1 Introduction

Wouldn’t it be nice if you could have more insight into the quality of a product, while it is developed, and not afterwards? Would you like to be able to estimate how many defects are inserted in the product in a certain phase, and how effective a (test) phase is in capturing these defects? The simple but very effective model described in this paper makes it possible! The model has been used at Ericsson to develop and release a new version of a network management product. The project using the model followed normal development phases, from initial requirement setting to test at the first customer site. The model was used to control the project, by putting quality next to planning and budget, evaluate risks, and to take decisions regarding release and maintenance.

This paper will first highlight why there was a need for such a model, and why existing measurements didn’t fulfill this need. Then the model itself, and the actual implementation in the project is described. Following that some of the conclusions that were drawn from the model, using feedback sessions, is described. This explains how the project has benefited from the model. At the end there is an exploration into the future, regarding both the model and the needs and directions of the organization regarding measurements on product quality.

2 Why a Model?

Within Ericsson there has always been focus on the quality of developed products, next to planning and budget. Initially measurements like fault density were used. But fault density has major drawbacks; one being that you can only measure it after a phase is completed, and another is that it does not give any insight on the causes if a product has a quality risk. For instance, a high fault density can either mean that there is a quality risk, or it means that the product was more thoroughly tested than other products, or both. The same applies for a low fault density, the reason could be that insufficient testing was done and that defects remain undetected in the product (a product quality risk), or that the product has a better quality and thus less defects were found, or both. Studies outside of Ericsson have also revealed the limited value of fault density; see for instance [1].

Given the shortcomings of fault density, there was a need for a new measurement that would give more insight: a measurement usable to plan quality at the start of the project, and track it during project phases, instead of after a phase or project is finished. Enabling corrective actions and reduction of quality risks in a project. An additional project need was to estimate the number of latent defects in a product. The purpose is twofold. First it can be used to decide if the product can be delivered to customers, or released, knowing the quality (for more details, see [3]). Second it helps to plan the support and maintenance capacity needed to resolve the defects that are anticipated to be reported by customers.

To address this need, a defect prediction model was developed. The model makes it possible to take corrective actions and reduce quality risks during the project. For instance, it informs the project of quality risks throughout the project phases and can be updated continuously as new defect data become available (see [2]). The model relies on detailed defect data, especially with respect to the phases of defect insertion and detection. The model is based on earlier research at IBM. To develop the model, descriptions from Watts Humphrey [4] and Stephen H. Kan [5] have been used.
3 How does the Model Look?

To get more insight into the quality of the product during development, it is needed to measure the software development processes with two views: Introduction and detection of defects.

Introduction is done during the specification, design and coding phases; defects are either introduced into documents or into the actual product. Measuring introduction gives an indication of development phase quality. Detection of defects is done via inspections and test during all the phases of the project. Measuring detection gives insight into the effectiveness of verification phases. By using these two measurements, a project can determine if there is a quality risk, and what the origin is: Too many defects in the product and/or insufficient testing to capture the defects.

Specific measurements have been defined in the model:
- Defect density: # of inserted defect per phase, insight on development quality
- Detection rate: % of defects detected, insight in effectiveness of verification
- Fault slip through: % and # of defects that was not detected in the phase were introduced
- Defect classification: Kind of defect, where should the defect have been found?
- Defect level: # of latent defects in the released product.

In the next paragraphs, development phase quality and detection effectiveness is explained in detail.
3.1 Development Phase Quality

