

Requirements Traceability an Overview

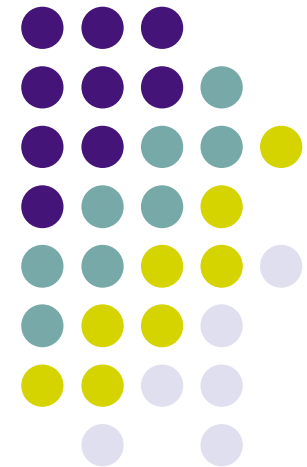
within: Requirements Engineering II

by

Reto Zenger (05-709-597)

Michael Küchler (06-712-681)

02.11.2009





Agenda

- What is traceability
- Why is traceability needed
- What are the problems related to traceability
- How should traceability be dealt with
- Evolution of traceability over time



What is traceability

- *Traceability is about **documenting the relationships** between layers of information - for instance, between system requirements and software design.* (Dick 2005)
- *Requirements traceability, then, is defined as the **ability to describe and follow the life of a requirement**, in both a forward and backward direction, ideally through the whole systems life cycle.* (Jarke 1998)



Pre-traceability

- Pre-traceability links business and IT
- Traceability from stakeholder forward to requirements
 - From which stakeholder is a requirement originated
- Traceability from requirements backward to stakeholder
 - Where does a requirement contribute to the fulfillment of a stakeholders needs



Post-traceability

- Post-traceability links requirements to solutions
- Traceability from requirements forward to system components
 - Which component contributes to the fulfillment of the requirement
- Traceability from system components backward to requirements
 - Why is a system component necessary

Interpretations of traceability



- View of low-end users
 - Traceability as defense against criticism and liability lawsuits
- View of high-end users
 - Traceability as an investment in corporate knowledge asset management within and beyond information systems engineering



Why is traceability needed

- Traceability makes connections between rationales and results explicit
 - Consequences of rationales are easily assessable
 - Decisions and results can be traced back to their cause
- Traceability is likely to lead to improved software process maturity



Benefits of traceability

- Increased software quality
- Increased understanding of the decisions and the design of the solution
- Increased efficiency
 - Possibility of tool-support in impact analysis
 - Avoidance of “gold plating”, i.e. detection of unnecessary features
 - Conflicts / errors are found easier and earlier

Problems related to traceability



- Customers view:
 - Traceability necessitates an extra effort from both, the customer and the supplier, but only the supplier experiences a direct revenue
 - The customer may not know exactly what information the supplier needs
 - Traceability must be maintained through the whole software life cycle, but often it is not
 - Relevant information needs not to be accessible

Problems related to traceability



- Suppliers view:
 - Uncertainty about quality of traceability information
 - Uncertainty about completeness of traceability information
 - Uncertainty about relevance of traceability information

How can traceability be dealt with



- Without tool-support, requirements traceability is almost impossible in large projects
- The usage of the traceability tool has to be integrated into the software development process

How should traceability be dealt with



- As much traceability as necessary
 - In order to gain benefits from traceability a certain extra effort is inevitable
 - Efforts should be concentrated on important and high-risk issues
- As less traceability as possible
 - Prevent over engineering on unimportant and low-risk issues

Evolution of the understanding of traceability over time



- Traceability used to be seen as a technique to improve a system description
- Today, traceability is an important tool in the software development process
 - Enables quality control

Questions?



... thanks for your attention!





References

- Dick, J. (2005). Design traceability. *IEEE Software* **22**, 6 (Nov./Dec. 2005). 14-16.
- Gotel, O., A. Finkelstein (1994). An Analysis of the Requirements Traceability Problem. *1st International Conference on Requirements Engineering*, Colorado Springs. 94-101.
- Jarke, M (1998). Requirements Traceability. *Communications of the ACM* **41**, 12 (Dec. 1998). 32-36.
- Ramesh, B., M. Jarke (2001). Toward Reference Models for Requirements Traceability. *IEEE Transactions on Software Engineering* **27**, 1 (Jan 2001). 58-92.