CS3213: Foundations of Software Engineering

Requirements Validation

Misunderstood or Ambiguous Requirements

Image by Gemini

Wrong Intuition

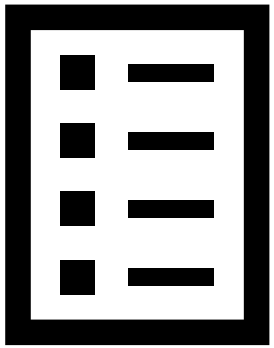
Image by Gemini

Requirements Validation

Checking that requirements define the system that the customer really wants

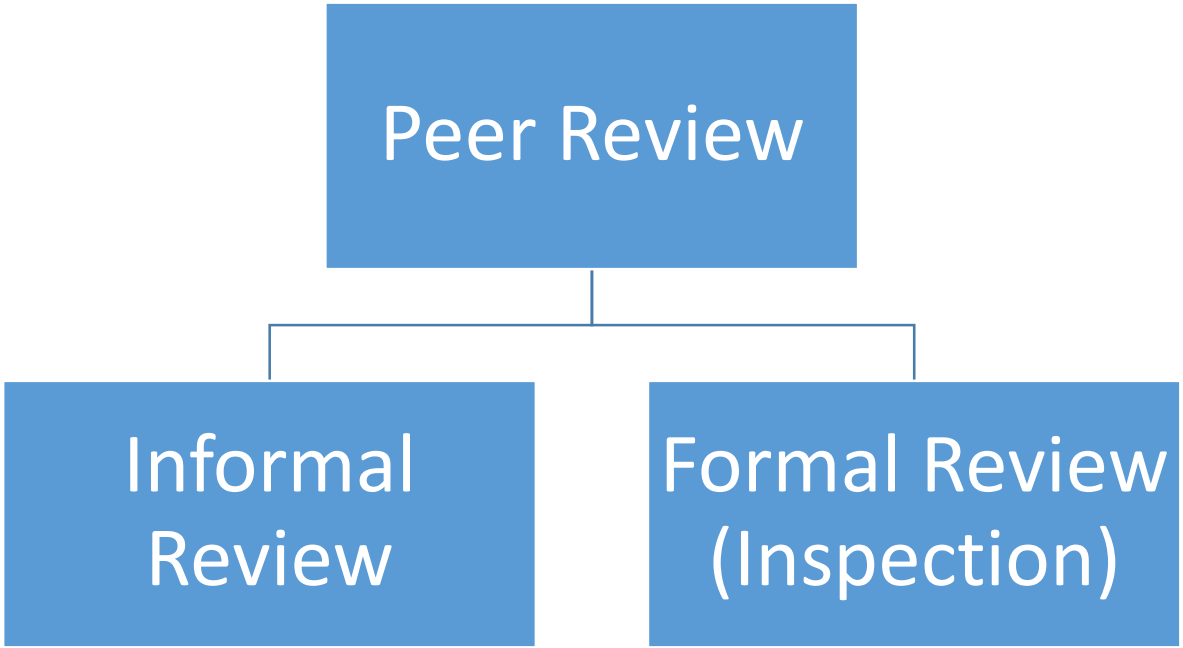
Requirements Validation

Agile

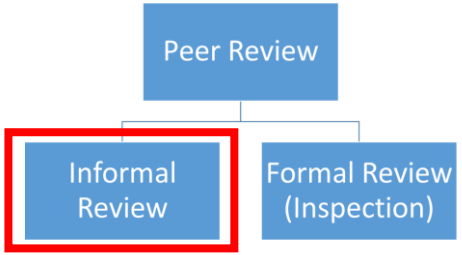


Plan-driven

Peer Reviews

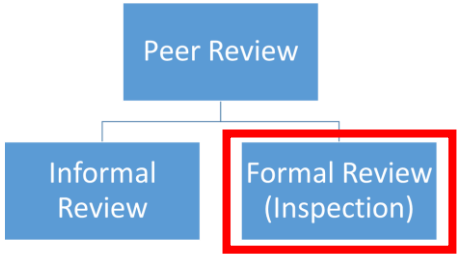


Informal Reviews



- **Informal reviews:** collecting unstructured feedback
  - A *peer deskcheck*, in which you ask one colleague to look over your work product.
  - A *passaround*, in which you invite several colleagues to examine a deliverable concurrently.
  - A *walkthrough*, during which the author describes a deliverable and solicits comments on it

