

# Software Safety Risk Evaluation process

---

*Yorick Bouma, 3407020, Group III*

## Notice of Originality

I declare that this paper is my own work and that information derived from published or unpublished work of others has been acknowledged in the text and has been explicitly referred to in the list of references. All citations are in the text between quotation marks (“ ”). I am fully aware that violation of these rules can have severe consequences for my study at Utrecht University.

Signed:

Name: Yorick Bouma

Date: 13-4-2012

Place: Utrecht

## Introduction

The Software Safety Risk Evaluation (SSRE) process is used to identify, analyze, consolidate and mitigate software safety risks (Hill & Tilley, 2010). The purpose of SSRE is to collect all interesting information about the software safety risks, so that this information can be used to make a well informed decision about what to do with a specific system. Depending on the acquired information it can be decided to: accept the risks and continue using the system as is or adapt the system to meet the safety requirements. The domain where this process could be used for according to Hill and Tilley (2010) is the domain of legacy systems. Bennett (1995) informally defines legacy systems as “large software systems that we don’t know how to cope with but are vital to our organization”. These systems could contain safety risks and therefore should be evaluated for future use, the SSRE process can be used for this evaluation.

The whole SSRE process consists of five phases, each with its own deliverables that will support the person that will make the final decision about the legacy system. These phases are: detection, risk specification, assessment, consolidation and mitigation. In the first phase – detection – a list of safety related question should be answered by the participating project team members, this will result in a number of safety risks. In the risk specification phase each of the safety risks will be categorized according to the source of risk impact, resulting in an overview of the spread of the safety risks across the different sources of risk impact. The purpose of the third phase – assessment – is to create a risk profile for every safety risk; this is done by evaluating all individual risks for possible consequence and probability of occurrence. In the consolidation phase similar risks are grouped, and for each of the resulting groups its importance is determined by evaluating the amount of important safety risks in that particular group. In the final phase of the SSRE process plans are made for the groups with the largest amount of important safety risks.

The SSRE process as described in this paper is developed by Hill and Tilley (2010), they developed this process to determine what to do with several legacy systems of NASA. Janice Hill works at the IV&V Facility at NASA in Fairmont, West Virginia, USA. She is a PhD student and her advisor is the co-author, Scott Tilley (Ph.D Student Organization, 2012). Scott Tilley is a professor at the Department of Computer Sciences at Florida Institute of Technology in Melbourne, Florida, USA (Tilley, 2012).

## Example

Since Hill and Tilley (2010) developed this SSRE process in 2010, no actual case studies can be found of this process except the ones they describe in their paper. All their case studies are about legacy systems at NASA, in this example the software safety risk evaluation process will be described with a general legacy system as an example. In this example our legacy system has a above average amount of risks, mainly in the field of safety requirements and documentation.

## Detection

The first step is to detect software safety risks in the project. This is done by asking the participating project team members questions taken from the Software Safety Risk Taxonomy Based Questionnaire (TBQ) (Hill & Victor, 2008). These questions investigate a wide range of software safety to detect as much risks as possible, some examples of these questions can be found in Table 1. As a result a number of risks will be identified; in our example a total of 200 risks are identified.

Sample questions
Was a Preliminary Hazard Analysis (PHA) performed for this system?
Was a System Safety Analysis (SSA) performed for this system?
Are the system and software requirements analyzed for proper flow down from the system level requirements?

Table 1. Software Safety Risk TBQ sample questions

## Risk Specification

In this step of the process all of the risks detected in phase one are labeled according to the source of their risk impact. This way one can easily see which source of risk impact causes the most risks. The following sources of risk impact are defined: performance, support, cost and schedule. Table 2 gives an overview of the number of risks of each label for the project; note that the total doesn't add up to 200 because one risk can have several labels. A template for this step is available in Appendix.

