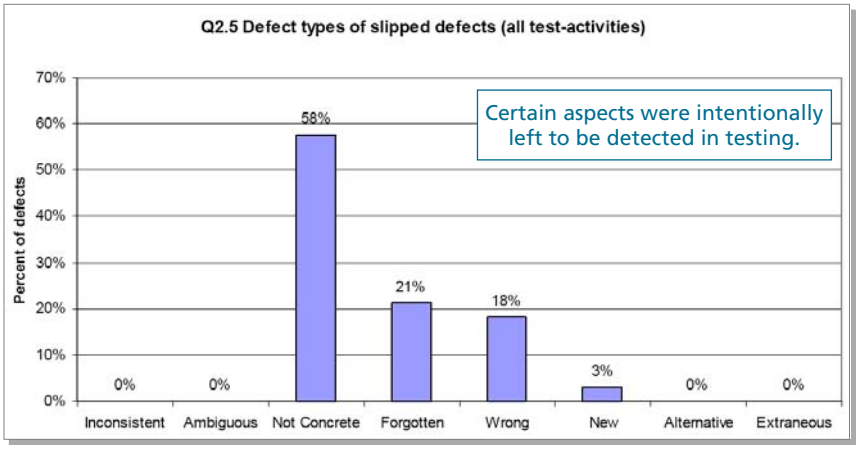
- In both projects, a high percentage (84% resp. 97%) of defects slipping through analysis and design inspections were of very critical or critical severity.

Type of slipped defects (Project A)



- Many "wrong" defects undetected → the design inspections were not thorough enough.
- Many "ambiguous" defects undetected → overall high ratio of "ambiguous" defects due to the nature of Project A (description of user outputs).

Type of slipped defects (Project B)



- Inspectors focused first on common and important aspects of the case to be implemented, whereas rare aspects of the case were intentionally left to be detected by testing.

Allianz – Benefits (of inspections and DFM)

- Customer satisfaction
  - Fewer questions in a call-center (due to improved understandability of customer letters)
  - Improved customer acceptance
  - Follow-up projects (due to high customer satisfaction)
- Organizational improvements
  - Reduced rework effort (early defects detected and removed before testing)
  - Organizational learning (of the inspection participants)
  - Process improvement (new process improvement potentials identified)
  - Meeting deadlines
  - Better communication between software development stakeholders

Part 4: Industrial Case Studies

- Bosch GS
- Allianz
- **IBS**
- Summary

IBS – Context

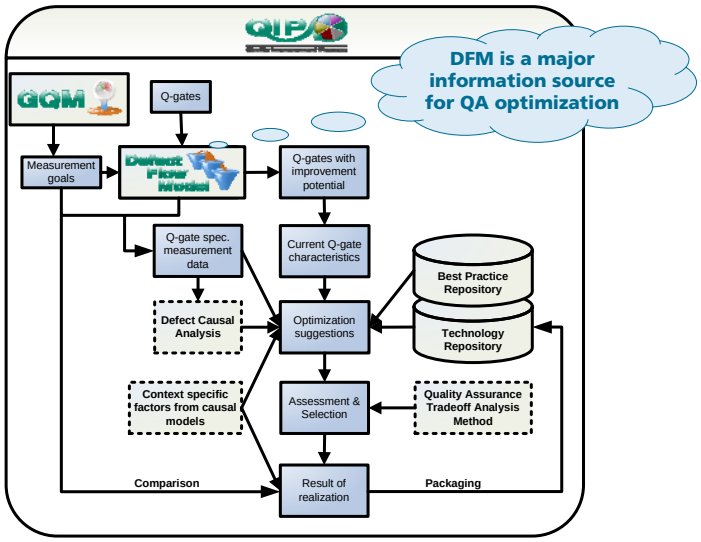
- IBS AG – A medium-sized enterprise
  - Software solutions in the area of quality and productivity management in industrial organizations
  - Approx. 185 employees
  - Revenues in 2007 approx. 20 million €
- Product:
  - Software suite with individually configurable products
  - 4,000 customer installations worldwide
  - New functionality integrated based on customer requirements
  - Short release cycles (approx. 3 months)

Quality management system "*CAQ=QSYS® Professional*" and production management system "*IBS:prisma*"

IBS – Motivation

- Ensure high quality
  - Variety of quality assurance techniques
- QA strategies
  - Coordinated usage of QA methods during all phases
  - Limited by schedule and budget constraints
  - Have to be adapted to the specific goals and context
- Small and medium-sized enterprises
  - Limited resources for software process improvement
  - No explicit team for software process improvement
  - Limited knowledge in software measurement

