Fault Reporting Processes in Business-Critical Systems

Jon Arvid Børretzen
Dept. Computer and Information Science (IDI)
University of Science and Technology (NTNU)
Trondheim, Norway

PhD thesis presentation, December 17th 2007
Outline

• **Background and motivation**
  – Motivation
  – Definitions
  – The BUCS Project

• Research Goals and Process

• Contributions

• Summary
Motivation of the study

• The number of software systems we depend on in daily life is steadily increasing
  – In a car ~10 million LOC
  – In a mobile phone >5 million LOC
• Especially many that are of an economic nature: Business-Critical
• Obtain better understanding of such systems (BUCS project)
• Focusing on Software Faults and Fault Reporting
Definitions

• **Business-critical software**: Software whose failure could threaten the stability of a business.

• **Fault**: Potential flaws in a software system (i.e. incorrect versus explicitly stated requirements).

• **Failure**: Erroneous external behaviour.

• **Fault reports**: Logs of reported faults and failures used to remove these problems.
BUCS Project research context

- **BUCS** – Business-Critical Software (2003-2007)
- Financed by the Norwegian Research Council under the IKT-2010 program, project number 152923/V30
- Based on cooperation with Norwegian IT industry
- 3 PhD positions, budget of NOK 5.6 million
Outline

• Background and motivation

• Research Goals and Process
  – Overview and Progress
  – Research Questions
  – Research methods

• Contributions

• Summary
Overview of the studies

Phase 1
- Study 1: Preliminary Interview Study 2003
- Study 2: Literature study of safety methods 2004

Phase 2
- First fault report analysis study 2005
- Second fault report analysis study 2006

Phase 3
- Interviews on fault reports 2007
- Assessing hazard analysis vs. fault report analysis 2007

June 2003

Quantitative study

C
Contribution

P
Paper

Input

Industrial cooperation

NTNU
Innovation and Creativity
Research questions

RQ1. What is the role of fault reporting in existing industrial software development?

RQ2. How can we improve existing fault reporting processes?

RQ3. What are the most common and severe fault types, and how can we reduce them in number and severity?

RQ4. How can we use safety analysis techniques together with failure report analysis to improve the development process?
Applied research methods

- Literature studies
  - Background studies
- Data mining of historical projects
  - Much available and unused data material
  - Convenient for industrial cooperation
- Structured interviews
  - Practitioners perspective and input
- Case study
  - Exploratory study
Contributions

C1. Describing how to utilize safety criticality techniques to improve the development process for business-critical software. (RQ3, RQ4) (P1, P7)

C2. Identification of typical shortcomings in fault reporting. (RQ1, RQ2) (P3, P4, P6, P8)

C3. Improved model of fault origins and types for business-critical software. (RQ1, RQ2, RQ3) (P2, P4, P5, P6, P7)
# Studies performed

<table>
<thead>
<tr>
<th>Study</th>
<th>Description</th>
<th>Type</th>
<th>Paper</th>
<th>Gave input to study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interviews with company representatives</td>
<td>Qualitative, explorative</td>
<td>(P1)</td>
<td>Study 3</td>
</tr>
<tr>
<td>2</td>
<td>Literature study</td>
<td>Qualitative, descriptive</td>
<td>P1</td>
<td>Study 6</td>
</tr>
<tr>
<td>3</td>
<td>Fault report study of four projects</td>
<td>Quantitative, explorative</td>
<td>P2, P3, P5</td>
<td>Studies 4, 7</td>
</tr>
<tr>
<td>4</td>
<td>Fault report study of five projects</td>
<td>Quantitative, confirmative</td>
<td>P4</td>
<td>Studies 5, 6, 7</td>
</tr>
<tr>
<td>5</td>
<td>Fault reporting and management interviews</td>
<td>Qualitative, confirmative</td>
<td>P6</td>
<td>Study 7</td>
</tr>
<tr>
<td>6</td>
<td>Hazard analysis vs. fault report analysis - DAIM</td>
<td>Quantitative and qualitative</td>
<td>P7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Fault management and reporting</td>
<td>Qualitative, descriptive</td>
<td>(P8)</td>
<td></td>
</tr>
</tbody>
</table>
Studies performed (1)

- **Study 1 (2003):** Interviews with companies about business-critical software
  - Goal: To get an overview of the state-of-the-practice
  - Results: Description of industry’s view of “business-critical”, their activities and challenges

- **Study 2 (2004):** Combining safety methods for use in business-critical systems development
  - Goal: Showing possible integration of common development methods with techniques used in safety-critical systems development
  - Results: Description of technique and tool integration, suited for early development process phases.
Studies performed (2)

• Study 3 (2004-2005): Fault report study of industrial projects in 4 companies
  – Goals: Study fault distribution in industrial business-critical software
    • CBSE studies (cancelled in early 2005 due to lack of relevant data)
  – Results: Identifying numerous and severe fault types, and describing experience with fault reports for quality assessment

• Study 4 (2006): Fault report study of 5 industrial projects in 1 company
  – Goal: Expand on the knowledge gained from Study 3
  – Results: Confirming findings from Study 3, a large part of the faults reported originate from early development phases
Studies performed (3)

• Study 5 (2007): Interviews on fault management (same organization as study 4)
  – Goal: Further explain findings in study 4
  – Results: Common shortcomings and potential improvements in fault reporting and management were elicited

