HazOp in business critical software development

TORGRIM LAURITSEN

NTNU
Trondheim, Norway

Newcastle University
Newcastle, United Kingdom
Overview

- Background
- Hazard and Operability analysis (HazOp)
- Experiment
- Results
- Threats to validity of the experiment
- Conclusion
Businesses rely more and more on IT systems.

Software must
- have a low probability of causing loss of assets and reputation
- avoid business interruption
- give high Return Of Investment (ROI)

We will use the term “business safe software” for this characteristic.
Background - II

Safety critical industries such as avionics, automotive, chemical, nuclear power, medicine, etc.

Safety analysis techniques
- identify weak areas of the design
- stimulate design changes that will improve the safety

Evaluate if safety analysis techniques can help to develop business safe software.
Hazard and Operability analysis (HazOp) - I

Analyse design documents and identify possible problems that can arise during the operation and maintenance of the software.

HazOp is a group activity with a leader, secretary, customer with domain knowledge and software developers.

HazOp has two structuring devices: Guide words and HazOp table to help the group to identify hazards, causes and consequences and possible solutions.

The guide words should be used on the study nodes – points in the system where we want to focus.

Usually study points are points where
- the system interacts with its environment – e.g. users input
- two or more parts of the system exchange information – e.g. a network connection
# Hazard and Operability analysis (HazOp) - II

<table>
<thead>
<tr>
<th>Guideword</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>No / none</td>
<td>Message not sent when it should be</td>
</tr>
<tr>
<td>Other than</td>
<td>Message sent at wrong time</td>
</tr>
<tr>
<td>In addition to</td>
<td>Message sent at correct time and also at an incorrect time</td>
</tr>
<tr>
<td>More than</td>
<td>Message sent later/more often than intended</td>
</tr>
<tr>
<td>Less than</td>
<td>Message sent earlier/less often than intended</td>
</tr>
<tr>
<td>Earlier</td>
<td>Message sent earlier within message sequence than intended</td>
</tr>
<tr>
<td>Later</td>
<td>Message sent later within message sequence than intended</td>
</tr>
</tbody>
</table>

Guide words used on a sequence diagram (Time focus)
<table>
<thead>
<tr>
<th>Guide word</th>
<th>Study node</th>
<th>Causes</th>
<th>Consequences</th>
<th>Possible solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other than</td>
<td>Buyer sends an item request</td>
<td>Mismatch of what is perceived and what is ordered</td>
<td>Wrong items will be sent to the customer</td>
<td>Insert check to ensure correct information</td>
</tr>
<tr>
<td>Other than</td>
<td>Buyer signs for order</td>
<td>Could be a imitation of the signature -&gt; fraud</td>
<td>Company will lose money</td>
<td>Insert barrier that check signature</td>
</tr>
<tr>
<td>No</td>
<td>Buyer pays invoice</td>
<td>Company receives no money for the items sold.</td>
<td>Company do not make profit from the sale.</td>
<td>Implement a credit check and log who placed the order</td>
</tr>
<tr>
<td>.....</td>
<td>.....</td>
<td>.....</td>
<td>.....</td>
<td>.....</td>
</tr>
</tbody>
</table>
Experiment - I

Evaluate if **HazOp** can help to develop business safe software.

- Does HazOp find more hazards than ad hoc brainstorming techniques?
- Will software developers see the benefit of using HazOp?
- Should the customers be involved in the HazOp analysis?
- Can HazOp lead to reduction in testing time?
Experiment - II

- 14 participants from 4 different IT companies in Newcastle upon Tyne, UK.
- 2 companies provides middleware products
- 2 companies work as system and service providers for Newcastle University.
- 1 - 35 years of experience.
- BSC, MSC and PhD degrees.
Experiment - III

We split the participants at each company into two groups
1. Group using HazOp
2. Group using their own ad hoc technique

In the experiment the participants should identify possible hazards, problems and solutions from
- a use case (purchase system, buy goods)
- a sequence diagram (if enough items in stock, remove items, if not, reorder items).
## Results - I

<table>
<thead>
<tr>
<th>Category</th>
<th>HazOp group</th>
<th>Ad hoc brainstorming group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoice / payment problems</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Order problems / reasonability</td>
<td>13</td>
<td>6</td>
</tr>
<tr>
<td>Seller promotes/registers wrong details</td>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>Stock information / problems</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Wrong information from customer</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Deliver / customer address problems</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Return (damaged) goods</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>50</strong></td>
<td><strong>57</strong></td>
</tr>
</tbody>
</table>

Most frequent categories
Neither of the groups found more hazards than the other

Look closer at the answers from the experiment questionnaire.

A systematic technique (like HazOp) is defined in advance and is based on a step-wise process
Results - III

Ad hoc group satisfied with all hazards and problems identified, but dissatisfied with not having any step-wise process to follow.

HazOp group satisfied with the process which is easy to follow and the guidewords “kick starts” the hazard brainstorming process.

HazOp gives a thorough analysis and early identification of problems. Design changes before implementation to prevent hazards to arise.

A systematic technique like HazOp will give extra work effort in the initial phase (analysis and design), but it gives obvious benefits!

The customer should be involved in the safety analysis.

Preventive safety analysis gives actually more testing.
Threats to validity of the experiment

- Real IT professionals
- Participants domain knowledge
- Grouping the participants
- Coding the results from experiment and questionnaire
- Small sample from a realistic environment
Conclusion

Developers of business critical software should use safety critical analysis techniques to make safer and better software.

- How to convince the developers to start using safety critical analysis techniques?
- Standards
- Customer demands