The quality of a product depends on the number of defects that are inserted during the development phases. Mistakes are made in every phase, from specification to implementation. Defects that are detected and removed increase the likely quality of the end product. However, those defects reflect the inefficiency of the development process. Defects which are not detected in the phase in which they are inserted increase rework and decrease product quality, since they remain in the product after release and can surface when customers use the product. The aim is to remove defects as early as possible and to have as few defects as possible in the product when released, thereby delivering quality products.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Expected #def</th>
<th>Expected DD</th>
<th>Act. Size</th>
<th>Fnd #def</th>
<th>DD Act</th>
<th>Not found yet</th>
<th>% Found</th>
<th>% Exp of total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification</td>
<td>4</td>
<td>10</td>
<td>0.4</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>100%</td>
<td>4%</td>
</tr>
<tr>
<td>High Level Design</td>
<td>12</td>
<td>107</td>
<td>0.112</td>
<td>107</td>
<td>10</td>
<td>0.09</td>
<td>83%</td>
<td>12%</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>12</td>
<td>47</td>
<td>0.255</td>
<td>47</td>
<td>10</td>
<td>0.21</td>
<td>83%</td>
<td>12%</td>
</tr>
<tr>
<td>Implementation</td>
<td>70</td>
<td>15000</td>
<td>4.667</td>
<td>13000</td>
<td>18</td>
<td>0.00</td>
<td>52%</td>
<td>26%</td>
</tr>
<tr>
<td>Total</td>
<td>98</td>
<td>42</td>
<td>56</td>
<td></td>
<td></td>
<td></td>
<td>43%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Figure 3: Defect Model, Insertion Table

An organization will produce software with a number of defects between statistical limits based on the maturity of its development processes (See the CMMI [6] for more information about quantitative management). Depending on the maturity of the organization, the distance between the limits becomes smaller and increases the reliability of defect estimates. Additionally, the defect insertion and detection profile across the development phases becomes repeatable which makes it possible to estimate the number of defects in a future phase when one or more phases are finished. Based on these estimates, defect control limits can be defined. When the actual number of defects goes over a limit, it signals a potential quality risk to be investigated.

At the start of a phase the number of inserted defects can be estimated. During execution of the phase this estimate can be adjusted based on the number of defects actually detected. Since it is sometimes difficult to estimate the number of defects, an alternative method is to estimate the size of the produced documents or code, and use size multiplied by the defect density to estimate the number of defects. In all cases, it is better to do a rough estimate, and adjust it during a project, than to do no estimate. Historical data of earlier projects can be very useful when estimating defect introduction. Also industry data can be used when no historical data are available (see [1, 7-9]).

3.2 Defect Detection Effectiveness

The aim of verification is to detect the inserted defects, preferably in the earliest phase that they can economically be detected. The effectiveness can be expressed with a detection rate, that is:

\[
\text{Detection Rate} = \frac{\text{Number of defects detected}}{\text{Number of defects present in product}}
\]

An organization has a detection rate for a certain phase, which can be estimated within certain statistical limits. Initially when no historical data of an organization is available, industrial figures can again be used, see [1, 7-9]. An alternative for the detection rate is to estimate the absolute number of defects that are likely to be found in the current phase. Based on that number and the number of defects present, the detection rate can be derived.
Defect detection

<table>
<thead>
<tr>
<th>Phase</th>
<th>Target detection rate: 70% document, 60% code, 50% test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Availability</td>
</tr>
<tr>
<td></td>
<td>Det #</td>
</tr>
<tr>
<td>Specification</td>
<td>4</td>
</tr>
<tr>
<td>High Level Design</td>
<td>14</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>15</td>
</tr>
<tr>
<td>Implementation</td>
<td>79</td>
</tr>
<tr>
<td>Unit test</td>
<td>39</td>
</tr>
<tr>
<td>Function test</td>
<td>31</td>
</tr>
<tr>
<td>System Test</td>
<td>16</td>
</tr>
<tr>
<td>Network Test</td>
<td>7</td>
</tr>
<tr>
<td>Installation</td>
<td>5</td>
</tr>
<tr>
<td>First Customer</td>
<td>4</td>
</tr>
<tr>
<td>Average/Total</td>
<td>95</td>
</tr>
</tbody>
</table>

Table 4: Defect Model, Detection Table

During the execution of a phase, the detection rate can be adjusted based on the actual number of defects detected. If, for instance, a detection rate of 50% is expected, and 46% of the expected defects are detected halfway through the phase, then either the number of defects that was inserted will be higher than initially expected, or the actual detection rate is higher – fewer defects were inserted than were predicted. If the first is true, then there is a quality risk in the product, which needs to be investigated. Also it gives a signal usable to improve the process phase where the defects were introduced. In a next increment of the project, defect introduction can thus be reduced. If fewer defects were inserted and thus the resulting detection rate is higher, then further investigation is warranted to understand how this was accomplished. That would make it possible to learn and improve verification in other projects, based on the positive experiences from this one.

