Requirements Specification

CSE 403 - Spring 1997

Outline

- Requirements vs. specification vs. design.
- Approaches to requirements specifications.
- Case Study: A-7 requirements spec. [Heninger]

Requirement Analysis

- Understand, clarify, and precisely state the actual requirements for the desired system.
- These requirements may differ significantly from the original concept.

Requirements Specification

Three Types of Requirements

- Functional requirements
  - "A user must be able to retrieve customer records that contain matching partial information entered."
- Performance requirements
  - "95% of all the information retrieval must complete within two seconds."
- Other constraints
  - "The server software must remain independent of the database server chosen."
  - "The client software must be written in Java."

Requirements, Specification & Design

- Requirements---a statement of criteria that a system must satisfy.
- (Requirements) Specification---a representation of those requirements in human and possibly machine-readable form.
- Design---a "how to" statements describing an implementation of a system satisfying requirements.
- What vs. How
  - The distinctions are somewhat arbitrary and fuzzy.
  - Utilize models?

What's in Requirements Specification?

- *Not all aspects are applicable to all systems.*
- Description of software functions.
  - Desired outputs.
  - Required inputs.
  - The association between the inputs and the outputs.
- Description of what the software should *not* do.
- Description of the environment in which the software is to operate.
  - Human operators (role and qualifications), organizational structure, physical characteristics of environment, maintenance organization for hardware and software, economic incentives for installation of the system, ...

What's in Requirements Specification?

- Description of interface between the system and its environment.
  - Man-machine interface (including dialogs, etc.), planning operating procedures, hardware interface to electronic equipments, etc.
- Descriptions of data.
  - Bounds on input and output data.
  - Plausible ranges of data at specific points of computation.
  - Required precisions.
- Description of time and throughput requirements.
- Correlation of software requirements to system requirements.

What's in Requirements Specification?

- Possible error sources and suitable measures for the prevention of such errors and/or the minimization of their effect.
- Explicit quantitative acceptance criterion for each individual requirement.
  - a measure to determine success.
- Description of life-cycle considerations.
  - Subsets, fundamental assumptions, potential changes, etc.
- Description of constrains on implementation.

Desirable Properties

- Requirements specification should have the following properties:
  - Clear and understandable
  - Unambiguous
  - Consistent
  - Complete
- What can we do to make the requirements specification to maintain these properties throughout its life-cycle?

General Approaches

- Informal specifications
  - Free-form, natural languages.
  - Ambiguity and lack of organization can lead to problems with incompleteness, inconsistency and misunderstandings.
- Systematic specifications
  - Standardized syntax.
  - Basic consistency and completeness checks.
  - Imprecise semantics implies that other sources of errors may still be present.
- Formal specifications
  - Rigorous syntax and semantics; precise (often mathematical) form; provides basis for mathematical verification of consistency and completeness of specification and of equivalence between specification and program.

Data Flow Specifications

- Data flow diagrams describe systems as "collections of data [arrows] that are manipulated by functions [bubbles]."[GJM]
- Often used to capture flow of data and operations for information (data processing) systems.
- Semantics are imprecisely defined.
  - Semantics of the symbols (i.e., the operators) are not generally defined.
  - Control aspects can be interpreted in different (reasonable) ways:
    - All inputs vs. any inputs, do outputs get identical data, etc.
    - No support for concurrency and synchronization.
  - Control issues make data flow diagrams inappropriate for embedded and real-time systems.

Case Study: A-7 [Heninger]

- Part of a project to apply "modern" (circa 1979) software engineering techniques to a real system---the operational flight program for the A-7 aircraft.
  - Build a new modern system and compare to existing system---must be functionally identical.
  - Therefore, need a requirements specification.
- There existed no complete (or even reasonably complete) description of the requirements specification.
  - This was first part of the project, and it led to this approach to requirements specification.
  - And to a 500-page document (condensed from several shelves of documentation.)

Document Design

- Key questions
  - What kinds of questions should it answer?
  - Who are the readers?
  - How will it be used?
  - What background knowledge does a reader need?
- Principles
  - State questions before trying to answer them.
  - Separate concerns.
  - Be as formal as possible.

Document Objectives

- Specify external behavior only.
- Specify constraints on the implementation.
  - Can't change physical interfaces.
- Be easy to change.
- Serve as a reference tool.
  - Answer specific questions quickly, not general explanation.
- Record forethought about the life-cycle of the system.
- Characterize acceptable responses to undesired events.
  - Because a pilot knows more about how to handle sensor failures than a programmer does.

Example Table of Contents [Lamb]

- Introduction
- Environment characteristics
  - Hardware, peripherals, people, ...
- Fixed interfaces
  - Devices, operating system, databases, ...
- Input and output
- Modes
- Software functions
- Constraints and goals
- Quantitative requirements
  - Timing, precision, etc.
- Response to undesired events
- Life-cycle considerations
  - Subsets, fundamental assumptions, potential changes.
- Glossary
- Sources

Not every requirements specification will have all of this information.

How About Ours?

- Clear, unambiguous, consistent, complete.
- Primarily, if not exclusively, in natural language.
  - Intelligently structure your requirements specification.
  - *Maybe*, data flow diagrams.
- It needs to satisfy two roles.
  - To the customer: specifies what will be built.
  - To the developer: specifies what to build.