

Requirements Engineering: Intro - Why RE? What is it?

CECS 590

Requirements Engineering – Outline

- **WHY do we need Requirements Engineering and what is it?**
- Principles: Definitions, process, roles, problem/solution view, artifact orientation
- System Models: Decomposition and abstraction, system views
- Frameworks: What reference structures can I use for requirements?
- Business Case Analysis: Why are we building this system?
- Stakeholders: Who are the people to talk to about requirements?
- Goals and Constraints: What are the major objectives for the system?
- System Vision: What exactly do we want to achieve?
- Domain Models: What are the surrounding systems ours interacts with?
- Usage Models: How will the system interact with the user?
- Software quality models: How to determine the quality characteristics?
- Quality requirements: How to specify which qualities need to be met?
- Process requirements: How to specify constraints for development?
- Towards a system specification: How to hand over to design?
- Quality assurance: How to ensure that RE is done in a good way?
- Change management: How to evolve requirements?

Why do we need RE?

- Why Requirements Engineering?
- What are the problems?
- How is Requirements Engineering defined?
- What is Requirements Management?

What is Requirements Engineering?

- Def.: **Requirements Engineering** (RE) denotes the iterative, systematic, efficient and effective approach to elaborating an explicit requirements specification that is agreed upon by all stakeholders.
- It includes the major phases of
 1. Elicitation
 2. Analysis
 3. Specification
 4. Validation & Verification

WHY requirements engineering?

- Without defining the fundamental characteristics of a system under development, no system can be developed.
- Even if you think you don't do RE, you still handle requirements somehow.
- If you handle them in a systematic way, you understand the customer's problem and can then develop the best solution.

Motivation

These guys want a webpage:

“We dig holes”

www.youtube.com/watch?v=V8AoWTK0itQ

What requirements do they have?

Exercise – let's write requirements

Simple template for phrasing requirements:

[when?] [under what conditions?]

THE SYSTEM SHALL | SHOULD | WILL <process>
<thing to be processed> [<process details>*]

Discussion

- Why is RE important?
- What are challenges in requirements engineering?
- What are problems in RE?

Problems in RE*

* Acc. to a study by Mendez 2012

1. Incomplete and / or hidden requirements
2. Inconsistent requirements
3. Terminological problems
4. Unclear responsibilities
5. Communication flaws within project teams and with customers
6. Moving targets (changing goals, business processes and / or requirements)
7. Technically unfeasible requirements
8. Stakeholders with difficulties in separating requirements from previously known solution designs
9. Underspecified requirements that are too abstract and allow for various interpretations
10. Unclear / unmeasurable non-functional requirements

Challenge: successful RE is hard to measure

Definitions

- What is a Requirement?
- RE und RM
- RE in the process

What is a Requirement?

- Def.: A requirement is
 1. a constraint/ability/characteristic that a stakeholder requires for a product or process in order to solve a problem or reach a goal.
 2. a constraint/ability/characteristic that a system has to satisfy in order to fulfill a contract, a standard, a specification or other given formal documents.
 3. a documented representation of a constraint/ability/characteristic as defined in 1. or 2.

Source: IEEE-Standard, Std. 610.12-1990 [IEEE610.12.1990]

Characterization of Requirements

- Form / **Syntax**: representation in natural language, picture, formula, diagram, or model
- Content / **Semantics**:
which characteristic is required?
- **Quality**: Both have a strong influence on specific qualities of the requirement, e.g., precision, abstractness, understandability, unambiguousness, verifiability
- Terms strongly related to a requirement:
 - **Artefact**: documentation of one or more requirements
 - **Stakeholder**: individual with an interest in the system
 - **Source**: origin of a requirement, e.g., stakeholders, (legacy) systems in use, laws & standards, literature
 - **Rationale**: justification for demanding a specific requirement

Categorization of Requirements

- Classification of requirements (in principle negotiable)
 - Functional requirements
 - Quality requirements
 - Process requirements
- Further relevant information in RE (non-negotiable)
 - Domain knowledge
 - Terminology
 - Constraints
(e.g. interfaces, legal constraints, physical constraints)

Requirements Management and RE

Recap: **RE** is the approach to elaborating a requirements specification agreed by stakeholders.

Def.: **Requirements Management** (RM) efficiently and effectively manages the elaboration and usage of requirements in the whole software lifecycle.