The combination of measurements on defect insertion and defect detection gives a more detailed view of the quality of the product, and effectiveness of the development processes. This provides a project with better means to track and control quality. As stated earlier, the theory behind this model is not new. The focus of Ericsson, and this paper, was to implement this model, and determine if and how it can be valuable for projects.

4 Implementation of the Model

The defect introduction and detection model as described in the earlier paragraphs was implemented in a pilot project for a network management product. The project requirements were to:

1. Upgrade the product with new versions of hardware platform, operating system and database.
2. Extend the product with a new interface, used to communicate with a telephone exchange.

Requirement 1 mainly had to do with modifications in installation and configuration scripts, with a focus on compatibility; no new functionality was added. For requirement 2, new functionality was developed on a different platform; hence this resembled a “normal” development project. Given the two distinct requirements, the decision was made to split the project in two increments with separate teams, which were overlapping in time. The model copes with these two increments separately, since different processes were used.

Given that the first part of the project was combined for the two increments, and also final testing was combined, the basic introduction/detection model was adjusted. The model, based on the development phases of the project, consisted of:
A tool for the model was developed using a spreadsheet. The purpose of the tool was to estimate the number of defects inserted and detected by phases, and to track all defects from inspections and tests against these estimates. The tool supported analysis of the data with both calculated values and graphs comparing actuals to estimates in terms of current status and trends over time.

Together with the model and the tool, an approach was defined to introduce the measurement in the project. This was based on the following key implementation issues:

1. A project sets targets, collects data, and compares actual data with estimates to see where there are quality risks.
2. Those who have performed the measured process are the only persons that can do validation and analysis of the data.
3. The purpose of the model is to steer the project, hence focus should be to reach conclusions and perform corrective and preventive actions.

**Ad 1:** The project had two targets:

- Maximum Defect Density at introduction: 1 defect per page.
- Minimum Detection Rate in a phase (inspection/test): 50% of defects found.

The first target is used to measure development phase quality; this is the number of defects that is made in a phase. Of course this is initially an estimate. During the project, when data is collected and defects are traced back to their introduction phase, actual defect density is calculated and compared with the estimate. The second target measures verification phase effectiveness. It is initially also an estimate, but during the phase the actual detected defects are entered in the model and an actual detection rate is calculated, which is again compared with the estimate.

**Ad 2:** Regular meetings have been held with project team members to go through the data. Initially the meetings were used to “plan quality”, by estimating the number of defects that was inserted in a certain phase, based on the number and size of documents or code to develop. Then data was collected, and the actuals were feedback to the developers together with the estimates in feedback sessions, in which the estimates were revised if needed.

**Ad 3:** In the same meetings where the data was discussed, the developers (designers, testers, and project managers) were questioned about the data. The purpose was to determine the story behind the data, i.e. why there were more or less defects in a document or phase as initially was estimated, and why the detection rate was higher or lower as expected.
Related to 2 and 3, feedback was a critical success factor. More information about the approach for the feedback, and the results, will be documented in a separate paper.

The initial model as described in chapter 3 has been extended during the project with Orthogonal Defect Classification (ODC) [10], and Test Matrices based on requirements. These techniques were chosen for detailed analysis of the defect data, helping the project in making good decisions about process improvement and test focus. From ODC, the concept of triggers was used to determine test progress, and effectiveness of the test phases. This helped us answering questions like “are we really finding System Test defects in the System Test phase, or are we still finding Function Test defects?” The Test Matrix was used to allocate defects to requirements. The matrix showed for each major requirement if defects have been found, and in which test phase. This was compared with the requirement focus of the test phase, and used to take decisions regarding if requirements were sufficiently verified, and could be released.

