Software Engineering

Requirements Engineering

Dr. Sayed AbdelGaber
Professor
Faculty of Computers and Information
Helwan University

REQUIREMENTS ENGINEERING

- The process of establishing the services that the customer requires from a system.

- The constraints under which the system operates and is developed.

Requirements Engineering

- "… to cover all of the activities involved in **discovering, documenting,** and **maintaining** a set of requirements for a computer-based system."

[Source: Kotonya and Sommerville, 2001]

- Importance of requirements illustrated by need for rigor, structure, repeatable techniques, etc.

- Key concepts:
  - Discovery (capture, elicitation, identification)
  - Documenting (specifying, modeling)
  - Maintenance (changes in environment of system)

Requirements analysis

WHAT ARE REQUIREMENTS?

- They are descriptions of how the system should
  - ✓ *behave,*
  - ✓ *application domain* information,
  - ✓ *constraints* on the system's operation, or
  - ✓ *specifications* of a system property or attribute.

[Source: Kotonya and Sommerville, 2001]

- A requirement is a *statement of need*, something that some class of user or other *stakeholder wants*"

[Source: Alexander and Stevens, 2002]

WHAT ARE REQUIREMENTS?

Requirements definition is a careful assessment of the needs that a system is to fulfill.

It Says

- **Why a system is needed?**
- **What system features will serve and satisfy?**
- **How the system is to be constructed?**

Requirements represent a specification for the new system.

Why is Requirements Engineering Important?

- Identifies stakeholder needs to ensure that you are building the right system
- Help you manage the development process to ensure a quality system
- Identify defects early, reducing costs

Why are Requirements Important?

Consequences of poor or incorrect requirements are:

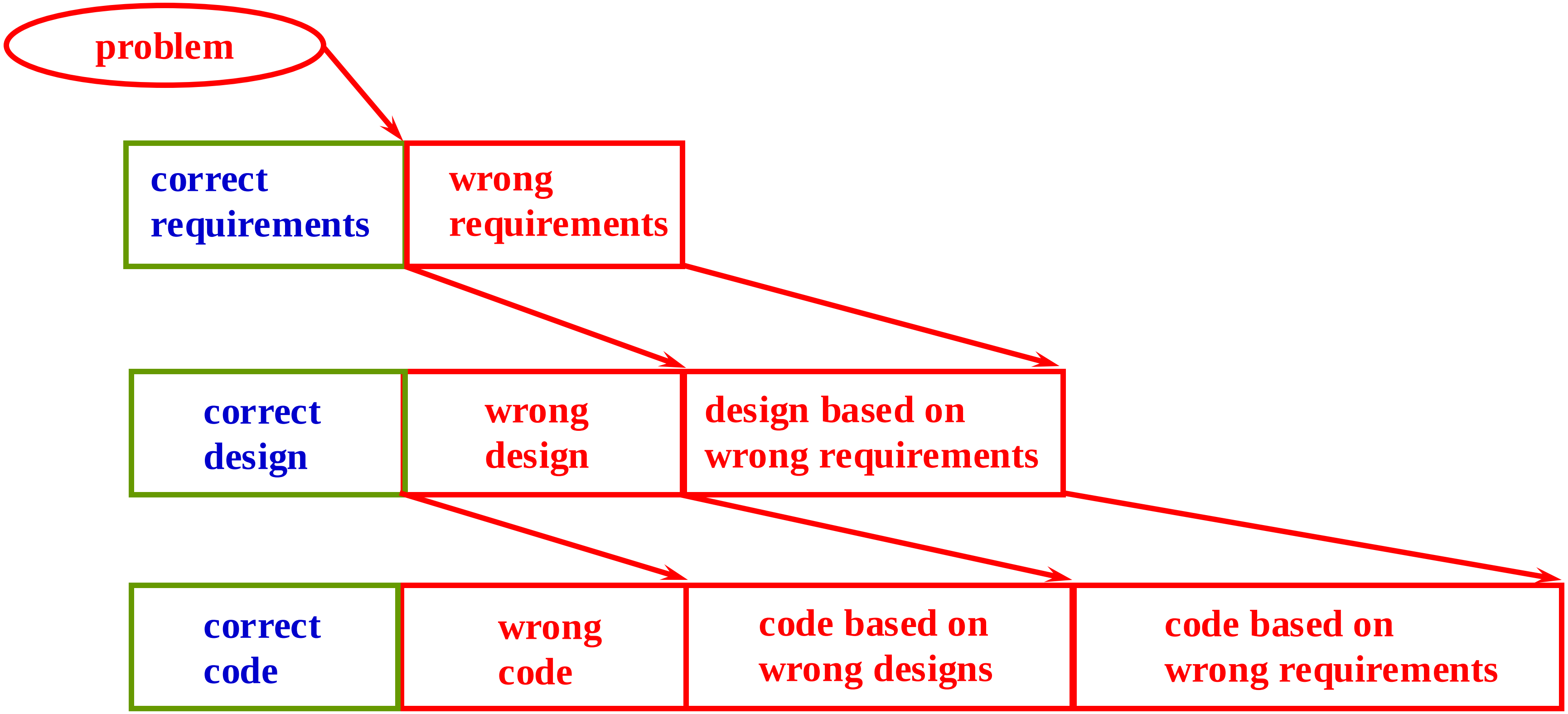
- **System delivered later than planned.**
- **System cost more than planned.**
- **Customers and end-users unsatisfied (not used or even scrapped)**
- **Maintenance related costs higher.**
- **System is unreliable or does not work.**

important!

