Requirements Engineering I

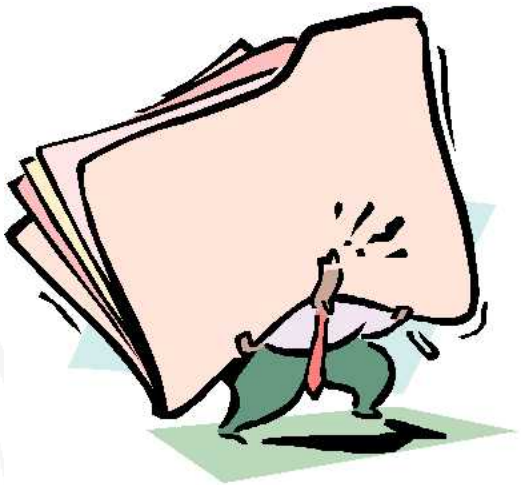
Chapter 10

Requirements Management

© 2021 Martin Glinz

Tasks of requirements management

- Organize
  - Store and retrieve
  - Record metadata (author, status,...)
- Prioritize
- Keep track: dependencies, traceability
- Manage change



© 2021 Martin Glinz

10.1 Organizing requirements

Every requirement needs

- a unique identifier as a reference in acceptance tests, review findings, change requests, traces to other artifacts, etc.
- some metadata, e.g.
  - Author
  - Date created
  - Date last modified
  - Source (stakeholder(s), document, minutes, observation...)
  - Status (created, ready, released, rejected, postponed...)
  - Necessity (critical, major, minor)

© 2021 Martin Glinz

Storing, retrieving and querying

Storage

- Paper and folders
- Files and electronic folders
- A requirements management tool

Retrieving support

- Keywords
- Cross referencing
- Search machine technology

Querying

- Selective views (all requirements matching the query)
- Condensed views (for example, statistics)

© 2017 Martin Glinz

10.2 Prioritizing requirements

- Requirements may be prioritized with respect to various criteria, for example
  - Necessity
  - Cost of implementation
  - Time to implement
  - Risk
  - Volatility

- Prioritization is done by the stakeholders
- Only a subset of all requirements may be prioritized
- Requirements to be prioritized should be on the same level of abstraction

© 2021 Martin Glinz

Simple prioritization (by necessity)

Ranks all requirements in three categories with respect to necessity, i.e., their importance for the success of the system

- Critical (also called essential, or mandatory)

  The system will not be accepted if such a requirement is not met

- Major (also called conditional, desirable, important, or optional)

  The system should meet these requirements, but not meeting them is no showstopper

- Minor (also called nice-to-have, or optional)

  Implementing these requirements is nice, but not needed

© 2017 Martin Glinz

Selected prioritization techniques

Single criterion prioritization

- Simple ranking

  Stakeholders rank a set of requirements according to a given criterion

- Assigning points

  Stakeholders receive a total of n points that they distribute among m requirements

- Prioritization by multiple stakeholders may be consolidated using weighted averages. The weight of a stakeholder depends on his/her importance

© 2017 Martin Glinz

Selected prioritization techniques – 2

Multiple criterion prioritization

- Wiegers' matrix [Wiegers 1999]
  - Estimates relative benefit, detriment, cost, and risk for each requirement
  - Uses these values to calculate a weighted priority
  - Ranks according to calculated priority values
- AHP (Analytic Hierarchy Process) [Saaty 1980]
  - An algorithmic multi-criterion decision making process
  - Applicable for prioritization by a group of stakeholders

© 2017 Martin Glinz

10.3 Traceability

[Gotel and Finkelstein 1994]

