



# Assignment 4

## Traceability and Requirements Evolution

*Prof. Dr. Martin Glinz*

*Dr. Norbert Seyff*

*Dr. Eya Ben Charrada*

## I. Tasks

### Individual Tasks

- Read the mandatory items in the reading list
- Prepare two questions about each paper to ask your classmates. These questions can, for example, be about aspects of the paper that are not clear to you, or about your classmate's opinion on interesting aspects.
- Be prepared to give a short summary of each paper in class. This summary should address the following questions:
  - What is the main message of the paper?
  - What are the expected benefits of the proposed method or the paper in general?
  - What are weaknesses of the paper in your opinion?
- Be prepared to answer the questions given in Sect. III below in class

### Group Tasks

- Prepare a 10-12 minutes presentation (plus 6-8 minutes of discussion) on the theme assigned to your course group (cf. Sect. IV) and choose two persons from your group to present it.
  - At the beginning of your presentation, relate your topic to the session's topic (as represented by the mandatory reading)
  - Browse/read additional papers and/or web pages where necessary.
  - Send your presentation to Norbert and Eya after the session to share it with others.

## II. Reading List

### Mandatory reading

[Jarke 1998] and [Dick 2005] motivate and introduce traceability, while [Ramesh 2001] establishes reference models for it.

### Optional Reading

Traceability was first systematically discussed and analyzed in [Gotel 1994], thus establishing this area of research.

### Theme-specific reading

[Cleland-Huang 2010], [Hayes 2007]: Advanced approaches for generating traceability links.  
[Delater 2013], [Eaddy 2008]: Tracing requirements to source code.  
[von Knethen 2003], [Ben Charrada 2012]: Traceability for impact analysis and co-evolution.

### III. Questions

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

### IV. Themes for Presentation

Themes will be assigned by the assistant who tutors this course; your group can apply for a theme.

#### A. Advanced Approaches for Generating Traceability Links

How can traceability links be automatically generated? How effective are current traceability link recovery techniques? Can and should humans be replaced for defining traceability?

#### B. Tracing Requirements and Source Code

What are the benefits of tracing requirements to source code? How can source code be traced to the requirements during the development phase and after it?

#### C. Traceability for Impact Analysis and Co-Evolution

What is post-requirements traceability used for? How can traceability support the evolution of software systems?

### References

Ben Charrada, E., A. Koziolk, M. Glinz (2012). Identifying Outdated Requirements Based On Source Code Changes. *Proceedings of the 20th International Requirements Engineering Conference (RE 2012)*.

Cleland-Huang, J.; Czauderna, A.; Gibiec, M.; Emenecker, J. (2010). A machine learning approach for tracing regulatory codes to product specific requirements. *International Conference on Software Engineering*, vol.1, pp.155,164,

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

Delater, A., B. Paech(2013). Tracing Requirements and Source Code during Software Development: An Empirical Study. *ESEM 2013*: 25-34

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

Hayes, J. H., A. Dekhtyar, S. Sundaram, E. Holbrook, S. Vadlamudi, A. April (2007). REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery. *ISSE 3(3)*: 193-202.

Eaddy, M., A. Aho, G. Antoniol, Y. Guéhéneuc: CERBERUS: Tracing Requirements to Source Code Using Information Retrieval, Dynamic Analysis, and Program Analysis. *ICPC 2008*: 53-62

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.

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