Source of risk impact	Number of risks
Performance	138
Support	89
Cost	133
Schedule	139

Table 2. Number of risks per source of risk impact

## Assessment

The goal of this third step is to create a risk profile for each individual risk detected in phase one. To create such a profile two things should be determined: the possible consequence (Catastrophic, Critical, Marginal or Negligible) and the probability of occurrence (Likely, Probable, Possible, Unlikely or Improbable) of the risk. The product of the possible consequence and the probability of occurrence define the risk's exposure. An example of the results of defining this possible consequence and probability of occurrence for the risks can be found in Table 3; as can be seen a scale of 1 to 7 is used for the magnitude of a risk. A template for this step is available in Appendix.

Rank	Number of risks
1	12
2	15
3	27
4	53
5	39
6	31
7	23

Table 3. Number of risk profiles per risk exposure

## Consolidation

In this fourth step of the SSRE process multiple similar risks are grouped, such a group is called a 'risk area'. Apart from this grouping each individual risk is assigned a rating for how important this risk is for the project. The risks are ranked on a ranking scale from 1 to 5 with decreasing importance; an example result of this step is shown in Table 4. After ranking the individual risks, in each group the risks ranked 1 to 3 are added together. This total number of the three highest ranked risks per group is then divided by the total number of risks; this gives a percentage on which we can sort the most important risk areas in the following step.

Rank	Number of risks
1	40
2	51
3	56
4	26
5	27

Table 4. Number of risks per rank

## Mitigation

In the final step of this process – mitigation – the list, sorted by the percentages calculated in the previous step, is put in use. This list isn't final; the participating project team members are allowed to reorder the risk areas. This list, an example is shown in Table 5, and all other information acquired in the preceding steps is used to create a report for this project. This report contains the approach for the mitigation. The report states for each risk what should be done with it: accept the risk, mitigate the safety risk or decide that the risk is not really a safety risk. For each decision an accompanying justification is given, or a description what adjustments should be done to meet the safety requirements.

Rank	Risk area
1	Safety Requirements
2	Documentation
3	Software Safety Analyses

Table 5. Top three risk areas

## Process-deliverable diagram

The Process-deliverable diagram (Weerd & Brinkkemper, 2008) consists of two integrated diagrams, namely a process view on the left-hand side and a deliverable view on the right hand. These two sides are then connected by arrows to link certain deliverables on the right-hand side as output of activities on the left-hand side. In Figure 1 this diagram is used to give an overview of the Software Safety Risk Evaluation (SSRE) process developed by Hill and Tilley (2010). On the left hand-side the five different phases and their activities are visible; these are further explained in Table 6. On the right-hand side some of the deliverables from the examples are visible; their explanation is in Table 7.

