Fraunhofer IESE

Definition, Application, and Maintenance of Defect-Flow Models

Part 5: Practical Exercises

Instructors:	Michael Kläs and Dr. Jens Heidrich
Tutorial Date:	Dec. 11, 2009
Location:	IPA/SEC Building

Exercise 1: Definition of Defect Classification Schema

Step 1: Goal-oriented identification of scheme attributes (20 min)

Please specify a defect-related measurement goal that should be addressed by the defect classification and identify the required attributes of the classification scheme. You can choose between two alternatives:

A – You may specify a measurement goal of interest in your context.

B – You may specify a measurement goal based on the example scenario provided.

Alternative A

Object	*Purpose*	*Quality Aspect*	*Viewpoint*	*Context*
System Test	Characterize	Effectiveness	Tester	Project X
Questions to be answered				
Q1: "How effective is the system test?" M1: Defects detected by system test / (defects detected by system test + defects found in field)				
Classify the defects found with respect to the following attributes:				
A1: "detection activity" = { system test, other QA activity, field }				

Alternative B

"With our product Y, we have the feeling that customers consistently complain about performance issues. Although we have analyzed the reported performance problems and tried to fix them, a large number of performance issues seem to be a constant problem in all releases of our product. Now, we want to determine whether there is really a large number of performance issues in the product and if so, we want to find the origin of these issues in our development process to start a process improvement initiative in the next step."

Object	*Purpose*	*Quality Aspect*	*Viewpoint*	*Context*
Performance issues (detected by customer)	Characterize	Origin in development process	Quality Manager	Product Y

Questions to be answered

Q1: "What percentage of defects reported by customers are performance issues?"
M1: Reported performance issues / (reported performance issues + reported issues that address other qualities)
Q2: "Which activities introduce the reported performance issues?"
M2: Distribution of reported performance issues across the development activities

Classify the defects found with respect to the following attributes:

A1: "impact on product quality" = { performance, integrity/security, reliability, usability, functionality, other }
A2: "injection activity" = { requirements, high-level design, low-level design, code }

Step 2: Refine selected classification attribute (10 min)

Please refine one of the attributes identified in Step 1. Write a short definition of the attribute (e.g., what question is being answered) and define appropriate attribute values. Provide a name, definition, and example for each attribute value.

Keep in mind the properties of a good classification scheme: *orthogonality*, *completeness*, *understandability*, and *number of values (7±2)*.

Attribute Name	
Attribute Definition	

Attribute Value Name	*Definition*	*Example*

Exercise 2: Application of Defect Classification Scheme

Step 2.1: Apply scheme for defect classification (15 min)

Please classify the following defect. Apply the ODC attribute values provided for "defect type" and/or the defect attribute vales you developed in Step 1.2.

ID	*Defect Description*	*Defect Type Value*	*Your Attribute Value*
1	It is not enough to tell people only to enter one- to three-digit numbers. Some will enter letters or ten-digit numbers and others will press five times to see what happens. One can enter such numbers, but the program is not able to cope with it.	Checking	
2	The program can not only detect an error but correct it, without having to bother anyone about it, by checking other data or a set of rules. This is desirable, but in this case the "correction" was not correct.	Algorithm/Method	
3	A re-entrant program can be used concurrently by two or more processes. A routine serving two processes does not keep its data separate, so that what it does for one process corrupts what it does for the other one.	Timing/Serial	
4	A process produces a lot of output. Formatting all this information and sending it to the printer or screen takes a lot of computer time. When the computer is operating under heavy load, the output-intensive process should try to send out less, but it doesn't.	Function/Class/Object	
5	The program tries to send 100 characters per second across a connection that only supports transmission of up to 10 characters per second.	Interface/O-O Messages	

Please describe the difficulties you observed during the classification of the defects.

For instance, is there an unambiguous classification of each defect? Does the defect description provide the information required for an unambiguous classification? If not, when would the required information be available (when the defect is detected or when it is fixed, or not at all)? Describe the issues you observed.

