Problems in Requirements Engineering and Requirements Modelling in Practice - Interview Guide

Grischa Liebel

November 16, 2015

Abstract

This document describes the Interview Guide used during the semi-structured interviews at different automotive OEMs and Suppliers. The interviews are planned for one hour. Details about the case study design can be obtained directly from the author.

1 Interview Guide

1.1 Part A, Introduction

- The interviewer shortly presents the topic and the research questions (2-3 min).
- The interviewer points out that the interview is anonymous, covered by an NDA agreement with the respective company.
- The interviewer asks for permission to record the interview, in order to facilitate data analysis later on.
- The interviewer also points out that the interviewee will receive the transcribed interview once it is available and may review it and/or object to some of the stated points.
  In this case, the interviewers may not use these parts of the data.
- Finally, the interviewer points out that the final data analysis for each company will be discussed with the contact person at the respective company and possibly with the interviewees as well, if desired.

1.2 Part B, Demographic Questions

1. In which role do you work at [Company]?
2. Could you shortly introduce your work at [Company]?
3. How long have you worked with Requirements Engineering/Software Engineering/Verification?
4. In which other areas at [Company] have you worked before?
5. How long have you worked overall in the automotive industry?

1.3 Part C, Topic Questions

1. Could you shortly describe the Requirements Engineering process at [Company] based on your role?
   Comment: E.g. from a Requirements Engineer's point of view, this will be the elicitation process, from a developer's view more the exchange/usage of requirements, etc.

2. When do you elicit new requirements within [Company]?
   For non requirements engineers: Their point of view, their role

3. How do you elicit new requirements within [Company]?
   For non requirements engineers: Their point of view, their role
   Quality requirements?

4. How are requirements documented?
   Req. Eng.: How do you do it?; SE: How are they provided to you?

5. How do you specify variability in your requirements?
   All roles: How is it done right now? Req. Eng.: Where does the variability come from? Who decides on the variants?

6. (How) Do you reuse requirements?
   SE/Verification: Do you often get similar requirements? Do they seem to be copied or follow a similar schema?

7. Is there a glossary of terms which have to be used during specification or guidelines to follow for writing the specification?
   Probably implicit?

8. At which level of abstraction are requirements specified?

9. How is the right level of abstraction determined?
   Are there any policies/rules? Is it up to the Req. Eng.?

10. How much does the level of abstraction change throughout the process? When are the requirements refined?
    Goes back to the second question, different phases/iterations.

11. Is this the right abstraction, in your opinion?

12. How is the specification used?
    During requirements engineering, during negotiation, during development, testing, etc.?

13. How are requirements traced to/from later artifacts?
    Unidirectional traces? Bidirectional?

14. How do you handle ambiguous or unclear requirements?
    Req. Eng.: How do you avoid them?, Developer/Tester: What do you do with them?

15. Do you know of any problems you had in the past with ambiguous/unclear requirements?

16. Do you have ways to measure the quality of your specification? Which ones?

17. Are you using any kind of models for specifying or clarifying requirements within [Company]?
Yes:

    (a) What is the purpose of these models?
    Non-functional properties as well?

    (b) Are the models used exclusively or in combination with NL requirements?

    (c) What are the (dis-)advantages of this method?

    (d) Is this use of models widespread?

    (e) Which aspects of modelling/models do you like/dislike in particular?

    (f) How do you provide tracing to elements in these models?

    (g) For which additional purposes could models be benefitial?

    No:

    (a) Why do you not use any models?

    (b) What would have to change in order for you to introduce models?

    (c) Do you think it could be benefitial to use some kind of model?

    (d) For which purposes?
    Only functionality, or also non-functional properties?

18. Which tools, languages, formalisms would you like to use during requirements engineering? Why?
What might possible risks of doing so be?

19. Which ones wouldn't you like to use? Why not?

1.4 Part D, Finish

- The interviewer thanks for the participation
- Additionally, the interviewer encourages the interviewee to come back with any comments or questions after the interview, if they may arise.

2 Definitions

1. **Abstraction**
"(1) A view of an object that focuses on the information relevant to a particular purpose and ignores the remainder of the information. " - [1]

2. **Ambiguity**
"A word or expression that can be understood in two or more possible ways." - [2]

3. **Elicit**
"To call forth or draw out (as information or a response)" - [2]

4. **Model**
   "A formal representation of entities and relationships in the real world with a certain correspondence for a certain purpose" [3]

5. **Requirement**
   "A requirement is: [..]
   (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document." - [1]

6. **Reuseability**
   "The degree to which a software module or other work product can be used in more than one computer program or software system." - [1]

7. **Traceability**
   "(1) The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another." - [1]

8. **Variability**
   "Variability is the ability to change or customize a system. Improving variability in a system implies making it easier to do certain kinds of changes. It is possible to anticipate some types of variability and construct a system in such a way that it facilitates this type of variability." - [4]

References

[1] Ieee standard glossary of software engineering terminology. *IEEE Std 610.12-1990* (Dec 1990), 1–84.

[2] Merriam-Webster Inc. Merriam-webster online: Dictionary and thesaurus. `www.merriam-webster.com`, Sept. 2014.

[3] Stachowiak, H., Ed. *Allgemeine Modelltheorie*. Springer, Wien [u.a.], 1973.

[4] van Gurp, J., Bosch, J., and Svahnberg, M. On the notion of variability in software product lines. In *Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on* (2001), pp. 45–54.