RM includes activities from project management:

- Budget calculation, risk management, planning, ...
- Project control: Incremental development, time boxing, ...

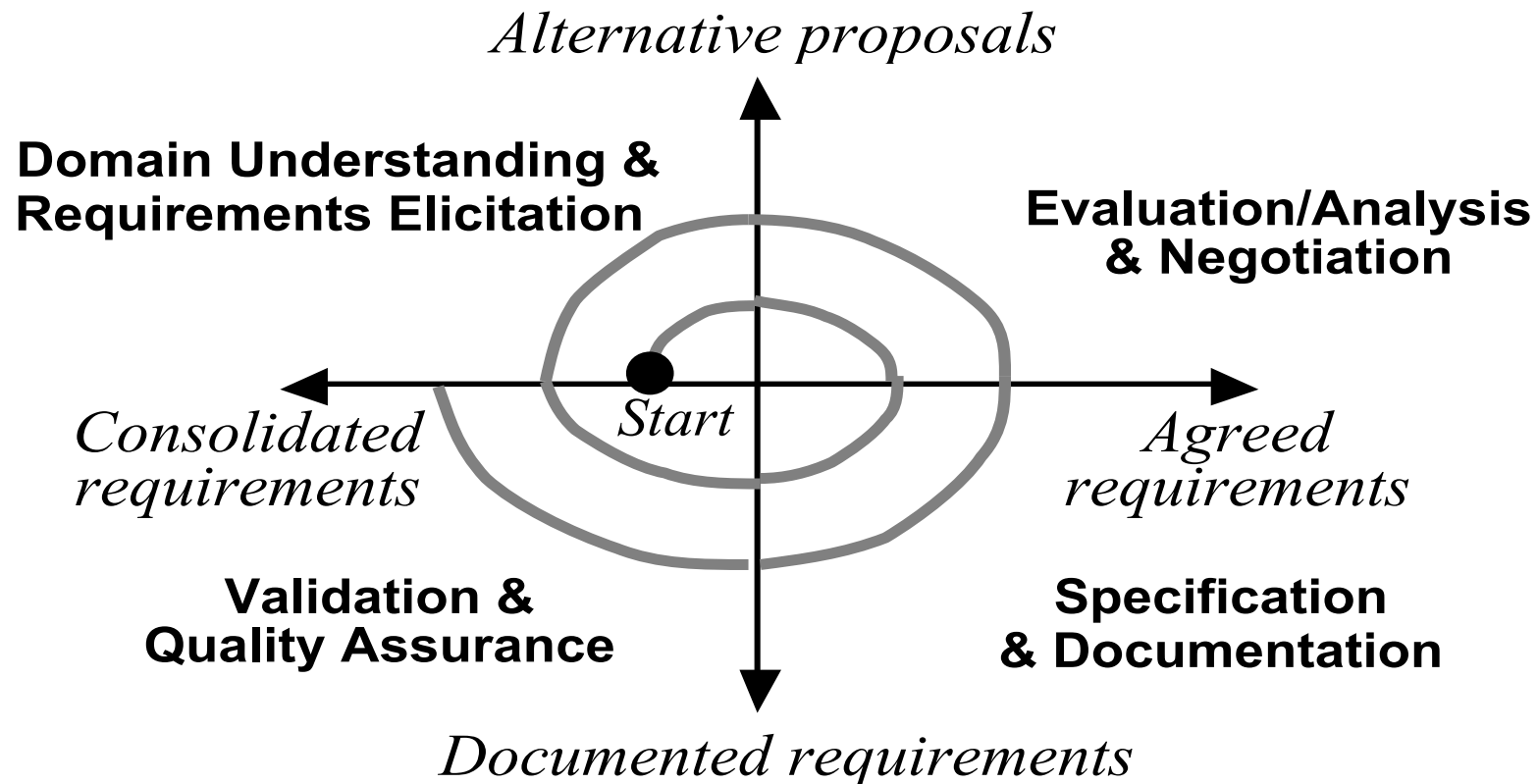
Tasks in Requirements Management

- Administration of requirements
- Structuring, documenting and archiving
- Attribution of requirements
- Rationale management and traceability
- Justification of requirements
- Linking in between requirements and to documentation artefacts
- Validation and verification
- Change management including impact analysis
- Version management and configuration management
- Claim management
- Support for distributed RE
- Tool support

