

Software Requirements

CSC 4700 Software Engineering

Lecture 3

Based on Sommerville, Chapter 6

Requirements engineering (RE)

- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.
- Requirements are:
 - descr. of system services and constraints
 - generated during the RE process

Why?

- Ensure everyone is on the same page
 - PMs and Developers
 - Developers and Developers
- Identify incomplete thoughts
- Reevaluate based on feedback
 - Developers are not always present to get feedback first hand

What is a requirement?

- High-level abstract statement of a service or of a system constraint
- Detailed mathematical functional specification
- Used for:
 - Contract bid
 - Contract itself

Two Approaches

Sequential (old way)

- Non-negotiable
- Defined up front
- Highly-detailed
- Limited interaction with product manager
- Changes handled formally
- Don't know everything up front

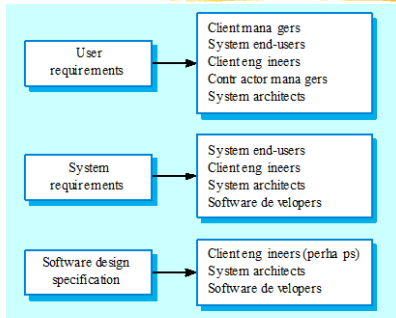
Agile (new way)

- Continuously negotiated, Re-prioritized, re-scoped
- Manipulated to meet business goals
- Flexibility in face of uncertainty
- Backlog instead of detailed document

Types of requirement

- User requirements
 - Diagrams & descriptions for customer to understand
- System requirements
 - Details of system's functions, services and operational constraints
 - What should be implemented

Requirements readers



Functional requirements

- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.
- Functional **user requirements** may be high-level statements of what the system should do but functional **system requirements** should describe the system services in detail.

Examples of functional reqs.

- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.
- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall be able to copy to the account's permanent storage area.

Requirements imprecision

- Problems arise when requirements are not precisely stated.
- Ambiguous requirements may be interpreted in different ways by developers and users.
- What causes imprecision?
 - Miscommunication
 - English (natural lang.) is imprecise
 - Incompleteness

Requirements completeness and consistency

- In principle, requirements should be both complete and consistent.
- Complete
 - They should include descriptions of all facilities required.
- Consistent
 - There should be no conflicts or contradictions in the descriptions of the system facilities.
- In practice, it is impossible to produce a complete and consistent requirements document.

Non-functional reqs.

- These define system **properties** and **constraints** (e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.)
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless.

Non-functional classifications

- Product requirements
 - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organizational requirements
 - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- External requirements
 - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

Examples of Non-functional reqs.

- Product requirement
 - 8.1 The user interface for LIBSYS shall be implemented as simple HTML without frames or Java applets.
- Organisational requirement
 - 9.3.2 The system development process and deliverable documents shall conform to the process and deliverables defined in XYZCo-SP-STAN-95.
- External requirement
 - 7.6.5 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

Goals and requirements

- Non-functional requirements may be very difficult to state precisely and imprecise requirements may be difficult to verify.
- Goal
 - A general intention of the user such as ease of use.
- Verifiable non-functional requirement
 - A statement using some measure that can be objectively tested.
- Goals are helpful to developers as they convey the intentions of the system users.

Domain requirements

- Derived from the application domain and describe system characteristics and features that reflect the domain.
- Domain requirements be new functional requirements, constraints on existing requirements or define specific computations.
- If domain requirements are not satisfied, the system may be unworkable.

Domain requirements problems

- Understandability
 - Requirements are expressed in the language of the application domain;
 - This is often not understood by software engineers developing the system.
- Implicitness
 - Domain specialists understand the area so well that they do not think of making the domain requirements explicit.

User requirements

- Should describe functional and non-functional requirements in such a way that they are understandable by **system users** who don't have detailed technical knowledge.
- User requirements are defined using natural language, tables and diagrams as these can be understood by all users.

Problems with natural language

- Lack of clarity
 - Precision is difficult without making the document difficult to read.
- Requirements confusion
 - Functional and non-functional requirements tend to be mixed-up.
- Requirements amalgamation
 - Several different requirements may be expressed together.

Guidelines for writing requirements

- Invent a standard format and use it for all requirements.
- Use language in a consistent way. Use shall for mandatory requirements, should for desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon.

Problems with NL specification

- Ambiguity
 - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult.
- Over-flexibility
 - The same thing may be said in a number of different ways in the specification.
- Lack of modularization
 - NL structures are inadequate to structure system requirements.

The requirements document

- The requirements document is the official statement of what is required of the system developers.
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it
