CS446
Software Engineering

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

September 29, 1996

Requirements and Specifications

Richard Bartels

University of Waterloo
Faculty of Mathematics
Department of Computer Science

September 29, 1996

Outline

- Establishing Requirements
- Levels of Precision
- Requirements Validation
- Requirements Evolution
- Requirements vs. Specifications
- Constraints and Non-functional Requirements
- Modes of Communication and Expression

Reference:

This is material from Chapters 4 and 7 of Sommerville

Observations

It is unlikely that a software project will be developed on time and within budget, or prove to be satisfactory, if the developers do not have a clear idea of what is to be developed.

The most important thing to get clear in establishing the requirements of a software product is what the client *needs* rather than what the client *wants*; the two are not always the same.

The person to whom you are speaking may be the client, but he/she is often not the person who will actually use the software to be developed. What is needed and what is wanted may have to be interpreted in the light of this fact.

Whatever requirements are established must be clearly understood by the client; they must also be clearly understood by the developer at the managerial level, and they must also be clearly understood by the developer at the technical level. All three understandings must be mutually consistant.

In larger projects it may be necessary to have someone or some department solely responsible for maintaining consistency with requirements throughout. This is an important aspect of *Software Quality Assurance*

Definitions

Requirements definition

A statement intelligible to client and developer that sets out what services a system is to provide and under what constraints it will perform. It is usually framed in plain language and in diagrams, and it is the result of developer/client discussions. It states *what* services are to be provided without restricting *how* they will provided.

Requirements specification

A more precise and specific restatement of the requirements definition that serves as the contract between the purchasing client and the developer. It is framed in precise language and diagrams, should be thought of as legally binding, and is the result of developer/client negotiations. It will include statements about how customer acceptance is to be obtained and how delivery is to take place.

Software specification

A restatement of the requirements specification that is precise enough to be used for the basis of design and implimentation. It may be framed in technical terms; e.g., pseudo code or formal predicates, and using technical graphical methods; e.g. flowcharts or data-flow diagrams.

Informal Example: A Turnstyle

Requirements definition

Provide a device that prevents entry until money is paid.

Requirements specification

Provide a coin-operated entry device that is normally locked, unlocks when coins of a certain value are deposited, allows one person to enter, and then locks again. The device should not accept coins while unlocked.

Software specification

- The system will involve the following abstractions
  - *coin box*, *wheel gate*
  - *locked state*, *unlocked state*
  - *accepting state*, *not accepting state*
  - *turn action*, *coin deposit*
- The *coin box* is initially in the *accepting state*.
- The *wheel gate* is initially in the *locked state*.
- When the *coin box* is in the *accepting state* and receives a *coin deposit*, the *coin box* enters the *not accepting state*, and the *wheel gate* enters the *unlocked state*.
- When the *wheel gate* is in the *unlocked state* and receives a *turn action*, the *wheel gate* enters the *locked state*, and the *coin box* enters the *accepting state*.
- The *wheel gate* will not permit a *turn action* when it is in the *locked state*.
- The *coin box* will not permit a *coin deposit* when it is in the *not accepting state*.

Readers

Requirements definition

- Client administrative managers
- Development administrative managers
- End users

Requirements specification

- Client administrative managers
- Development administrative managers
- End users
- Client technical managers
- Development technical managers
- Contract managers

Software specification

- Development technical managers
- System architects
- Software developers

Requirements are Difficult

In large, new systems it is difficult, if not impossible, to define a complete and consistent set of requirements on the first try.

- The user community may be diverse and have diverse needs
- The project may never have been tried before and all eventualities may not be foreseen or the problem may be poorly understood
- Constraints on development time and expense may be imposed that are in conflict with necessary requirements

As a result, it may be necessary to revise requirements during the development phase.

- It is important that this be done in a controlled and organized fashion and only as necessary
- Revisions should be negotiated and agreed upon between client and developer, not made arbitrarily by either party
- When requirements are revised, it should be expected that the project's pricing and delivery schedule has to be renegotiated

Requirements: Volatility and Life Cycle

Feasibility study

Estimate what the requirements might be and whether they can be met technically and in a cost-effictive manner.

Requirements analysis

Gather information from the client and the end-users and, by an iterative process of proposal, feedback, and clarification, arrive at prospective requirements.

Rapid prototyping

A useful tool during requirements analysis consisting of a mock-up of the potential system; usually this involves creating a system with the proposed human interface and some canned input/output but having no underlying functionality.

Requirements definiton

The full, initial set of requirements and constraints, presented in document form to all parties for agreement in principle.

Requirements specification

The final, expanded, carefully worded set of requirements and conditions to be met for satisfactory delivery that accompanies the formal contract.

Reviews

Periodic reviews are made of all requirements in the light of development experience, and of continuing consultation with and intermediate deliveries to the client. These reviews should be part of established milestones, and any changes to requirements proposed by either party should be raised and negotiated.

Requirements Validation

Requirements reviews are conducted by

- walking through the requirements manually
- also, sometimes, automatic consistency checking (if requirements are sufficiently formalized)

Each requirement should be checked for

- Validity
  - is the requirement really what is needed?
- Consistency
  - does the requirement conflict with others?
- Completeness
  - are all functions and constraints included?
- Realism
  - can each requirement really be met?
- Verifiability
  - is each requirement as stated realistically testable?
- Comprehensibility
  - is each requirement properly understood?
- Traceability
  - is the origin and reason for each requirement recorded?

Above All: Adaptability

- Is each requirement reasonably independent of the others?
- Can it be changed without requiring extensive changes to other requirements?
- Are requirements documents written with enough flexibility and organization for easy change?

Controlled, Consistent Evolution

- The requirements documents should be updated consistently whenever the system implementation changes
- The requirements should always reflect the system implementation
- The system implementation should always reflect the requirements
- The requirements documents should be modular and easy to update
- The requirements history should be kept in a configuration management system
- The system implementation history should be kept in a configuration management system
- The two histories should be correctly and consistently cross-referenced