Formal Reviews/Inspections



- **Formal requirements reviews:** the requirements are analyzed systematically by a team of reviewers who check for errors and inconsistencies.
- **Inspection:** Tedious and time-consuming, but one of the highest-leverage software quality techniques available

Incomplete, Conflicting, and Ambiguous

★☆☆☆☆

Anonymous

Reviewed in Singapoore in January 2025

This requirements document is a poor example of requirements engineering. First, there are numerous incomplete requirements, leaving much of the functionality unspecified. Second, the requirements that are specified conflict with each other, causing confusion and uncertainty. Third, the requirements are expressed in an unclear and ambiguous manner, making them open to interpretation.

Was this review helpful?

Helpful Report

Image by ChatGPT (refused to fix the Singapore typo)

Image by Gemini

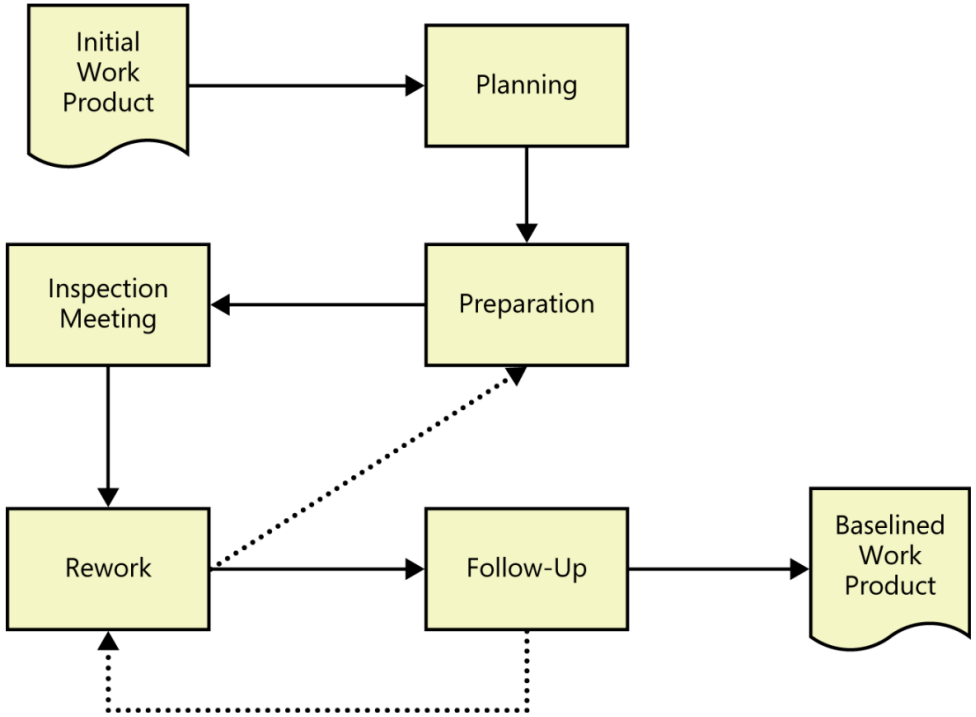
Inspections: Participants

- The author of the work product
- People who are the sources of information
- People who will do work
- People who are responsible for interfacing systems