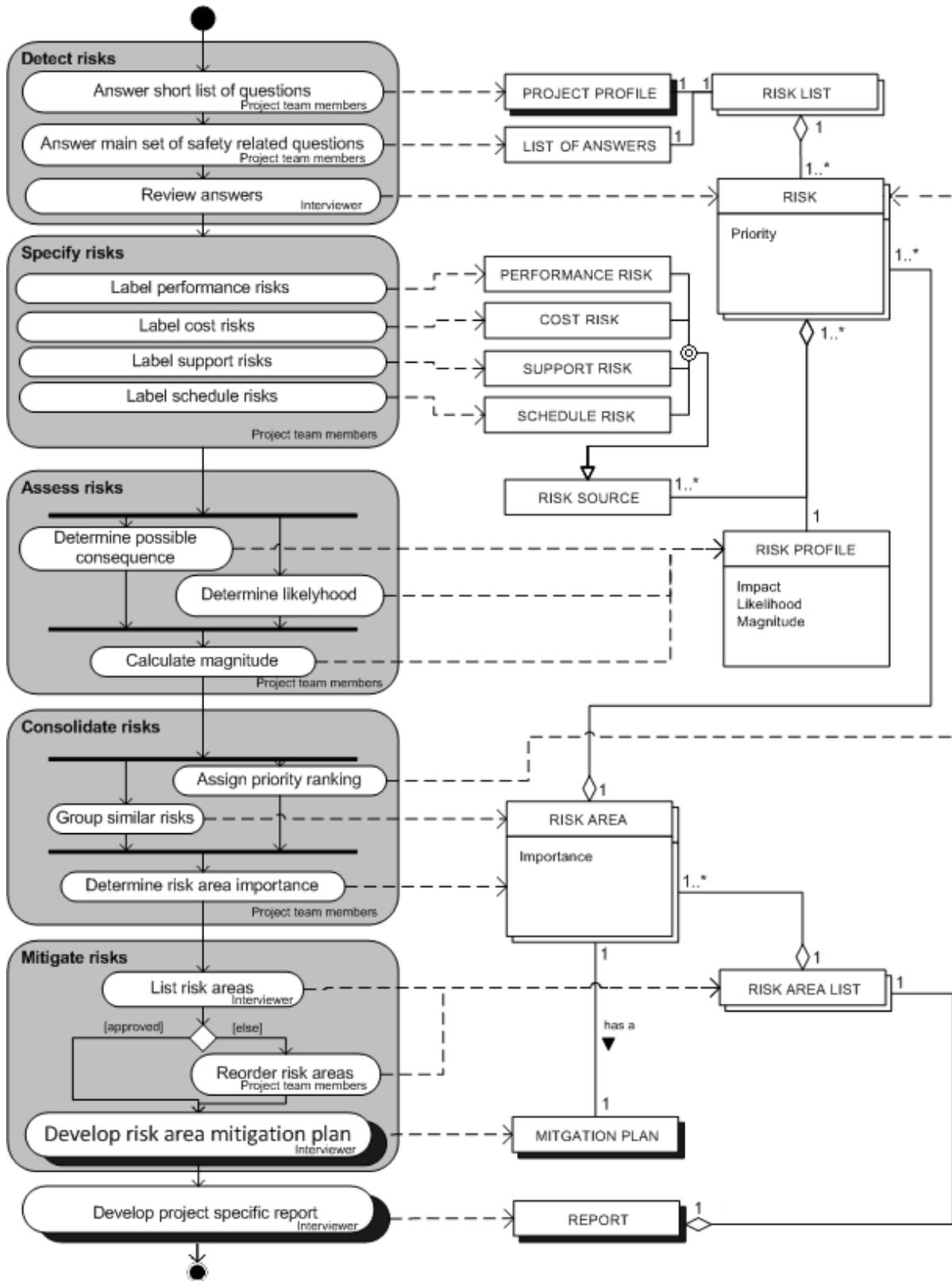


Figure 1. The PDD of the SSRE process

<b>Activity</b>	<b>Sub-Activity</b>	<b>Description</b>
Detect risks	Answer short list of questions	The answers on a short list of questions are used to create a PROJECT PROFILE.
	Answer main set of safety related questions	The answers on a main set of safety related questions are given and collected in the LIST OF ANSWERS. These questions are taken from the Software Safety Risk Taxonomy (Hill & Victor, 2008).
	Review answers	The answers from the LIST OF ANSWERS are reviewed and the answers that point to an issue or concern are entered as RISKS.
Specify risks	Label performance risks	All RISKS that have performance as the source of the RISK impact are labeled. For each RISK a PERFORMANCE RISK as RISK SOURCE is added.
	Label support risks	All RISKS that have support as the source of the RISK impact are labeled. For each RISK a SUPPORT RISK as RISK SOURCE is added.
	Label cost risks	All RISKS that have cost as the source of the RISK impact are labeled. For each RISK a COST RISK as RISK SOURCE is added.
	Label schedule risks	All RISKS that have schedule as the source of the RISK impact are labeled. For each RISK a SCHEDULE RISK as RISK SOURCE is added.
Asses risks	Determine possible consequence	For all RISKS the possible consequence (Likely, Probable, Possible, Unlikely or Improbable) is determined.
	Determine likelihood	For all RISKS the probability of occurrence (Catastrophic, Critical, Marginal or Negligible) is determined.
	Calculate magnitude	For all RISKS the magnitude is calculated. This is done by multiplying the possible consequence with the probability of occurrence of the RISK. The scale of this magnitude is from 1 to 7.
Consolidate risks	Group similar risks	Similar RISKS from different RISK SOURCEs are grouped together which form RISK AREAs.
	Assign priority ranking	To all RISKS a priority is assigned based on how important the RISKS are to the project. This ranking is done a scale from 1 to 5.
	Determine risk area importance	The importance of each RISK AREA is calculated. This is done by taking all RISKS in the particular RISK AREA with priority 1, 2 and 3 and then divide this by the total numbers of RISKS in the RISK AREA.

