Requirements Engineering Framework

Requirement Engineering From System Goals to UML Models to Software Specifications, Alex Van Lamsweerde, 2009

Behzad Nazarbakhsh
Faiz Ali shah
Software Objects Representing Real Objects

• The software (model) contains objects that represent objects in the system environment.
• The software (model) should represent the environment correctly (Satisfy Requirement).

Figure 1.1: Phenomena and statements about the environment and the software-to-be [Lamsweerde]
What is Requirement Engineering Process

• The process of collecting, elaborating, correcting or adapting statements that may differ in mood and in scope called requirement engineering process (Jackson, Parnas & Mady 1995).

• A system requirement is a **Perspective Statement** to be enforced by the software-to-be, possibly in corporation with other system components and formulated in terms of environmental phenomena (e.g. Train doors shall always remain closed when the train is moving).

• The requirement engineering process involves **Statements (Descriptive + Perspective)** about the system-to-be that differ in scope.

• Process, Actors, and Products are three elements which must be differentiated during requirement engineering process.
Activities of Requirement Engineering Process

- Domain understanding
- Requirement elicitation
- Evaluation and agreement
- Specification and documentation
- Requirement consolidation
Domain Understanding

• **Objective:** Studying the system -as-is within its organizational and technical context.

• **What to do?**
  ▪ **Understanding organization structure:** strategic objective, business policies, roles plays by organizational units, actors, and dependency among them.
  ▪ **Understanding scope of system:** objectives, required components, the concept which relies on, tasks that must be involved in, information that following it, and constraints.
  ▪ **Understanding stakeholders**
  ▪ **Strength and weakness of the system-as-is.**
Requirement Elicitation

• **Objective:** Discovering candidate requirement (Stakeholders problems) and shape the system-to-be.

• **What to do?**
  - Exploring new technologies (keep strength of current system and address the weakness)
  - Improvement the system-to-be.
  - Finding organizational and technical constraints.
  - Finding boundaries (what should be automated and what not).
  - Illustrating the interaction between software and environment.
  - The domain properties and assumption about the environment.
  - Requirement that software must meet.
Evaluation and agreement

• **Objective:** To make informed decisions about issues raised during the elicitation process.

• **Negotiation:**
  ▪ Conflicting concerns
  ▪ Assessing risks
  ▪ Requirement prioritization (high and low priority, easy requirement, and new constraints)

• Outcome of this activity is in **Draft Proposal & Risk Documentation:**
  ▪ Final decision.
  ▪ Agreed requirements
  ▪ Assumptions
Specification and documentation

• **Objective:** Detailing, structuring and documenting the agreed characteristics of the system-to-be.

• **Outcome of this activity is Requirement Document (RD):** objectives, concept definition, relevant domain properties, responsibilities, system requirements, software requirement, and environmental assumption

Requirement Consolidation

• **Objective:** Analyzing carefully the specification resulting from the preceding activities (Quality assurance).

• **What to do?**
  - Validating with stakeholders and **Verifying** to find inconsistencies.

• Outcome of this activity: Consolidated requirements document (e.g. detected errors, consequence of not fixing errors, etc.), Prototype or mock-up (requirement validation), additional test data coming out of verification, development plan.
Requirement Engineering Process Artefacts

• Draft proposal
• Software requirement specification.
• Risk documentation for risk analysis
• Interaction matrix to find overlapped requirements and conflicts.
• Consolidated requirements document.