Requirements Analysis

MD. ISRAFIL MAHMUD RAJU
LECTURER
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DAFFODIL INTERNATIONAL UNIVERSITY

Requirements Analysis

- Software engineering task bridging the gap between system requirements engineering and software design.
- Provides software designer with a model of:
  - system information
  - function
  - behavior
- Model can be translated to data, architectural, and component-level designs.
- Expect to do a little bit of design during analysis and a little bit of analysis during design.

Analysis Objectives

- Identify customer's needs.
- Evaluate system for feasibility.
- Perform economic and technical analysis.
- Allocate functions to system elements.
- Establish schedule and constraints.
- Create system definitions.

Software Requirements Analysis Phases

- Problem recognition
- Evaluation and synthesis
  - focus is on what not how
- Modeling
- Specification
- Review

Management Questions

- How much effort put towards analysis?
- Who does the analysis?
- Why is it so difficult?
- Bottom line - who pays for it?

Feasibility Study

- Economic feasibility
  - cost/benefit analysis
- Technical feasibility
  - hardware/software/people, etc.
- Legal feasibility
- Alternatives
  - there is always more than one way to do it

Requirements

- Requirement
  - features of system or system function used to fulfill system purpose.
- Focus on customer's needs and problem, not on solutions:
  - Requirements definition document
    - (written for customer).
  - Requirements specification document
    - (written for programmer; technical staff).

Types of Requirements - 1

- Functional requirements:
  - input/output
  - processing.
  - error handling.
- Non-functional requirements:
  - Physical environment (equipment locations, multiple sites, etc.).
  - Interfaces (data medium etc.).
  - User & human factors (who are the users, their skill level etc.).

Types of Requirements - 2

- Non-functional requirements (continued):
  - Performance (how well is system functioning).
  - Documentation.
  - Data (qualitative stuff).
  - Resources (finding, physical space).
  - Security (backup, firewall).
  - Quality assurance (max. down time, MTBF, etc.).

Requirement Validation

- Correct?
- Consistent?
- Complete?
  - Externally - all desired properties are present.
  - Internally - no undefined references.
- Each requirement describes something actually needed by the customer.
- Requirements are verifiable (testable)?
- Requirements are traceable.

Requirements Definition Document

- General purpose of document.
- System background and objectives.
- Description of approach.
- Detailed characteristics of proposed system (data & functionality).
- Description of operating environment.

Software Requirements Elicitation

- Customer meetings are the most commonly used technique.
- Use context free questions to find out
  - customer's goals and benefits
  - identify stakeholders
  - gain understanding of problem
  - determine customer reactions to proposed solutions
  - assess meeting effectiveness
- Interview cross section of users when many users are anticipated.

F.A.S.T

- Facilitated application specification technique
- Meeting between customers and developers at a neutral site (no home advantage).
- Goals
  - identify the problem
  - propose elements of solution
  - negotiate different approaches
  - specify preliminary set of requirements

Use Cases

- Scenarios that describe how the product will be used in specific situations.
- Written narratives that describe the role of an actor (user or device) as it interacts with the system.
- Use-cases designed from the actor's point of view.
- Not all actors can be identified during the first iteration of requirements elicitation, but it is important to identify the primary actors before developing the use- cases.

User Profile - Example

- Full Control (Administrator)
- Read/Write/Modify All (Manager)
- Read/Write/Modify Own (Inspector)
- Read Only (General Public)

Use Case Example - 1

- **Read Only Users**
  - The read-only users will only read the database and cannot insert, delete or modify any records.
- **Read/Write/Modify Own Users**
  - This level of users will be able to insert new inspection details, facility information and generate letters. They will be also able to modify the entries they made in the past.

Use Case Example - 2

- **Read/Write/Modify All Users**
  - This level of users will be able to do all the record maintenance tasks. They will be able to modify any records created by any users.
- **Full Control Users**
  - This is the system administrative level which will be able to change any application settings, as well as maintaining user profiles.

Analysis Principles

- Information domain of problem must be presented & understood.
- Models depicting system information, functions, and behavior should be developed.
- Models and problems must be partitioned in a manner that uncovers detail in layers.
- Analysis proceeds from essential information toward implementation detail
- Must be traceable.

Information Domain

- Encompasses all data objects that contain numbers, text, images, audio, or video.
- Information content or data model
  - shows the relationships among the data and control objects that make up the system
- Information flow
  - represents manner in which data and control objects change as each moves through system
- Information structure
  - representations of the internal organizations of various data and control items

Thank You