

# Examining the Software Specification

[Reading assignment: Chapter 4, pp. 54-62]

# Testing the specification

- You do not need to have code to start testing.
- Testing the specification can save on time and cost later on.
- Also, mistakes in specifications account for about 55% of all bugs.
- The specification is typically a written document using prose and pictures to describe the functional and non-functional aspects of the software.

# Requirements Specification: An Overview

- **Basic goal:** To understand the problem as perceived by the user.
- Activities of specification are **problem oriented**.
  - Focus on *what*, not *how* (this is design)
  - Don't cloud the specification with unnecessary detail.
  - Don't pre-constrain design in the specification.
- After specification is done, do **software design**:
  - solution oriented
  - *how* to implement the *what*

# Requirements Specification: An Overview

- Key to specification is good communication between customer and developers.
- Work from specification document as guide.

# Requirements Specification

- Basically, it's the process of determining and establishing the **precise** expectations of the customer about the proposed software system.

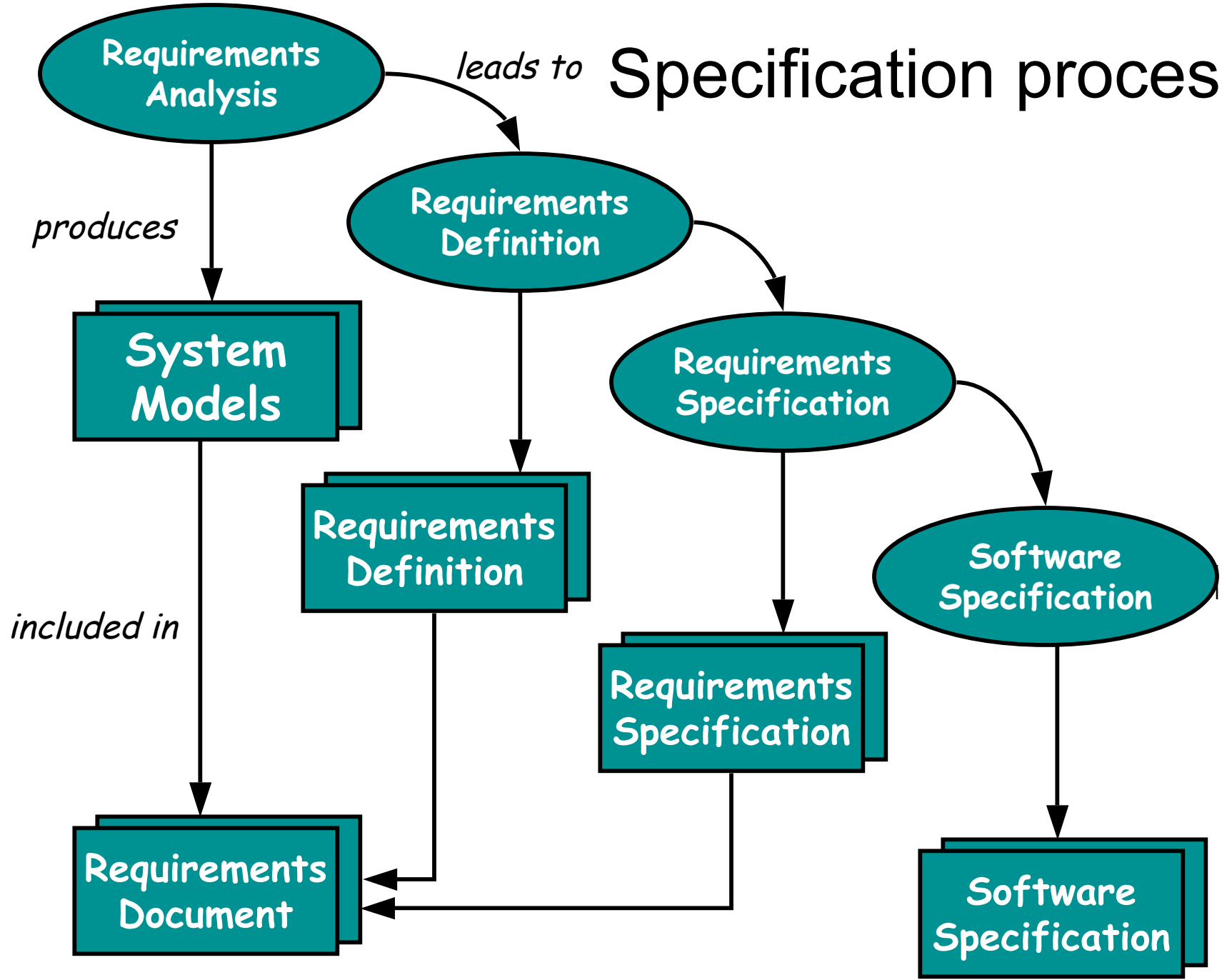
# Two kinds of requirements

- **Functional**: The precise tasks or functions the system is to perform.
  - *e.g.*, details of a flight reservation system
- **Non-functional**: Usually, a constraint of some kind on the system or its construction
  - *e.g.*, expected performance and memory requirements, process model used, implementation language and platform, compatibility with other tools, deadlines, ...

# The purpose of specification

- Raw user requirements are often:
  - vague
  - contradictory
  - impractical or impossible to implement
  - overly concrete
  - just plain wrong
- The purpose of specification is to get a usable set of requirements from which the system may be designed and implemented, with minimal “surprises”.