Inspection: Roles

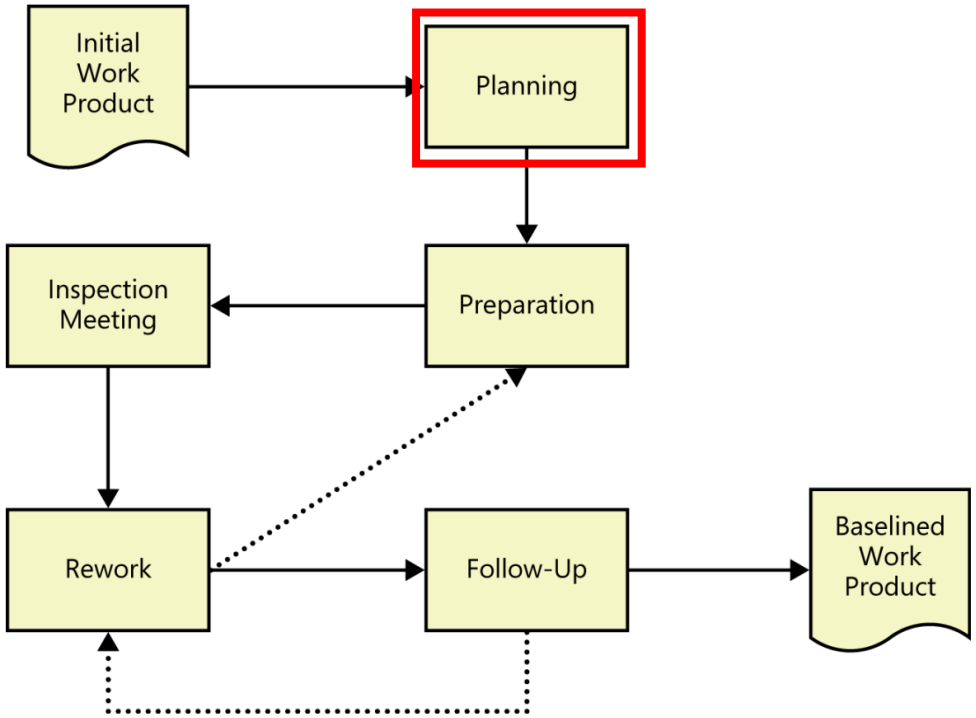
- Author
- Moderator
- Reader
- Recorder

Inspections: Process



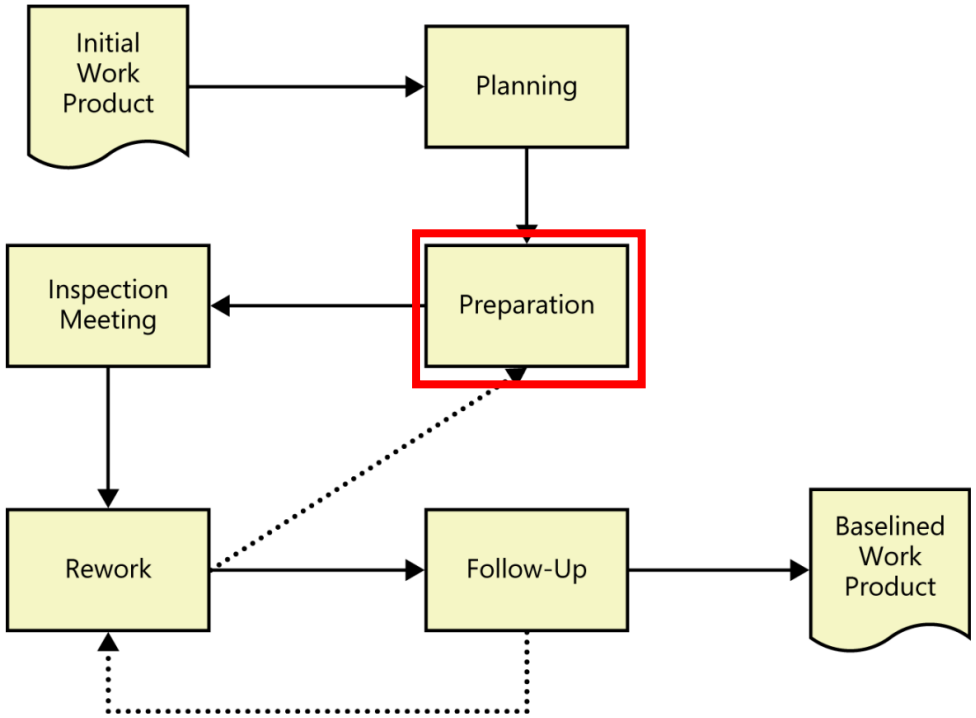
Software Requirements. Third Edition. Wiegers and Beatty.