Framework for balanced optimization of QA strategies



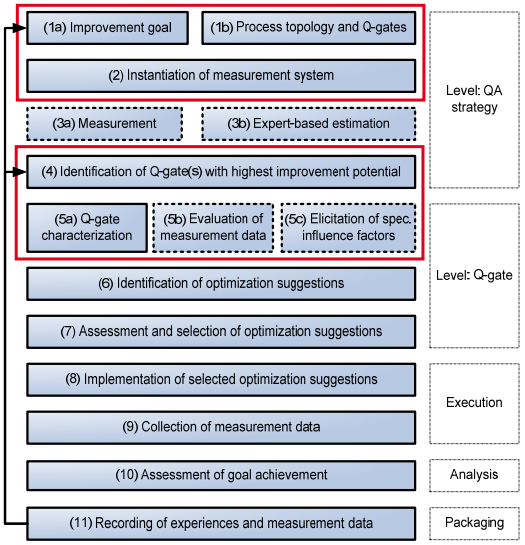
Balanced optimization process

Instantiate DFM

- Identify required measurement data by GQM
- Develop corresponding defect-flow model
- Evaluate developed DFM
- Provide data collection infrastructure
- Training & rollout

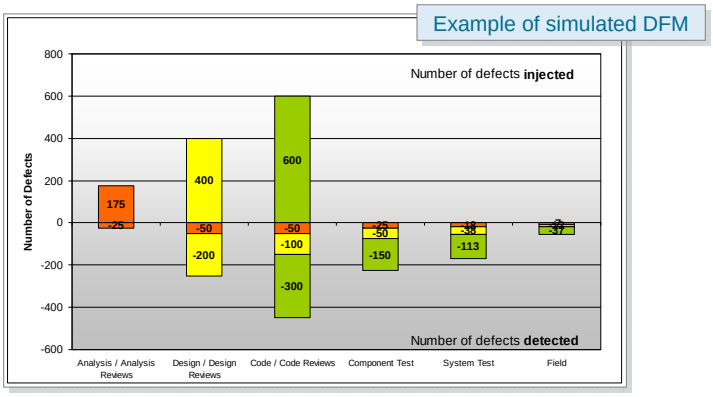
Analyze collected defect data

- Identify Q-gates with improvement potentials
- Identify Q-gate-specific improvement potentials



Motivation of defect-flow models

Example of simulated DFM



- Illustrated the benefits and basic usage of DFM

Defect classification schema used

Detection

Correction

Detection Activity	Defect Injection	Defect Flow
Type of Request Impact (Quality)	(Type of Request) Defect Type	Basic Classification
BU/Product Detected in Release	Location Corrected in Release Origin in Release	Localization and History
	Cost	Cost

Light-weight evaluation – missing classifications

Fraction of missing classifications	Total period	Current month
Type of Request	14%	3%
Location	15%	4%
Cost	13%	19%
Detection Activity	12%	2%
Impact (Quality)	14%	2%
BU/Product	10%	2%
Detected in Release	11%	2%
Origin in Release	34%	68%
Defect Injection	30%	75%

- Objective: Increase reliability
- Mean: Early detection of incomplete and wrong classifications
- Approach: Check data
  - Completeness
  - Consistency
- Effort: Low due to simple monitoring and review of collected data
- Suggested frequency: Periodically for timely response and reaction

- First light-weight evaluation showed a significant number of incomplete classifications
- A series of evaluation and feedback improved data quality (completeness & consistency)

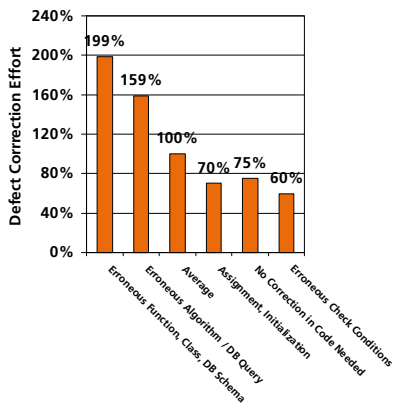
Low reliability for defect types

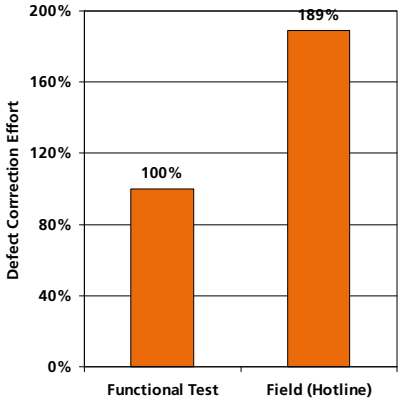
*DT = Defect type value

Kappa = 0.294		Rater 2 DT1*	DT2	DT3	DT4	DT5	DT6	
Rater 1	DT1	8%	3%	0%	3%	2%	0%	17%
	DT2	0%	8%	2%	0%	0%	0%	10%
	DT3	12%	12%	19%	7%	5%	0%	54%
	DT4	0%	0%	2%	7%	5%	0%	14%
	DT5	0%	0%	0%	0%	0%	0%	0%
	DT6	3%	0%	0%	0%	0%	2%	5%
		24%	24%	22%	17%	12%	2%	100%
Disagreement		*58%*	*50%*	*51%*	*56%*	*100%*	*50%*	