# Specification process





# The Specification document

- The official statement of what is required of the system developers.
  - Includes system models, requirements definition, and requirements specification.
  - Not a design document.
  - States functional and non-functional requirements.
- Serves as a reference document for maintenance.

# Specification document “requirements”

- Should be easy to change as requirements evolve.
- Must be kept up-to-date as system changes.

# Specification should state ...

- Foreseen problems:
  - “won’t support Win-3.x apps”
- Expected evolution:
  - “will port to MacOS in next version”
- Response to unexpected events/usage:
  - “if input data in old format, will auto-convert”

# Specification structure

- Introduction (describe need for system)
- Functional Requirements
- Non-Functional Requirements
- System Evolution (describe anticipated changes)
- Glossary (technical and/or new jargon)
- Appendices
- Index

# To summarize ...

- Specification focuses on determining *what* the customer wants, and not *how* it will be implemented.
- Specification is hard to get correct; it requires good communication skills.
- Requirements may *change* over time.
- Requirements specification requires iteration.
- The customer often doesn't have good grasp of what he wants.
- Bugs created in the requirements stage are very expensive to fix later.

# Specification reviews

- Involve people examining the specification with the aim of discovering anomalies and defects.
  - Reviewers reuse domain knowledge so they are likely to have seen the types of error that commonly arise.
- Does not require the execution of a system so may be used before implementation.
- Effective technique for discovering errors.

# Reviews and testing

- Reviews and testing are complementary and not opposing verification techniques.
- Both should be used during the V & V process.
- Reviews can check conformance with a specification but not conformance with the customer's real requirements.
- Reviews cannot check non-functional characteristics such as performance, usability, etc.

# Review pre-conditions

- A precise specification must be available.
- Team members must be familiar with the organization standards.
- Management must accept that reviews will increase costs early in the software process.
- Management must not use reviews for staff appraisal.



# What is a specification review?

- A process of identifying faults in the specification of a software system.
- Review should uncover both errors made in producing specification documents, and errors made earlier in the requirements engineering process.

# Limitations of conventional review approaches

- Too much information to go through, and not enough time to do it thoroughly.
- Unfamiliarity of individual reviewers with the overall goals of the design.
- No single part of the specification gets a thorough and complete evaluation.
- Burden is on reviewer to initiate action.
- One-on-one interaction between individual reviewers and specification team is limited.

## Better method:

### Active specification review process

- Change from “general” review to a set of more focused reviews.
- Use questionnaires to engage the reviewer in using the specification.
- More opportunities for one-on-one discussion between reviewer and specification team.

# An example

- We have been asked to review the specification for a hospital's order processing system.
- The order processing system allows users to order items for patients, such as tests or medications.

# Active specification review process

## Step 1: Prepare the specification for review

- Think about what criteria reviewers will use:
  - Well-structured
  - Simple
  - Adequate
  - Flexible
  - Practical
  - Easy to implement
  - Standardized

# Active specification review process

## Step 2: Prepare the documentation for review

- **Make assumptions explicit**
  - System can record the order pertaining to a patient.
  - It is possible to obtain all the orders for a patient.
  - System can determine and change the status of an order.
  - The order always contains at least one item.
  - The status of an order is always in one of the two states i.e active or cancelled.
- **Incorrect Usage Assumptions**
  - Cannot add or remove items once the order is placed.
  - Once an order is cancelled, the status cannot be set to active again.
  - An item is always added with respect to an order.

# Active specification review process

## Step 3: Identify the specialized reviews

- Focus the reviewer's attention on specific properties of the specification (e.g., data access).
  - Data access sufficiency.
    - E.g., provides all data required by the other features of the system.
  - Assumption Sufficiency.
    - E.g., contains all of the assumptions needed to access the feature's data.

# Active specification review process

## Step 4: Identify the reviewers needed

- People with different perspectives and expertise are needed as reviewers
  - Programmers and analysts who worked on the other features of the order processing system.
  - Programmers and analysts familiar with hospital information systems in general.



# Active specification review process

## Step 5: Design the questionnaires

- Make reviewers take an active role
- Make reviewers use the documentation
- Phrase questions in an active way
  - E.g., “Write down the exceptions that can occur” rather than “Are exceptions defined for every program?”

# Active Specification Review Process

## Step 6: Conduct the review

- Present an overview of the specification.
- Assign reviews to the reviewers.
- Reviewers complete their reviews, meeting with the specification authors as needed.
- Specification authors review completed questionnaires, and meet with reviewers to resolve questions.
- Specification authors produce new version of the specification.

# Specification attribute checklist

- Completeness
- Accuracy
- Precision
- Consistency
- Relevance
- Feasibility
- Code/Design-free
- Testability

# Specification terminology checklist

- Always, every, all, none, never, ... (absolutely sure?)
- Certainly, therefore, clearly, obviously, customarily, most, ... (persuasion lingo)
- Some, sometimes, often, usually, ordinarily, customarily, most, ... (vague)
- Etc., and so forth, and so on, such as, ... (not testable)
- Good, fast, cheap, efficient, small, stable, ... (unquantifiable)
- Handled, processed, rejected, skipped, eliminated, ...
- If ... then ... (missing else)

# Conclusions

- Reviewers focus on those areas they are best suited to evaluate
  - Time is used more wisely for all participants
  - More errors are likely to be found
- One-on-one communication with specification authors makes it easier for people to speak up.
- Few errors found does not necessarily indicate that the specification is good.
  - E.g., Perhaps the review process was not effective.

# You now know ...

- ... what a specification is
- ... how to review (test) a specification
- ... the benefits of an “active” specification review process