Empirical Studies of Inspection and Test Data at Ericsson

Amarjit Singh Marjara; Cap Gemini Norway AS
Reidar Conradi; NTNU
Børge Skåtevik, STC
Profes’99, 22-24 June, 1999, Oulu, Finland

Agenda

■ Background
■ The inspection method
■ Data
■ Observations/questions
■ Results
■ Conclusions
■ Recommendations
Purpose of the studies

- **H₁**: To investigate if there is a correlation between defects found during inspection/test and the complexity.

- **H₂**: To investigate if there is a correlation between the number of defects found in field-use and the complexity and the modification rate of a module.

- **H₃**: To investigate if there is a correlation between defects rates across phases and deliveries for individual documents/modules.


---

Background

- Data are collected at Ericsson AS, AXE-division, Oslo.
- Every development document (design, code,..) is inspected.
- Using Gilb method; an extension of Fagan’s.
- Data for many projects are analysed.
- The analysed data orginates from design, unit test, function test and system test. Code is not inspected in this manner.
Background 2(3)

- The paper is divided in two studies:
  - Study one:
    - Data from one project of 20,000 man hours.
    - It includes design, implementation, unit test and function test.
    - The initial phases, such as pre-study and system study, are excluded from 20,000 mh.

Background 3(3)

- The second study:
  - A study of 6 projects => 100,000 man-hours
  - It includes design, implementation, unit test and function test.
  - The initial phases, such as pre-study and system study, are excluded from 100,000 mh.
### The inspection method 1(2)

<table>
<thead>
<tr>
<th>Entry Evaluation and Planning</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Kickoff</td>
<td></td>
</tr>
<tr>
<td>Reading (individual)</td>
<td></td>
</tr>
<tr>
<td>Inspection Meeting</td>
<td></td>
</tr>
<tr>
<td>Causal Analysis</td>
<td></td>
</tr>
<tr>
<td>Discussion Meeting</td>
<td></td>
</tr>
<tr>
<td>Rework</td>
<td></td>
</tr>
<tr>
<td>Follow-up and Exit Evaluation</td>
<td></td>
</tr>
</tbody>
</table>

### The inspection method - 2(2)

- Provide special training for the moderators.
- Inspection meeting max two hours.
- Follow the recommended, “optimal” inspection rates for the actual document type.
- Do not cover too much complex material in a single review.
- Invite the most competent inspectors to the meeting.
- Avoid personal criticisms.
- Postpone long discussions till end of the meeting.
Inspection Data

- **Block** - name of the block (module).
- **Document type** - the type of document which is inspected.
- **Hours saved/lost** - for every inspection, estimating whether time has been saved or not.
- **Number of persons** - reading and participating in the inspection meeting.
- **Planning** - time spent on planning the inspection.
- **Kickoff meeting** - time spent on introducing the document to the participants.
- **Reading** - total time spent on individual preparation.
- **Inspection meeting** - time
- **Rework, follow up** - time
- **Defects found during reading**: classified (Super major or Major)
- **Defects found in inspection meeting, pages read, pages treated in inspection meeting**
  - Defect classification: Super major, Major (not refined defect classification)
- ** ===> summerised in inspection survey** stored in a DB

Collecting Data - testing

- **Unit test**
  - No of defects found and time spent
  - These data are available per module (=unit)
- **Function test, system test, field-use**
  - Cause
  - Priority - indicates the seriousness of the defect
  - These data are available per module
  - The time spent on system test and field-use is not available (final integration in Stockholm).
Results

STUDY 1 - How cost-effective are inspections?

Results from study 1:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Defects [#]</th>
<th>[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual reading</td>
<td>928</td>
<td>61.8</td>
</tr>
<tr>
<td>Inspection meeting</td>
<td>29</td>
<td>1.9</td>
</tr>
<tr>
<td>Desk check (code review + unit test)</td>
<td>404</td>
<td>26.9</td>
</tr>
<tr>
<td>Function test</td>
<td>89</td>
<td>5.9</td>
</tr>
<tr>
<td>System test</td>
<td>17</td>
<td>1.1</td>
</tr>
<tr>
<td>Field use</td>
<td>35</td>
<td>2.3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1502</strong></td>
<td><strong>100.0</strong></td>
</tr>
</tbody>
</table>