RE in the process

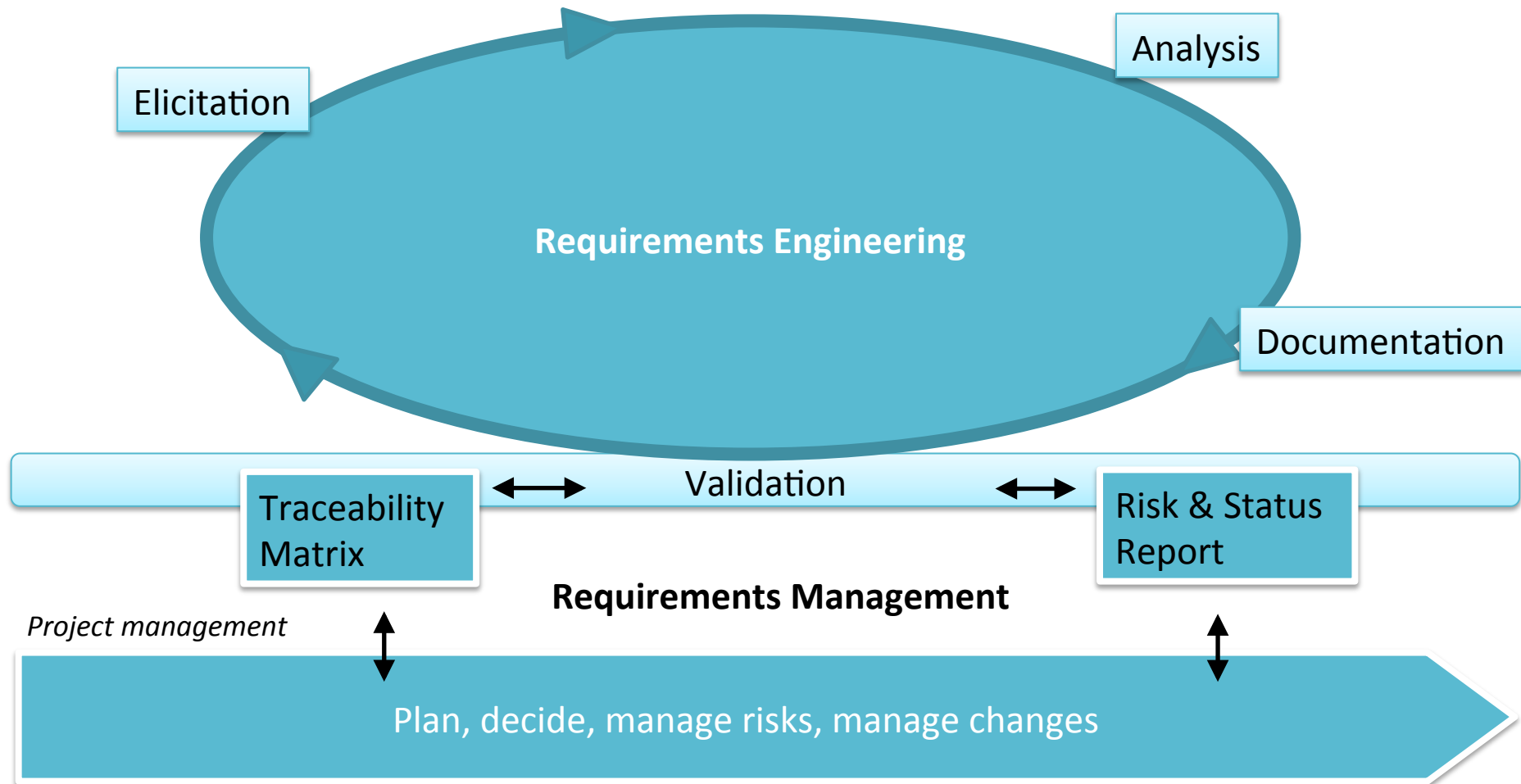
- Phases (not linear):
 - Elicitation
 - Analysis
 - Specification
 - Validation & Verification
- Development is **iterative**
- Phases are recurring
(req. and architecture co-development)

The RE process (simplified)



Source van Lamsweerde, 2009

The RE process



So far, so good...

- Take a deep breath.
- Questions so far?

- What is the one thing you heard that you deem most important?

Exercise: 1 minute paper

- Importance of RE:
Name five arguments that are especially convincing for you why systematic RE matters, for example, why RE is a success factor for software projects or why errors in RE can have serious consequences.