Mitigate risks	List risk areas	All RISK AREAs are listed in the RISK AREA LIST and ordered on importance of the RISK AREAs.
	Reorder risk areas	After the RISK AREA LIST is reviewed the RISK AREA LIST is optionally changed. This is done by reordering the RISK AREs.
	Develop risk area mitigation plan	For each RISKS AREA a MITIGATION PLAN is developed. This MITIGATION PLAN consists of strategies and activities regarding to the mitigation of the RISK AREA.
Develop project specific report		The data collected (RISK AREAs, RISK AREA LIST and MITIGATION PLANs) in the last two phases (Consolidate risks and Mitigate Risks) is used to develop a project specific REPORT.

Table 6. Activity table of the PDD of the SSRE process

Concept	Description
PROJECT PROFILE	A PROJECT PROFILE is created by the interviewer out of the answers of the project team members on a short list of question. This document consists of general information about the project that is used to get an overview of the project at hand (Hill & Tilley, 2010).
LIST OF ANSWERS	The LIST OF ANSWERS consists of answers that project members give to the main set of safety related questions that are based on the Software Safety Risk Taxonomy, and contained in the TBQ (Hill & Tilley, 2010).
RISK	According to the International Organization of Standardization (2009) RISK is the “effect of uncertainty on objectives”. In the SSRE process this risks are detected by the interviewer when he reviews the answers in the LIST OF ANSWERS from the project team members.
RISK SOURCE	The RISK SOURCE is the source of the RISK impact of a specific RISK, this sources can be performance, support, cost and schedule. This RISK SOURCE is assigned to each RISK by the project team members. It is possible for one RISK to have multiple RISK SOURCEs (Hill & Tilley, 2010).
RISK PROFILE	A RISK PROFILE is the risk exposure (magnitude result) of a RISK. This magnitude result is defined as the multiplication of the possible consequence and the probability of occurrence of a RISK. This defining is done by the project team members (Hill & Tilley, 2010).
RISK LIST	A list containing RISKS that are identified in the Detection phase of the Software Safety Risk Evaluation process by the interviewer (Hill & Tilley, 2010).
RISK AREA	A RISK AREA is a group of RISKS that are similar but can be from different sources of RISK impact, the grouping of these RISKS is done by the project team members (Hill & Tilley, 2010).
RISK AREA LIST	A list containing RISK AREs sorted on the importance of the RISK AREA. This importance is determined in the consolidation phase by the project team members (Hill & Tilley, 2010).
MITIGATION PLAN	A MITGATION PLAN for a specific RISK consists of a preliminary mitigation goal, a key or root cause, a mitigation strategy, mitigation activities, key

	measures and estimations of the resources needed to accomplish the work and is developed by the interviewer (Hill & Tilley, 2010).
REPORT	Hill and Tilley (2010) define a REPORT as: “specific report listing all of the software safety requirements from the NASA Software Safety Standard and the corresponding identified risks” (p. 301). This REPORT is developed by the interviewer with the data collected in the consolidation and mitigation phase.

Table 7. The concept table of the PDD of the SSRE process

## Related Literature

As mentioned in their paper the SSRE process developed by Hill and Tilley (2010) is a slightly modified version of SEI Software Risk Evaluation (SRE) practice (Higuera & Haimes, 1996). This SRE was one of the methodologies used for Software Risk Management (SRM), particularly the maintenance methodologies for SRM.