5 Results Using the Model

5.1 Data from the model

The model and tool have been used since 2001 in the project. 414 defects (status April 2003) have been collected, which were analyzed and classified on introduction phase, requirement, and phase where they could have been detected. The result data gives an estimate of 21 latent defects in the released product, expected to be found in the first six months of operation. This estimate was used as one of the criteria in the release decision; it was decided that this would be an acceptable quality level provided that sufficient maintenance support would be available to solve the 21 defects when detected by customers. The 6 months operation period ends in June 2003, until now (in the first 4 months) there are 14 defects detected.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Latent defects</th>
<th>Detected defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection</td>
<td>421</td>
<td>197</td>
</tr>
<tr>
<td>Test in project</td>
<td>224</td>
<td>194</td>
</tr>
<tr>
<td>1st customer test</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>Project totals</td>
<td>421</td>
<td>400</td>
</tr>
<tr>
<td>Maintenance</td>
<td>21</td>
<td>14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity</th>
<th>Act</th>
<th>Test</th>
<th>Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection</td>
<td>47%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test in project</td>
<td>87%</td>
<td>36%</td>
<td></td>
</tr>
<tr>
<td>1st customer test</td>
<td>87%</td>
<td></td>
<td>67%</td>
</tr>
<tr>
<td>Project totals</td>
<td>96%</td>
<td>91%</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6: Defect Figures from Pilot Project

Based on the estimated number of latent defects, the project has a defect detection rate of 95%, i.e. 400 of the 421 defects made in the project have been detected before the product is released. If we exclude the phases before test (that used inspections for verification) from the measurement, the detection rate is lower: only 91% of the defects left after inspections were detected in the test phases. This shows that inspection has contributed towards the quality of the released product. However, the average detection rate from inspections is 47%. According to industry data, inspections can detect between 60%-80% of the available defects, so there is room for improvement.

The targets that were set for the project have been met partially. The benefit of the model has been that it made the targets measurable, and that the project was able to take corrective actions on time. Except for some documents, the defect density remained below the target of 1 defect per page that was defined. The documents that went above the target were analyzed, and actions were taken to limit quality risks. Regarding the detection rate, the initially set target of 50% in all phases was partially met. Based on detailed analysis, the target was revised during the project.
Detection Rate

0%
50%
100%

Re  Arch  Design  Impl  UT  FT  ST  Deliv  Cust  Newb  Maint

Detection Rate
Initial Target (50%)
Modified Target

Figure 8: Detection Rates

It became clear during the project that the target of 50% detection rate was not challenging enough for the phases where inspection was used for verification. Higher detection rates were reached, and the target was revised towards 70% for documentation and 60% for code inspections. Note however that for requirement inspection the project did not meet this detection rate, and also for code the actual detection rate was lower than the revised target. For test phases it became clear that the target depended very much on the focus, and the effort that the project wanted to spend in that phase. The actual detection rate varied from 16% to 86%, where the lower detection rate was in final test phases, which tested with a limited scope, and the high detection rate was in a main system test phase. Based on that data the targets were also revised.

5.2 Analysis results and actions taken in the project

Even more important than the data was the benefit the project received by using the model. During the project, data feedback and analysis sessions were held where corrective actions based on the data were implemented. Major conclusions/actions included:

- A slip through of requirement defects was detected early in the architecture phase, based on a detection rate below the 50% target in requirements. At that time it became clear that the inspection of the specification had revealed 32% of the available defects. Investigation showed that good high-level design, combined with effective inspections, had revealed many requirement clarifications. Action defined was to monitor requirement defect detection in the design phase for quality risks; later it turned out that both the number and impact of the detected defects was limited. No more requirement defects were detected in later phases, final conclusion is that the requirements after initial clarification reached a high quality.