Results

Study 1- Cost of inspection and testing, defects per hour

<table>
<thead>
<tr>
<th>Activity</th>
<th>Defects [#]</th>
<th>Effort [h]</th>
<th>Of total [%]</th>
<th>Time spent to find one defect [h:m]</th>
<th>Estimated saved time by early defect removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection preparation</td>
<td>928</td>
<td>786.8</td>
<td>61.78</td>
<td>00:51</td>
<td>8196.2</td>
</tr>
<tr>
<td>Inspection meeting</td>
<td>29</td>
<td>375.7</td>
<td>1.92</td>
<td>12:57</td>
<td></td>
</tr>
<tr>
<td>Unit test</td>
<td>404</td>
<td>1257.0</td>
<td>26.91</td>
<td>03:07</td>
<td></td>
</tr>
<tr>
<td>Function test</td>
<td>89</td>
<td>7000.0</td>
<td>5.92</td>
<td>78:39</td>
<td></td>
</tr>
<tr>
<td>System test</td>
<td>17</td>
<td>-</td>
<td>1.13</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Field use</td>
<td>35</td>
<td>-</td>
<td>2.33</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Results

Study 1 - Time usage for inspections

- **Time usage for inspections**
  - Spent: 1474 hrs; whereof 1162.5 for individual reading and inspection meeting.
  - Planned: 2723 hrs, according to internal Gilb guidelines
  - Saved: 8200 hrs
    - if the defects had not been detected by inspection, but detected and repaired in the later phases (not field-use).
  - Inspections detect almost 65% of the registered defects, and unit test 27%. The remaining 7% is found in the later testing activities and 2% in field use!!
  - At Ericsson, the Gilb inspection process focuses on finding new defects in inspections, but only 2% of the defects are actually found in the inspection meeting (true negative). No data stored to verify "false positives".

Results

Study 1 - Number of pages inspected and defect detection rate
Results

Study 1 - Preparation rate and defect detection rate

There does not seem to be a well-defined correlation between these two variables. The dashed line shows the expected (intuitively) values.

Study 1 - field-use defects and defects found during the inspections per module
Results

Study 1: defects found in inspection and number of states in a module

Again, there does not seem to be a well-defined correlation between these two variables. Surprisingly, the no. of defects detected in inspections seems to be rather constant if the topmost value along the y-axis is removed. Intuitively, it should be more difficult to inspect a document with a large number of states, than with a small number of states. The dashed line shows the expected (intuitively) values.

Results

Study 1- number of defects in field-use versus states in a module

The number of states do seem to be correlated with number of defects in field-use, indicated with the dashed line. The number of system failures increases with increasing number of states. Thus, number of states represent the inherent complexity of a module.
**Results**

Study 1 - defects found in unit test versus states in a module

Surprisingly, the number of defects found in unit test seems to be independent of the number of states in a module. This is indicated in the figure below, if the topmost value along the y-axis is not considered.

![Graph showing relationship between defects found in unit test and number of states](chart.png)

**Regression Analysis**

Study 1 - Hypothesis 1

The number of states ($N_s$) is an important variable, because it correlates to the number of system failures (field-use defects, $N_{fu}$). The modification rate, $N_{mr}$, is included in the following regression equation.

$$N_{fu} = \alpha + \beta N_s + \lambda N_{mr}$$

Here $\alpha$, $\beta$, $\lambda$ are constants.
**Regression Analysis - study 1**

- **H₀**: the fault density of a module in field-use depends on the complexity of the module (no. of states) and the its modification rate. 
  High complexity and high modification rate will thus result in high fault density in operation.
- **H₁**: the fault density of a module in field-use does not depend on the module’s complexity and the modification rate.
- **H₀** is the null hypothesis, and **H₁** is the alternative hypothesis.

**H₀** can only be accepted, if \( \beta \) and \( \lambda \) are significantly different from zero and the significance level for each of the coefficients is better than 0.10.

---

**Regression Analysis - study 1**

The following values are estimated:

\[
N_{fu} = -1.73 + 0.084N_s + 0.097N_{mr}
\]

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coefficient</th>
<th>StdDev</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant (( \alpha ))</td>
<td>-1.732</td>
<td>1.067</td>
<td>-1.2</td>
<td>0.166</td>
</tr>
<tr>
<td>States (( \beta ))</td>
<td>0.084</td>
<td>0.035</td>
<td>2.38</td>
<td>0.063</td>
</tr>
<tr>
<td>Modrate (( \lambda ))</td>
<td>0.097</td>
<td>0.034</td>
<td>2.89</td>
<td>0.034</td>
</tr>
</tbody>
</table>

The estimated values for the coefficients are given above, along with the std. deviations, t-value for testing if the coefficient is 0, and the p-value for this test.