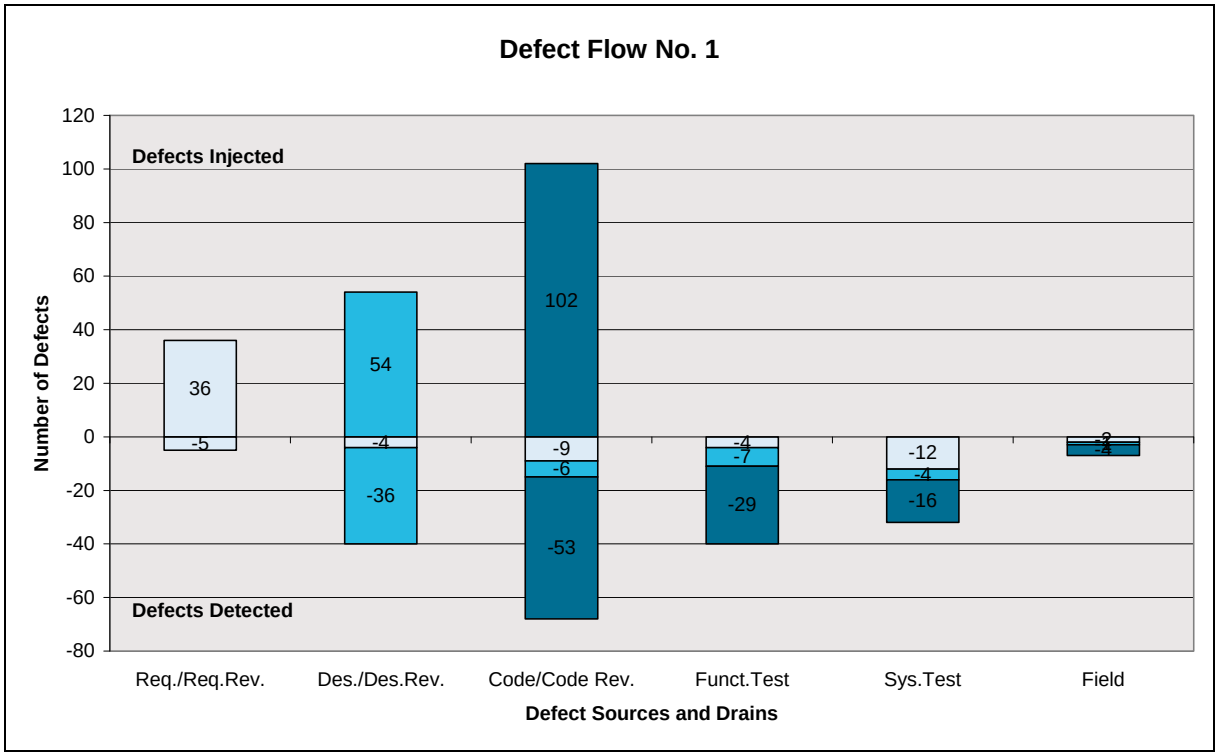
Observations	

Step 2: Analyze defect classification information (15 min).

Please consider the following three charts. Each shows an exemplary defect flow measured in a typical project of the respective organization. Each organization follows its own but stable development process. Due to simplification reasons, we use the same defect injection and detection activities for each defect-flow chart.

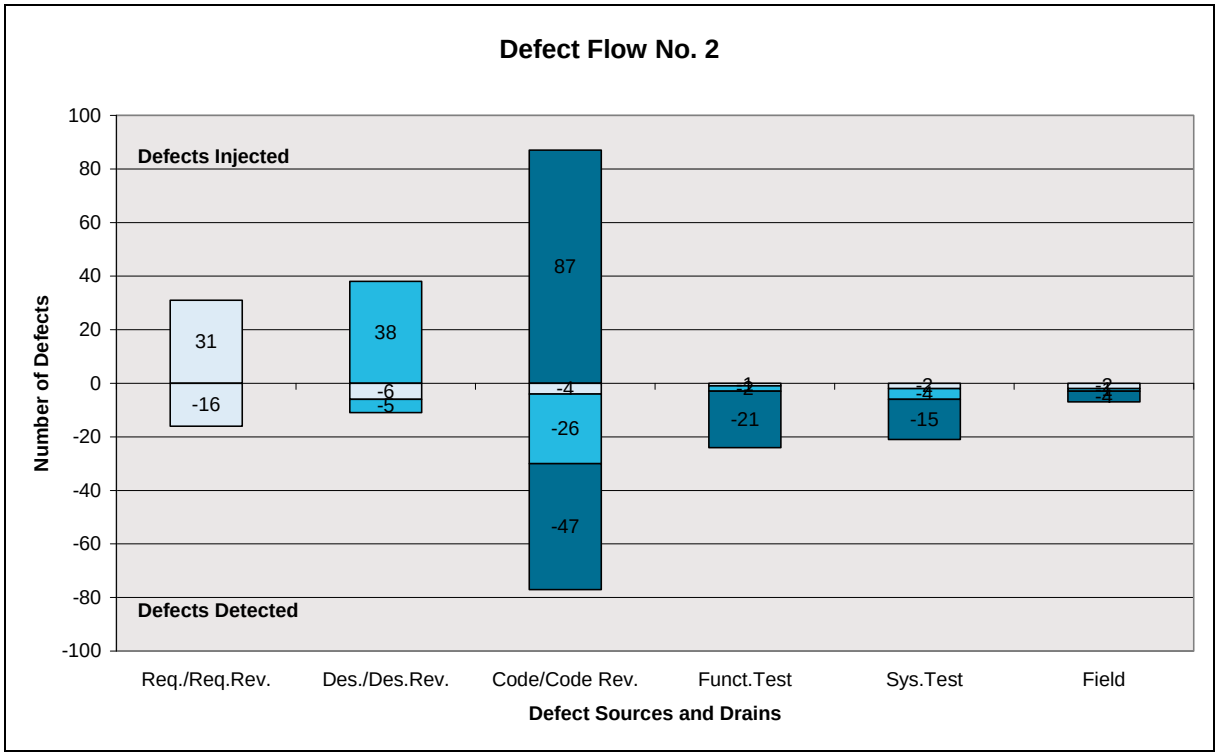
Try to interpret the charts and draw conclusions with respect to the development and quality assurance processes. Which improvement activities may one suggest? Is there additional defect information that may help you to decide on a specific improvement suggestion?