Inspections: Process



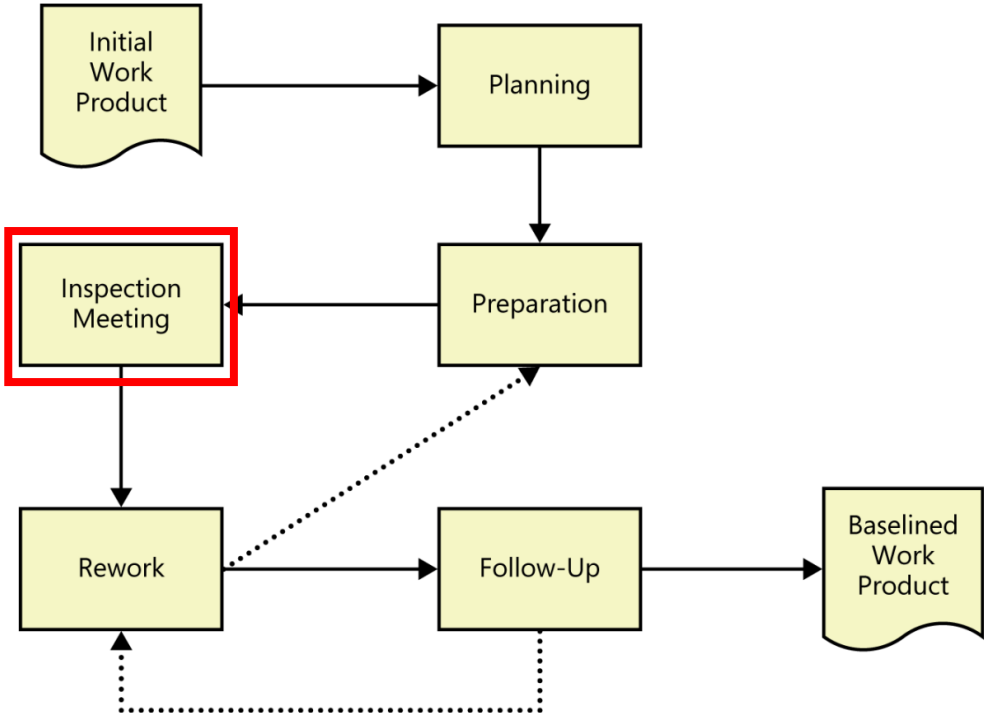
Software Requirements. Third Edition. Wiegers and Beatty.

Inspections: Process



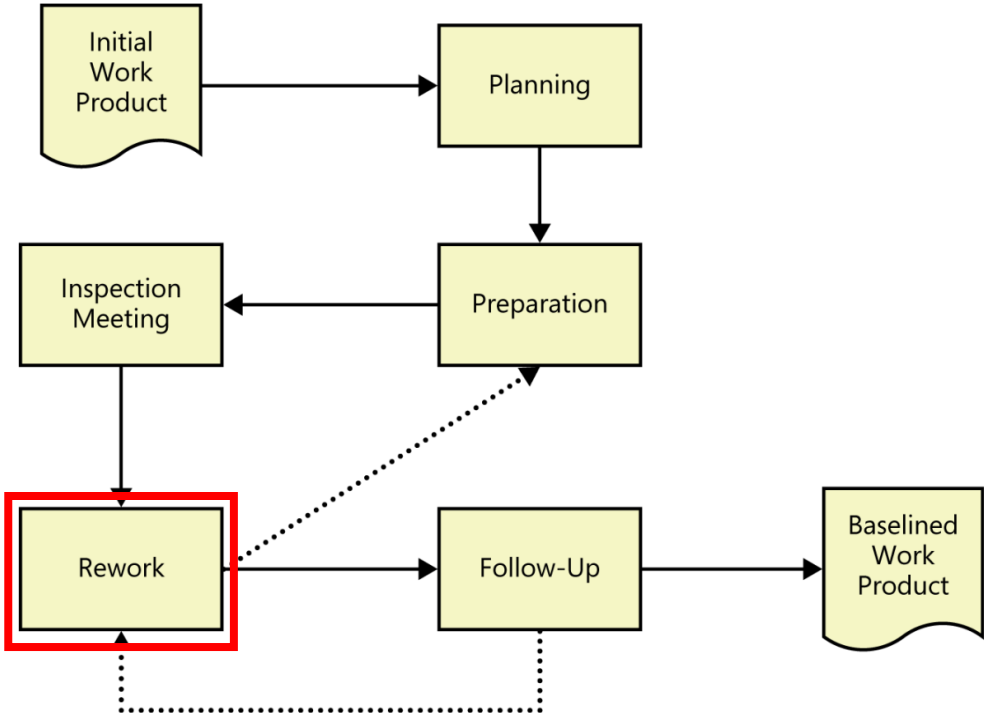
Software Requirements. Third Edition. Wiegers and Beatty.