The analysis of variance is summarised below:

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>2</td>
<td>26.68</td>
<td>13.34</td>
<td>9.96</td>
<td>0.018</td>
</tr>
<tr>
<td>Error</td>
<td>5</td>
<td>7.30</td>
<td>1.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>7</td>
<td>33.98</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It should be noted that the constant is not significant, but the states and modrate are significant. **H₀** can be accepted.
Results

Study 2 - total defects found

- Study 2:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Defects [#]</th>
<th>[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspection preparation</td>
<td>4478</td>
<td>71.1</td>
</tr>
<tr>
<td>Inspection meeting</td>
<td>392</td>
<td>6.2</td>
</tr>
<tr>
<td>Desk check</td>
<td>832</td>
<td>13.2</td>
</tr>
<tr>
<td>Emulator test</td>
<td>598</td>
<td>9.5</td>
</tr>
<tr>
<td>Total</td>
<td>6300</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Results

Study 1 - are inspections performed at recommended rates?
Regression Analysis
Study 2- Hypothesis 2

Hypothesis 2, uses the data presented above, and checks whether there exist a correlation between defects found during inspection/test and complexity for a module.

The regression equation used to state this hypothesis can be written as:

\[ Y = \alpha X + \beta, \]

where \( Y \) is defect density, \( X \) is the complexity and \( \alpha \), and \( \beta \) are constants.

\( H_0 \) can only be accepted if \( \alpha \) and \( \beta \) are significantly different from zero and the significance level for each of the coefficients is better than 0.10.

---

The following values was estimated: \( Y = 0.1023X + 13.595 \)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Estimate</th>
<th>Standard error</th>
<th>t</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta )</td>
<td>13.595002</td>
<td>18.52051</td>
<td>0.73</td>
<td>0.4720</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.1022985</td>
<td>0.093689</td>
<td>1.09</td>
<td>0.2901</td>
</tr>
</tbody>
</table>

It indicates that the linear regression line must be rejected if a significance of level 0.10 is assumed, i.e. \( H_0 \) must therefore be rejected.
Regression Analysis
Study 2- Hypothesis 3

To check for correlation between defect densities across phases and deliveries, we have analyzed the correlation between defect densities for modules over two projects.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Defect density - Project A</th>
<th>Defect density - Project B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect density - Project A</td>
<td>1.0000</td>
<td>0.4672</td>
</tr>
<tr>
<td>Defect density - Project B</td>
<td>0.4672</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

With a correlation coefficient of 0.4672, we cannot conclude that there exists a correlation between the two data set. We had only 6 modules with complete data for both projects for this test.

Conclusions 1(2)

The data analysis indicates:

- Inspections proved to be the most cost-effective process in defect detection, function testing the least effective.
- Inspections find 70% of the recorded defects, cost 10% of the development time, and yield an estimated saving of 20%.
- 8% (study 1: 2% and study 2: 6%) of the defects are found during the final meeting, 92% during the individual reading.
- But inspection meetings are more cost-effective than function test.
- Individual inspections and individual desk reviews are the most cost-effective techniques to detect defects.
- The recommended inspection rates are not followed, only 2/3 of the recommended time is spent.
Conclusions 2(2)

- The defect density of a module in field-use is depends on the complexity and the modification rate. The literature indicates that designs with lower complexity lead to lower defect rates. However, is it possible to create designs with low complexity in the area of real-time telecom software? Maybe the solution is to pay extra attention when designing the most complex parts of such a system?
- Finding new defects during meetings has been focused at Ericsson. Only 8% (study 2) of defects found in inspections are found here.
- The defect classification is too coarse

Recommendations

Record the correct data properly, and later analyse the data to answer questions and test hypotheses regarding e.g.

Is defect-prone module during inspection/test also defect-prone during field use?
- If one type of defect dominates one project, will the subsequent projects have the same type of defect or will these be eliminated?
- Does the process lead to find the of serious defects during reading and meetings?
- The defects found during the inspections should be classified (categories from basic test can be applied, or follow ISO/IEEE recommendations) to find out which type of defects are found in inspections.
- One should consider omitting the inspection meetings for some document types and may be perform asynchronous inspection ‘meetings’ (by utilising the web technology).