Organization 1



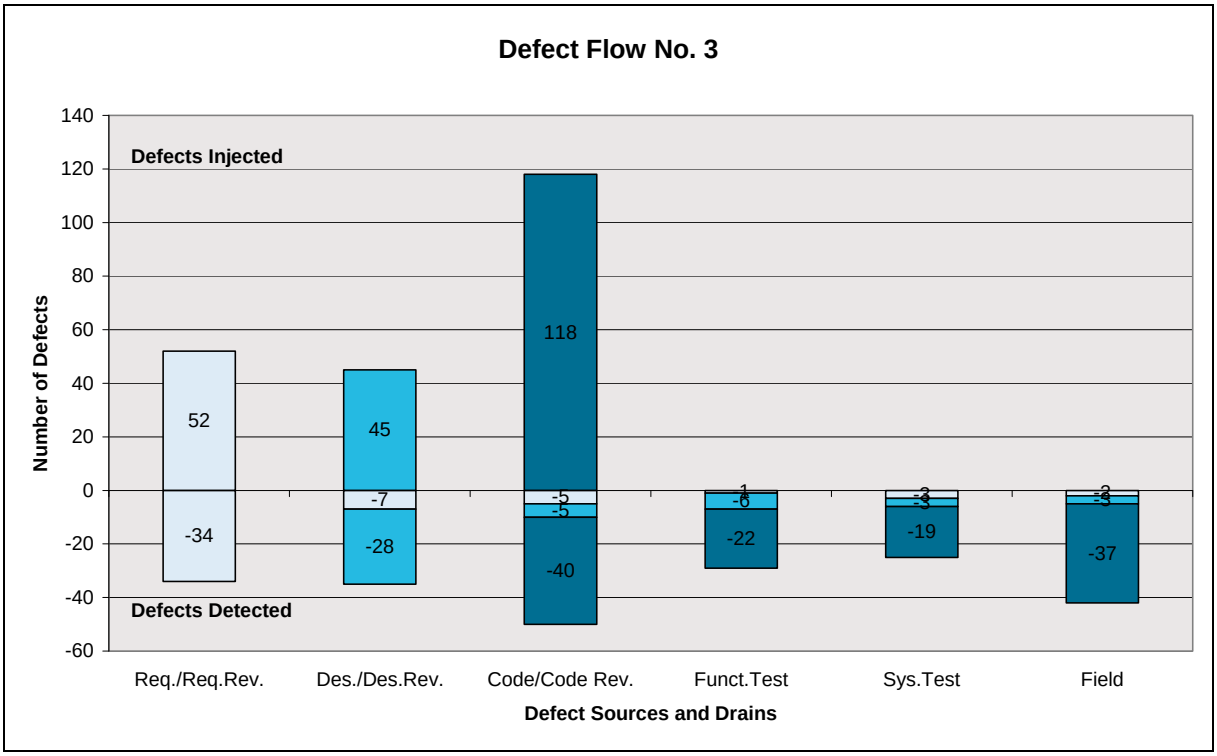
Interpretation	The requirement reviews are not very effective. Many requirement defects intrude deep into the development process. They are not found during the design phase but during coding and system test. This results in expensive rework in the late phases of the project.
Improvement Suggestions	Improve the requirements review process (e.g., introduce requirements inspections). To provide a checklist for the requirements reviews/inspections, a defect classification in terms of “correction type” may be helpful.

Organization 2



Interpretation	Many design defects are detected during coding and lead to unnecessary rework. The design reviews are not sufficiently effective (or did not take place). To provide a checklist for the design reviews/inspections, a defect classification in terms of “correction type” may be helpful.
Improvement Suggestions	Introduce or improve design reviews (e.g., use design inspections).

Organization 3



Interpretation	Many coding defects are found in the field; this should be avoided. The question is: Where to find them best? The effectiveness of code reviews seems to be okay. In this case, e.g., the analysis of the "correction type" distribution can provide more information.
Improvement Suggestions	If many "assignment" or "checking" defects occur, one should consider improving the code review or the function test. If there are many "interface" defects, one may focus the improvement activities on the system test.

Discussion of results (30 minutes)