Requirements in the Real World

Brian F. Hanks

February 28, 2002
Introduction

- What are requirements?
- Why are they important?
- Who cares about them?
- What do they look like?
- How do you determine them?
Who am I?

- 20+ years industry experience as software engineer and manager
- Director of Software Engineering
- B.S. 1980, M.S. 1987, Ph.D. 2004?
What is a Requirements Specification?

- Description of system behavior, environment, and characteristics.
- Can think of it as a contract between the customer and the developers.
Why are Requirements Important?

• What do you think?
Why are Requirements Important?

Requirements problems, such as:

- Lack of user input
- Incomplete requirements
- Changing requirements

are a significant cause of project problems:

- Late delivery
- Cost overruns
- Reduced functionality
Why are Requirements Important?

- Tell what system must do and conditions in which it must do it
- Provide a baseline for changes
- Support project scheduling, staffing, and estimation
- Assist with conflict resolution
- Provide basis for design and test plan
- Allow you to prioritize work
- Let you know when you are done
Who are Requirements for?

- What do you think?
Who are Requirements for?

- Customer
- Software developers
- Software test staff
- Technical writers
- Maintenance staff
What form do they take?

• Formal notations such as UML, DFDs, ER diagrams.
• Informal natural language
• UI prototypes
• Combination
Features of Good Requirements

• What features or attributes should requirements have?
Features of Good Requirements

• Correct - Do they describe the system the customer wants?
• Clear and Unambiguous
• Concise
• Complete - Is anything missing?
• Consistent - Do requirements agree with each other?
• Verifiable - Is there a way to test that a solution satisfies the requirement?
• Feasible - Can it be done with available resources and technology?
Examples

• The application must be user friendly.
• The system must respond quickly to user input.
• The product must calculate the lowest fare.
Examples

• “Members of the library can normally borrow up to six items at a time, but members of the staff may borrow up to 12 items at one time.”

• “There should be a ‘Quit’ button.”

• “gn - go to the beginning of line n, where the first line is line number 1. If n is negative, the cursor is not moved. If n is greater than the number of lines in the buffer then the cursor is placed after the last character in the buffer.”
What Needs to be in a Requirements Document?

- What do you think?
What Needs to be in a Requirements Document?

- Description of features and functionality
  - Inputs and Outputs
    * type (integer, character, string, ...)
    * range of acceptable values
  - Processing
  - Exception Handling
What Needs to be in a Requirements Document?

• Environment where application will run
• Type and number of users
• Data model
What Needs to be in a Requirements Document?

- System Attributes
  - These are sometimes called ‘non-functional requirements’
  - Performance
  - Security
  - Reliability
  - Portability
  - Scalability
How do you Determine Requirements?

• Interviews
• Negotiation
• Observation of current system or method
• Customer participation
• Templates and Guidelines
• UI prototypes
Concluding Remarks

- Requirements are hard to do well, but...
- They are critical to project success.
For More Information

- Practical Software Requirements - Benjamin Kovitz
- Software Requirements - Karl Wiegers
- Mastering the Requirements Process - James and Suzanne Roberts
- IEEE Std 830 - IEEE Recommended Practice for Software Requirements Specifications