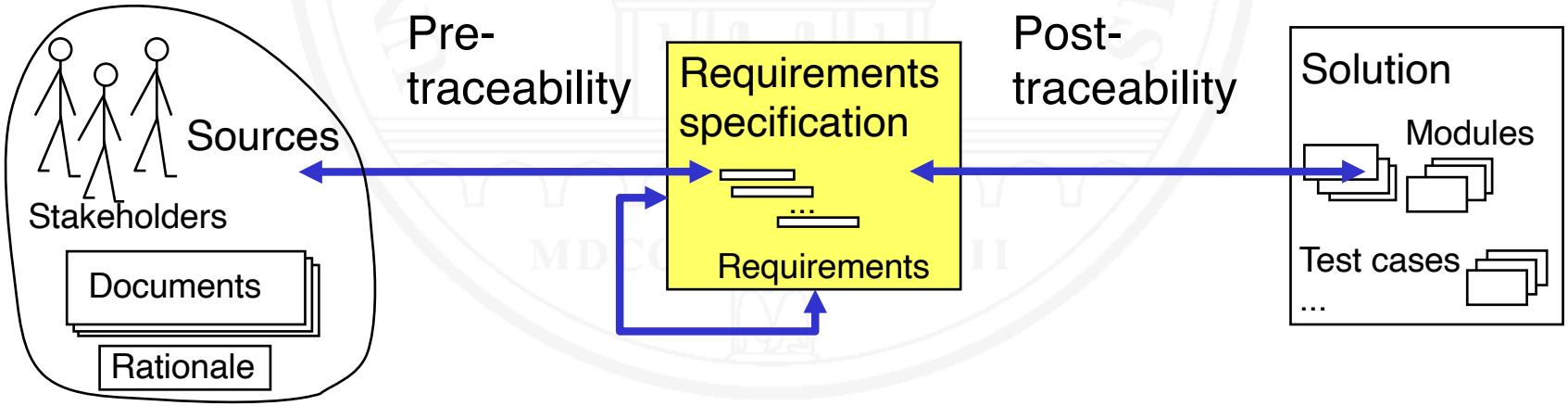
DEFINITION. Traceability – The ability to trace a requirement

(1) back to its origins,

(2) forward to its implementation in design and code,

(3) to requirements it depends on (and vice-versa).

Origins may be stakeholders, documents, rationale, etc.



© 2021 Martin Glinz

Establishing and maintaining traces

- Manually
  - Requirements engineers explicitly create traces when creating artifacts to be traced
  - Tool support required for maintaining and exploring traces
  - Every requirements change requires updating the traces
  - High manual effort; cost and benefit need to be balanced
- Automatic
  - Automatically create candidate trace links between two artifacts (for example, a requirements specification and a set of acceptance test cases)
  - Uses information retrieval technology
  - Requires manual post processing of candidate links

© 2017 Martin Glinz

10.4 Requirements evolution

The problem (see Principle 7 in Chapter 2):

Keeping requirements stable...

... while permitting requirements to change

Potential solutions

- Agile / iterative development with short development cycles (1-6 weeks)
- Explicit requirements change management

Every solution to this problem further needs requirements configuration management

© 2021 Martin Glinz

Requirements configuration management

Keeping track of changed requirements

- Versioning of requirements
- Ability to create requirements configurations, baselines and releases
- Tracing the reasons for a change, for example
  - Stakeholder demand
  - Bug reports / improvement suggestions
  - Market demand
  - Changed regulations

© 2017 Martin Glinz

Classic requirements change management

Adhering to a strict change process

(1) Submit change request

(2) Triage. Result: [OK I NO I Later (add to backlog)]

(3) If OK: Perform impact analysis

(4) Submit result and recommendation to Change Control Board

(5) Decision by Change Control Board

(6) If positive: make the change, create new baseline/release,
(maybe) adapt the contract between client and supplier

Change control board – A committee of customer and supplier representatives that decides on change requests.

© 2020 Martin Glinz

Requirements change in agile development

In agile and iterative development processes, a requirements change request ...

- ... never affects the current sprint / iteration, thus ensuring stability
- ... is added to the product backlog

Decisions about change requests are made when prioritizing and selecting the requirements for the subsequent sprints / iterations

© 2017 Martin Glinz