

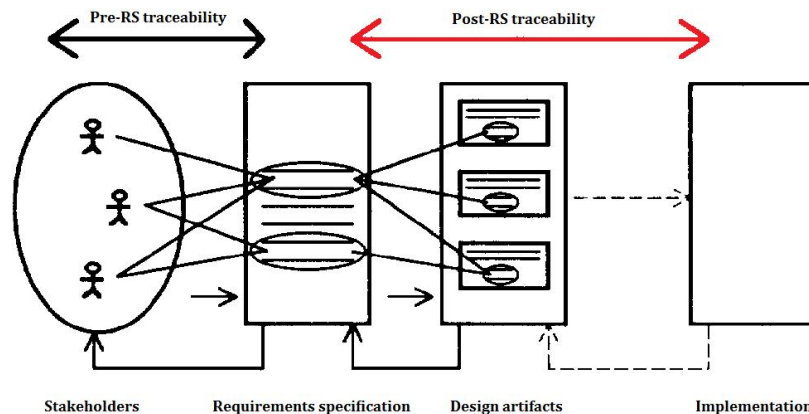
Impact Analysis with Traceability

David Oggier and Karin Bühler

Contents

- ▶ Post-requirements traceability
- ▶ Impact analysis with traceability
- ▶ Tool support for impact analysis with traceability

Post-requirements traceability



Post-RS traceability allows to:

- ▶ connect requirements to design artifacts and program code
- ▶ trace design and implementation decisions back to requirements
- ▶ test for sufficiency (are requirements satisfied?)
- ▶ test for necessity (does it need all the design artifacts)

Impact analysis

Problems of implementing changes:

- ▶ Difficulty to predict impact of a change
- ▶ Systems deteriorate over time because of frequent changes
- ▶ Code becomes harder and harder to modify

Impact analysis is the identification of:

- ▶ the effects of the implementation of a changing requirement
- ▶ the consequences on the system if the code is changed

Concepts of impact analysis

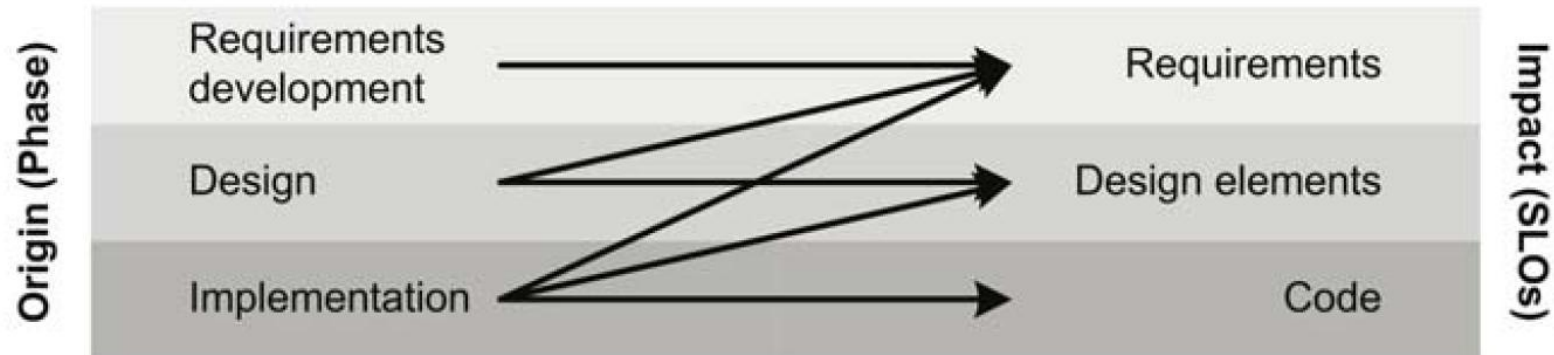
Software life-cycle objects:

- ▶ Central object of research in impact analysis
- ▶ Include requirements, design elements and source code
- ▶ Are connected and influence each other

Types of impact:

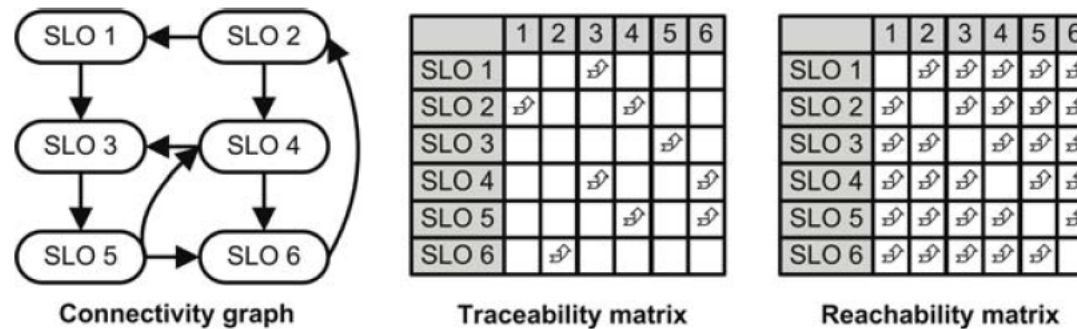
- ▶ Direct impact
- ▶ Indirect impact
- ▶ Side effects: unintended behavior
- ▶ Ripple effects: effects on other parts of the system

Benefits of impact analysis



- ▶ Allows for more flexibility in iterative development processes
- ▶ Changes are less costly and their impact evaluated much faster
- ▶ Very useful to determine cost associated with a change and whether it is beneficial to implement it

Recording change impact: traceability matrices



- connectivity graph and traceability matrix model direct impact
- reachability matrix models indirect impact
- traceability semantics?
- limiting indirect impact: threshold?

Measuring change impact

Decisive factors

- ▶ Number and kind of SLOs affected by the change
- ▶ Severity of the changes

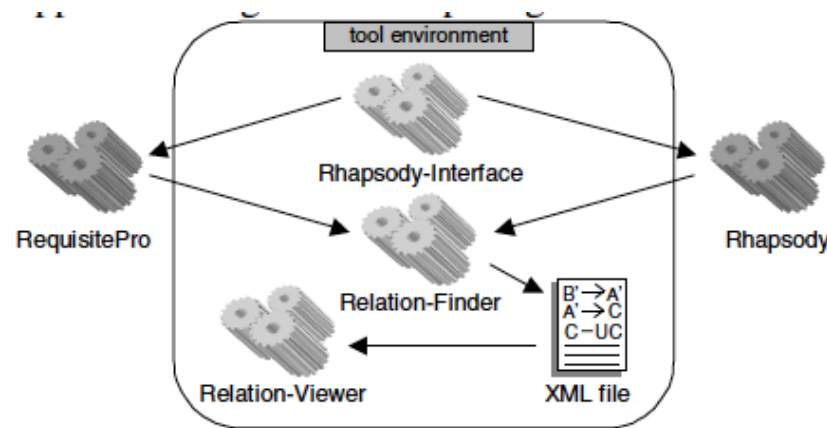
Success:

- ▶ Depends on completeness and consistency of traceability links
- ▶ On the level of requirements: often underprediction

Tool Support: Example QuaTrace

- ▶ Tool environment to establish, analyze, and maintain traces effectively and efficiently, for (semi)automatic impact analysis
- ▶ Supports the requirements engineer, project planner and maintainer
- ▶ Integrates and extends a requirements management tool (*RequisitePro*) and a CASE tool (*Rhapsody*)

Tool Support: Example QuaTrace ctd.



Main functionalities:

- ▶ Interface between *RequisitePro* and *Rhapsody*
- ▶ (Semi-)automated setting of traces
- ▶ Tailored (impact) analysis