- Data from defects inserted/detected, test requirement coverage, and Orthogonal Defect Classification, has shown that inspection effectiveness depends on several issues: Good and focused inspectors, qualified moderators, sufficient preparation, and thorough inspection planning. The detailed conclusions on inspections are used to further improve reviews and inspections in future projects. Though it was known that inspections are an effective way of detecting defects (as was to be expected from many earlier studies), our data confirmed this and has lead to more focus from management and buy in for further improving inspections.

- Data also made clear that test phases discovered defects that could have been found in earlier phases. Function Test found many inspection defects, where System Test discovered a lot of Function Test defects.
Based on Trigger analysis with Orthogonal Defect Classification, we could also determine our test progress. Together with a requirement based test matrix, the project has been able to predict where requirements are sufficiently verified, and where there are risks of latent defects. Test focus and scope has been changed during the project, based on data from the model, and remaining quality risks are on requirements that are seldom used.

- The estimated number of latent defects after released varied significantly during the first 6 months of the project, but became stable afterwards. Assumed is that the predicted number of defects will be within +/- 25% range of the estimate.

The number of latent defects is used in the decision to release the product, and to reserve maintenance capacity. Verification of this latent defect assumption is ongoing, as the product is sold and installed for multiple customers. The actual figures will be calculated after the product has been used in the field for 6 months, based on the received number of defect reports.

6 Conclusions

The Project Defect Model has been beneficial to the project. It has helped estimating, planning, and tracking quality during the project. This quality data has been used in the project together with time and cost data, to take better decisions. Also it has identified quality risks at an early stage, helping the project take corrective actions and decisions on product release and maintenance capacity planning.

The teams that have used the model have gained significant quantitative insight into their design and test processes, that is used in future projects. Feedback sessions of defect data analyzed by the team themselves proved to be very powerful. The model is being implemented into other project in the organization, to bring similar benefits.
There have been only some publications on defect models, as mentioned in [11]. This publication aims to support future research in this area, and wants to open the road for other projects that have used defects models to publish and share their conclusions. Main interest is on estimation techniques for defect insertion, but also additional publications on detection effectiveness would be useful.

Future extensions of the model will include effort spend in design and test phases. This would open opportunities to further measure and analyze productivity and test effectiveness.

Acknowledgments

The author wishes to thank many Ericsson colleagues, who brought in experiences with defect measurements, and gave feedback on this paper. Thanks also to Dave Zubrow (Software Engineering Institute) and Ram Chillarege (Chillarege Inc.) for reviewing a preliminary version of this paper and providing many useful comments.

About the Author

Ben Linders is Specialist Operational Development & Quality at Ericsson R&D, in the Netherlands. He is a Bachelor in Computer Science, and has done a Master study on Organizational Change. He is working in process- and organizational improvement from 1992 onwards. His focus is on implementing high maturity practices, with the aim of improving organizational performance, bringing business benefits.

Since 2000 he leads the Defect Prevention program. He coaches implementation of Root Cause Analysis (RCA), and has done many RCA sessions. Also he is local champion for Reviews and Inspections. He has defined and applied a Project Defect Model, used for quantitative management of the quality of products and effectiveness of verification.

He is a member of several (national and international) SPI, CMMI, and quality related networks, has written several papers, and regularly gives presentations on related subjects. He can be reached at +31 161 24 9885, email ben.linders@etm.ericsson.se.

References


http://reach.ucf.edu/~eel6887/DACSroi.pdf


[10] Orthogonal Defect Classification, IBM, Ram Chillarege. See:  


For more information, contact:  
Ben Linders  
Operational Development & Quality  
Ericsson Telecommunicatie B.V.  
Research and Development  
Tel: +31 161 24 9885  
Fax: +31 161 24 2617  
ben.linders@etm.ericsson.se  
http://www.ericsson.nl