• Study 6 (2007): Using hazard identification to identify faults
  – Goal: To investigate how hazard identification could be used to elicit potential faults in software
  – Results: Description and implementation of method for identifying faults using PHA. Quantitative results from a case study.
Studies performed (4)

- Study 7 (2005-2007): Experiences from fault report studies
  - Goal: Describe our experiences from several fault studies
  - Results: Identifying common shortcomings in fault reporting, suggesting actions to improve and support fault report analysis as a tool for process improvement
Outline

- Background and motivation
- Research Goals and Process
- Contributions
- Summary
Contribution C1

- We have described ways of integrating safety criticality techniques with regular development practices to improve the development process for business-critical software.
- We have proposed integrating safety techniques like PHA and HazOp into early development phases in order to help improve safety and reliability of the resulting software.
- In addition we have shown that the PHA technique is useful in eliciting hazards that are related to faults that are introduced in early development process phases.
Contribution C2

- We have described several issues concerning shortcomings in fault reporting.
  - The most striking is that commercial organizations generally do not exploit the fault report data they possess for more than day-to-day fault logging or at most shallow analysis.
  - Fault reporting is treated more as a necessary chore, than as a potential source for process improvement.
  - In addition fault data that could easily have been recorded for process improvement gains, e.g. correction effort or location of fault, are not even considered in fault reports.
Contribution C3

- Improved model of fault origins and types for business-critical software.
- The studies have given insight in what fault types are most common or severe in some business-critical software projects.
  - The most common faults were ones that originated from early process phases, namely specification and design.
  - Also described how severity of faults differ according to fault types.
Outline

• Background and motivation
• Research Goals and Process
• Contributions
• **Summary**
  – Answering the research questions
  – Discussion
  – Conclusions
  – Future work
Answering research questions (1)

• **RQ1.** *What is the role of fault reporting in existing industrial software development?*
  – Fault reporting seems to generally be underused and undervalued.
  – All software developing organizations have a fault reporting system in operation, but its use differs substantially.
  – Even where fault report data is thoroughly recorded and stored, it is not systematically used as a tool for software process improvement.

• **RQ2.** *How can we improve on existing fault reporting processes?*
  – Consciousness about potential for improvement by analyzing fault reports.
  – We need more formalized reporting schemes, and clearly defined procedures for reporting faults.
  – Introduce updated fault reporting schemes (fault type, severity, priority, effort, location etc) for the organization’s needs.
Answering research questions (2)

- **RQ3.** *What are the most common and severe fault types, and how can we reduce them in number and severity?*
  - We have found that the most common fault type is the “function” fault type, i.e. faults related to faults in the specification and design phases of development.

- **RQ4.** *How can we use safety analysis techniques together with failure report analysis to improve the development process?*
  - We have found that the PHA technique is useful for eliciting hazards that can be related to faults that are commonly introduced in early development phases.
Discussion

• Validity of findings in general
  – Replication of fault report studies
  – Triangulation through use of different research strategies

• Validity threats in individual studies
  – Data quality
  – Number of studied projects
  – Influence of subjective opinions (fault classification and interviews)
    • Formulation of questions
Discussion

• Relevance
  – Cooperation with Industry
    • Knowledge feedback and validation
  – Existing research
    • Little work done on “business-critical systems” as such
    • Contributions aimed at processes where reliability is important
  – Why these efforts especially for Business-Critical Software?
    • Consequences of failures can be severe
    • Introduce simple but targeted initiatives/techniques
    • Does incur an extra cost, need to evaluate the benefit
Conclusions (1)

• **Fault reporting as a tool for process improvement**
  – Quantification of fault profiles
  – Unexploited data resource
  – Large room for improvement using little effort

• **Empirical findings**
  – Most common and severe fault types
  – Identified development phases which contribute expensive and numerous faults
Conclusions (2)

• **Software safety and reliability from a fault perspective**
  – Fault management and reporting in the software development process may pinpoint areas of improvement in terms of software safety and reliability
  – Proposed how to utilize techniques taken from safety analysis in a business-critical setting
  – Our conclusion is that such techniques should be used early in the development phases, both because suitable techniques like PHA works well in early process phases, and also because identifying and correcting faults early is more efficient than correcting them in later phases.
Future work

• Empirical fault studies from all phases of the development process.
• Studies on fault reports including information on fault fixing effort and fault location, as well as fault type and severity.
• Further studies on the use of hazard analysis to identify faults.
• More work on the relationship between faults and hazards.
Acknowledgements

- Thanks to BUСS and NTNU for financial support.
- Thanks to my supervisors, Prof. Reidar Conradi, Prof. Tor Stålhane, and colleagues at BUСS and IDI.
- Thanks to the organizations I have cooperated with for the permission to perform studies and to publish the results.
Thank you for listening ☺
Papers

P1. Safety activities during early software project phases
P2. A study of Fault Reports in Commercial Projects
P3. Revisiting the Problem of Using Problem Reports for Quality Assessment
P4. Investigating the Software Fault Profile of Industrial Projects to Determine Process Improvement Areas: An Empirical Study
P5. The Empirical Studies on Quality Benefits of Reusing Software Components
P6. Fault classification and fault management: Experiences from a software developer perspective
P7. Using Hazard Identification to Identify Potential Software Faults: A Proposed Method and Case Study
(P8). Diverse Fault Management – a comment and prestudy of industrial practice