Reasons for Poor Requirements

- Not well understood (domain complexity).
- Requirements do not reflect the real needs of the customer.
- Misunderstanding between various stakeholders: customers, requirements engineers and developers.
- Expensive to make changes to requirements after they have been agreed upon (contract).
- Requirements evolution: existing system enhancement not clear.
- Requirements could be incomplete, ambiguous, inconsistent, overlapping, not implementable, ...

DEFECTS PROPAGATE AND GROW



problem
correct requirements
wrong requirements
correct design
wrong design
design based on wrong requirements
correct code
wrong code
code based on wrong designs
code based on wrong requirements

Effects of Poor Requirements Analysis

- "Once your software hits the field, removing a requirements defect costs at least a hundred times as much, assuming you can fix it at all.“

[Source: Lawrence, Wiegers, and Ebert, IEEE Software, 2001]

- Poor or incorrect requirements affects later stages of SDLC through exponentially increasing costs:
  - investment far greater during design and coding.

- Investing time in effective requirements analysis early saves time, effort, and money.

Types of Requirements

Usually classified into two categories:

Functional (Behavioral) requirements

Specify

- **The function that the system should provide.**
- **How the system should react to particular inputs.**
- **How the system should behave in particular situations.**

(affordance, capability) of the system

Non Functional (Non-Behavioral) requirements

Describe

- **Constraints on the services or functions offered by the system such as timing constraints.**
- **Constraints on the development process, standards, etc.**

(performance, reliability, usability, portability, ...)

Types of Requirements

- **General requirements**

  sets out in broad terms what the system should do

  For Example:

  The system shall maintain records of all library materials including books, serials, newspapers and magazines, ...).

- **Functional requirements**

  define system's functionality

  For Example:

  The system shall allow users to search for an item by title, author, or ISBN

Types of Requirements

- **Data requirements**

  define the type of data the system shall operate upon or produce

  For Example:

  The ISBN is a 5-part item: the "ISBN tag" and a 4-part identifier.

- **Implementation requirements**

  states how the system must be implemented

  For Example:

  The system's user interface shall be implemented using a WWW browser.

Types of Requirements

Non-Functional Requirement

- **Performance requirements**

  specify the minimum acceptable performance of the system

  For Example:

  The system shall support at least 20 transactions per second

- **Usability requirements**

  state user interface and system availability constraints

  For Example:

  [illegible] cal menu structure for navigation

- **Operational requirements**

  specify constraints that should be satisfied during system usage

  For Example:

  Reliability in terms of "mean-time to failure".

Properties of Good Requirements

- **Understandable by users**

  effectively contract and should be comprehendible.

- **Nonprescriptive**

  describes what the system should do not how

- **Correct**

  user judge of correctness

- **Complete**

  nothing missing from set or individual requirement

- **Consistent**

  does not contradict other requirements

- **Unambiguous**

  only one interpretation

- **Precise, concise, feasible, testable, traceable, ...**

Contents of Requirements Document

- The services and functions which the system should provide.
- The constraints under which the system must operate.
- Overall (emergent) properties of the system (reliability, maintainability, performance, usability, security, etc.).
- System's environment including related systems.
- Application domain information.
- Constraints on the development processes.

Requirements Specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Indication of other entities required.
- Pre and post conditions (if appropriate).
- The side effects (if any) of the function.

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.

IEEE/ANSI 830-1993
Requirements Specification Standard

1. Introduction
   - 1.1Purpose of document
   - 1.2 Scope of product
   - 1.3 Definitions, acronyms, abbreviations
   - 1.4 References
   - 1.5 Overview of remainder of document

2. General description
   - 2.1 Product perspective
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 General constraints
   - 2.5 Assumptions and dependencies

IEEE/ANSI 830-1993
Requirements Specification Standard

3. Specific requirements

Includes *functional*, non-functional, interface requirements (external interfaces, functionality, performance, *logical database requirements*, ...)

4. Appendices

5. Index

Lecture 3 Summary

- Requirements related problems contribute significantly to project failure.
- Requirements engineering incorporates activities such as discovery, analysis, documenting, and management.
- Many variations of requirement types.
- Different kinds of stakeholders have say in requirements.
- Requirements can be specified as documents, models, prototyping and formal methods.

Questions for the Week

1. Why is it essential to capture good quality requirements?

2. Describe ways in which requirements can be substandard? How do we improve the chances of capturing good quality requirements?

3. Identify the stakeholders that can claim an interest in requirements capture and for each, explain why.

4. What characteristics are useful for analysts to possess during requirements determination? Which is the most important and why?

Questions