**Requirements Engineering II**

Martin Glinz, Professor, Dr. rer. nat.
Samuel Fricker, Eya Ben Charrada

## Assignment 1: Requirements Traceability

### 1. Tasks

- Read the mandatory items in the reading list
- Be prepared to answer the questions given below in class
- Prepare a 15 minutes presentation (5-10 slides) on the theme assigned to your course group. Browse/read additional papers and/or web pages where necessary.
- Generate a requirements-to-stakeholders traceability matrix for the requirements document that you produced in exercise 2 of Requirements Engineering I.

### 2. Reading list

### Mandatory reading

[Jarke 1998] and [Dick 2005] provide a motivation and introduction. [Ramesh and Jarke 2001] establishes reference models for traceability. [Fricker et al 2009] describes how traceability is used for requirements communication.

### Theme-specific reading

[Gotel and Finkelstein 1994]: first systematic treatment of the traceability problem.
[Cleland-Huang et al 2007], [Huffman Hayes 2005]: automating traceability using trace analysis.
[von Knethen, Grund 2003], [Mäder et al 2008]: use and maintenance of post-requirements traceability.

### 3. Questions

- What is requirements traceability?
- What is the benefit of requirements traceability and what does it cost?
- How can one establish and maintain traces?
- What is the role of tools?

### 4. Themes for presentation

(Will be assigned by the research assistant who tutors this course; your group can apply for the theme you would like to work on)

A. An overview of the requirements traceability problem: what is traceability? Why is it needed? What are the problems related to traceability? How can and should traceability be dealt with? Did the understanding of traceability evolve over time?
B. Automated requirements traceability: how can traceability information be generated by software? Can and should humans be replaced for defining traceability?
C. Post-requirements traceability: what is post-requirements traceability used for? How can and should such information be maintained?

## References

Cleland-Huang, J., R. Settimi, E. Romanova, B. Berenbach, S. Clark (2007). Best Practices for Automated Traceability. *IEEE Computer* **40,** 6 (June 2007). 27-35.

Dick, J. (2005). Design traceability. *IEEE Software* **22**, 6 (Nov./Dec. 2005). 14-16.

Fricker, S., T. Gorschek, C. Byman, A. Schmidle (2009). Handshaking: Negotiate to Provoke the Right Understanding of Requirements. Accepted for *IEEE Software*.

Gotel, O., A. Finkelstein (1994). An Analysis of the Requirements Traceability Problem. *1st International Conference on Requirements Engineering*, Colorado Springs. 94-101.

Huffman Hayes, J. A. Dekhtyar, S. K. Sundaram (2005). Improving After-the-Fact Tracing and Mapping: Supporting Software Quality Predictions. *IEEE Software* **22**, 6 (Nov./Dec. 2005). 30-37.

Jarke, M (1998). Requirements Traceability. *Communications of the ACM* **41**, 12 (Dec. 1998). 32-36.

Von Knethen, A., M. Grund (2003). QuaTrace: A Tool Environment for (Semi-)Automatic Impact Analysis Based on Traces. *International Conference on Software Maintenance*, 2003. 246-255.

Mäder, P., Gotel, O., Philippow, I. (2008). Rule-Based Maintenance of Post-Requirements Traceability Relations. *16th IEEE International Conference on Requirements Engineering*, Barcelona. 23-32.

Ramesh, B., M. Jarke (2001). Toward Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering* **27**, 1 (Jan 2001). 58-92.