- Results stimulated revision of defect type descriptions and internal training

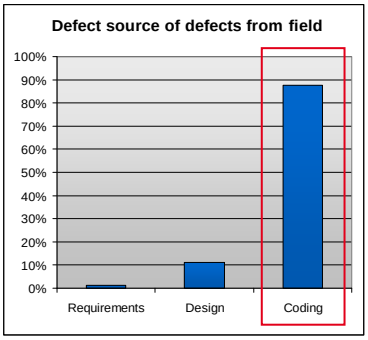
Initial analyses – effort for defect correction

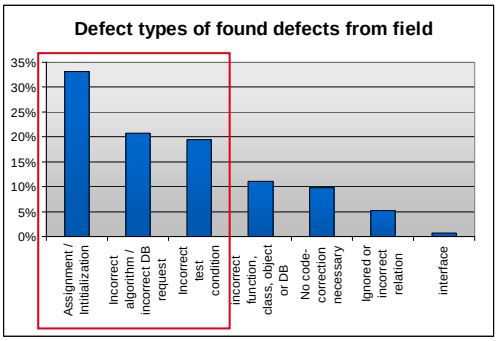




- Design-related defects (function, algorithm, database, …) are more expensive to fix
- Defects found in field are nearly twice as expensive than defects found in functional test

Identify Q-gate with highest improvement potential





- Majority of defects found in the field are coding defects
- High number of defects that can typically be found via developer self-tests
- Highest improvement potential ➔ developer self-test

IBS – Benefits and lessons learned

Benefits of DFM

- DMF are a core component of QA strategy improvement framework
- DMF allow to identify
  - Q-gates with improvement potentials
  - Q-gate-specific improvement potentials
- Provide reliable data for decision-making

Lessons Learned

- Assure reliability of defect classification
- Provide appropriate measurement infrastructure
- Most time-consuming activity:
  Instantiation of measurement system
- Important sources for improvement suggestion
  Measurement data and *quality gate questionnaire*

Part 4: Industrial Case Studies

- Bosch GS
- Allianz
- IBS
- **Summary**

Practical tips – defining defect-flow models

- Create positive atmosphere
  - Due to its negative overtone, the term "defect" is often replaced by other more positive terms
  - Example: The term "finding" is used, which refers to defects in a narrower sense but also to questions, improvement proposals, and comments
- Classify defects
  - Define a good defect classification scheme
  - Involve both domain experts and measurement experts
  - Measurement experts prepare an interview for capturing knowledge and experiences of domain experts
- Consider "correction type" as a central element of defect classification
  - Software practitioners think and talk about technical defects of certain documents
  - Correction type is thus easy to understand, as it represents the practitioners' natural way of thinking about defects
- Employ GQM paradigm when defining defect classification and defect-flow models

Practical tips – collecting and analyzing defect data (1/2)

Motivate stakeholders

- Remember that
  - For management, data collection is an additional project overhead
  - For developers, data collection is an additional burden to software developers that does not create immediate and tangible value (deliverables)
- Motivate and visualize the benefits of defect classification
  - Understanding of the benefit is essential for management commitment and data quality
- Estimate the current defect flow within the organization using available data and expert estimates
  - Provide stakeholders with an initial insight into the defect flow and the general strength and weaknesses of the V&V process

Practical tips – collecting and analyzing defect data (2/2)

- Support data collection, analysis, and interpretation with appropriate tools
- Drawing reasonable conclusions from the defect-flow data requires stable processes
  - For unstable processes, it is difficult to decide whether changes in the defect distributions are caused by chance (unstable processes) or are due to introduced process changes
  - e.g., effects of improving development and/or QA processes on defect flows cannot be evaluated

Summary – Defect-flow tutorial

Part I: Introduction

Part II: Defect Classification

- Defect Classification Theory
- Prominent Approaches

Part III: The Defect-Flow Model Approach

- DFM Principles
- DFM Creation and Introduction Process
- DFM Application

Part IV: Model Maintenance

- Defect-Flow Models as Part of IT Strategy
- Maintaining Defect Classification Schemes

Part V: Summary

Deploying DFM technologies

Preparation

- Characterizing the context of DFM
- Linking DFM technology to business objectives
- Developing a custom-specific model
  - Definition of appropriate attributes and values

If you need help, please do not hesitate to contact us

Deployment

- Integration with defect-tracking processes
- Tool support
- Training, coaching

Application

- Collecting, validating, and pre-processing measurement data
- Visualizing and interpreting results

Maintenance

- Maintaining model and measurement database
- Regularly verifying the model's efficiency

Thank you for your attention!

Michael Kläs
Phone: +49 (631) 6800 2110
Fax: +49 (631) 6800 1398
Email: michael.klaes@iese.fraunhofer.de

Dr. Jens Heidrich
Phone: +49 (631) 6800 2193
Fax: +49 (631) 6800 1398
Email: adam.trendowicz@iese.fraunhofer.de

Dr. Adam Trendowicz
Phone: +49 (631) 6800 2137
Fax: +49 (631) 6800 1398
Email: adam.trendowicz@iese.fraunhofer.de

Dr. Jürgen Münch
Phone: +49 (631) 6800 1301
Fax: +49 (631) 6800 1398
Email: juergen.muench@iese.fraunhofer.de