The questions in the first phase of the process were taken from the Software Safety Risk Taxonomy Based Questionnaire (TBQ) (Hill & Victor, 2008). These questions were based on an earlier paper by Hill (2007) on the topic of specializing Software Development Risk Taxonomy with safety elements and attributes. In this paper she proposes a new Software Safety Risk Taxonomy, which is an extension from the Software Engineering Institute (SEI) Software Development Risk Taxonomy (Carr, Konda, Monarch, Ulrich, & Walker, 1996) and the Risk Taxonomy Proposal for Software Maintenance (Higuera & Haimes, 1996).

In their paper Hill and Tilley (2010) also mention a tool they developed to maintain all the information collected by the SSRE process. This Legacy Systems Risk Database (LSRD) tool provides a framework for the process and has a supporting role in the decision making of the project manager.

An extensive search for literature not mentioned by Hill and Tilley (2010) that positions this topic has yielded nothing. The co-author – Scott Tilley – has not written any papers on this subject before he coauthored the paper of Janice Hill (The DBLP Computer Science Bibliography, 2012).

## References

- Bennett, K. (1995). Legacy Systems: Coping with Success. *Software, IEEE, Volume 12*, 19-23.
- Carr, M., Konda, S., Monarch, I., Ulrich, F., & Walker, C. (1996). *Taxonomy-Based Risk Identification*. Technical Report CMU/SEI-96-TR-6, Software Engineering Institute.
- Higuera, R. P., & Haimes, Y. Y. (1996). Software Risk Management. *Technical Report CMU/SEI-96-TR-012*. Software Engineering Institute.
- Hill, J. (2007). A Software Safety Risk Taxonomy for Use in Retrospective. *Proceedings of the 31st IEEE/NASA Software* (pp. 179-186). IEEE CS Press.
- Hill, J., & Tilley, S. (2010). Creating Safety Requirements Traceability for Assuring and Recertifying Legacy Safety-Critical Systems. *Requirements Engineering Conference (RE), 2010 18th IEEE International*, 297-302.

- Hill, J., & Victor, D. (2008). The Product Engineering Class in the Software Safety Risk Taxonomy for Building Safety-Critical Systems. *Proceedings of the 19th Australian Software Engineering Conference (ASWEC 2008)* (pp. 617-626). IEEE CS Press.
- International Organization for Standardization. (2009). *ISO 31000:2009*. International Organization for Standardization.
- Ph.D Student Organization. (2012). *Ph.D Student Organization - Florida Tech Computer Science Department*. Retrieved on February 17, 2012 from <http://cs.fit.edu/pso/>
- The DBLP Computer Science Bibliography. (2012). *DBLP: Scott R. Tilley*. Retrieved on February 17, 2012 from [http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/t/Tilley:Scott\\_R=.html](http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/t/Tilley:Scott_R=.html)
- Tilley, S. (2012). *Florida Institute of Technology*. Retrieved on February 17, 2012 from <http://www.fit.edu/faculty/profiles/profile.php?value=229>
- Webster, K. B., Oliveira, K. d., & Anquentil, N. (2005). A Risk Taxonomy Proposal for Software Maintenance,". *Proceedings of the 21st IEEE International Conference on Software Maintenance, (ICSM 2005)* (pp. 453-461). IEEE CS Press.
- Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. *Handbook of research on modern systems analysis and design technologies and applications*, 38-58.

## Appendix

Table 8 is a template for the deliverable of the Risk Specification phase and Table 9 is a template for the deliverable of the Assessment phase. These templates are also available as Excel files which automate the process.

Risk	Source of risk impact			
	Performance	Support	Cost	Schedule
1				
2				
...				
<b>Total # risks</b>				

Table 8. Template for the deliverable of the Risk Specification phase

Risk	Possible consequence				Probability of occurrence					Exposure
	Catastrophic	Critical	Marginal	Negligible	Likely	Probable	Possible	Unlikely	Improbable	
1										
2										
...										

Table 9. Template for the deliverable of the Assessment phase