Inspections: Process



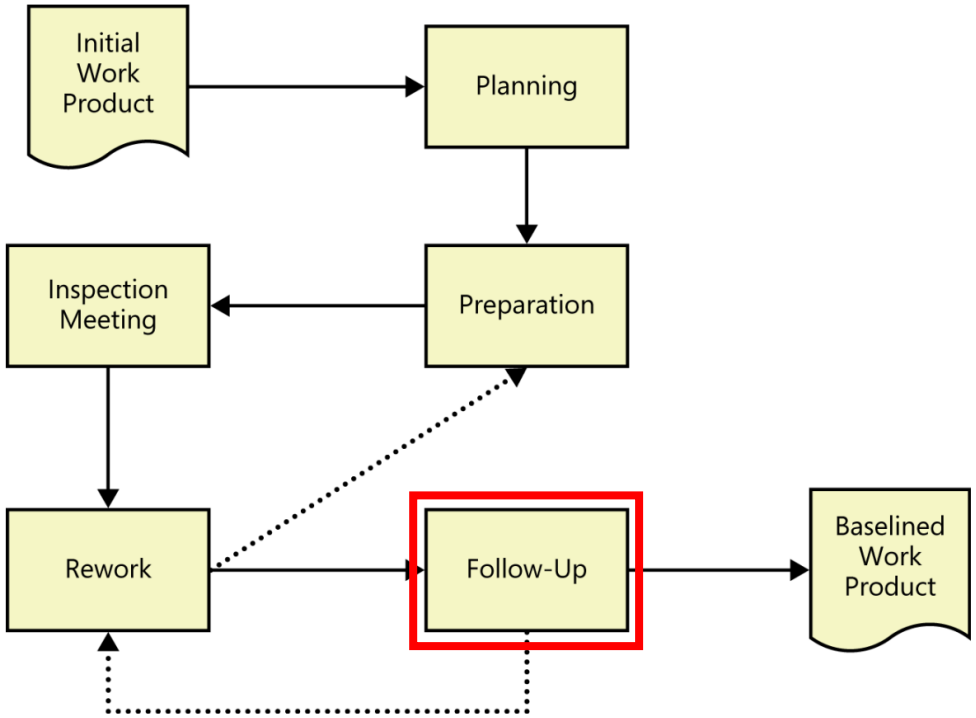
Software Requirements. Third Edition. Wiegers and Beatty.

Inspections: Process



Software Requirements. Third Edition. Wiegers and Beatty.

Inspections: Process



Software Requirements. Third Edition. Wiegers and Beatty.

Checks on the Requirements Document

- Validity checks
- Consistency checks
- Completeness checks
- Realism checks
- Verifiability

Completeness

- ❑ Do the requirements address all known customer or system needs?
- ❑ Is any needed information missing? If so, is it identified as TBD?
- ❑ Have algorithms intrinsic to the functional requirements been defined?
- ❑ Are all external hardware, software, and communication interfaces defined?
- ❑ Is the expected behavior documented for all anticipated error conditions?
- ❑ Do the requirements provide an adequate basis for design and test?
- ❑ Is the implementation priority of each requirement included?
- ❑ Is each requirement in scope for the project, release, or iteration?

Correctness

- ❑ Do any requirements conflict with or duplicate other requirements?
- ❑ Is each requirement written in clear, concise, unambiguous, grammatically correct language?
- ❑ Is each requirement verifiable by testing, demonstration, review, or analysis?
- ❑ Are any specified error messages clear and meaningful?
- ❑ Are all requirements actually requirements, not solutions or constraints?
- ❑ Are the requirements technically feasible and implementable within known constraints?

Quality Attributes

- ❑ Are all usability, performance, security, and safety objectives properly specified?
- ❑ Are other quality attributes documented and quantified, with the acceptable trade-offs specified?
- ❑ Are the time-critical functions identified and timing criteria specified for them?
- ❑ Have internationalization and localization issues been adequately addressed?
- ❑ Are all of the quality requirements measurable?

Organization and Traceability

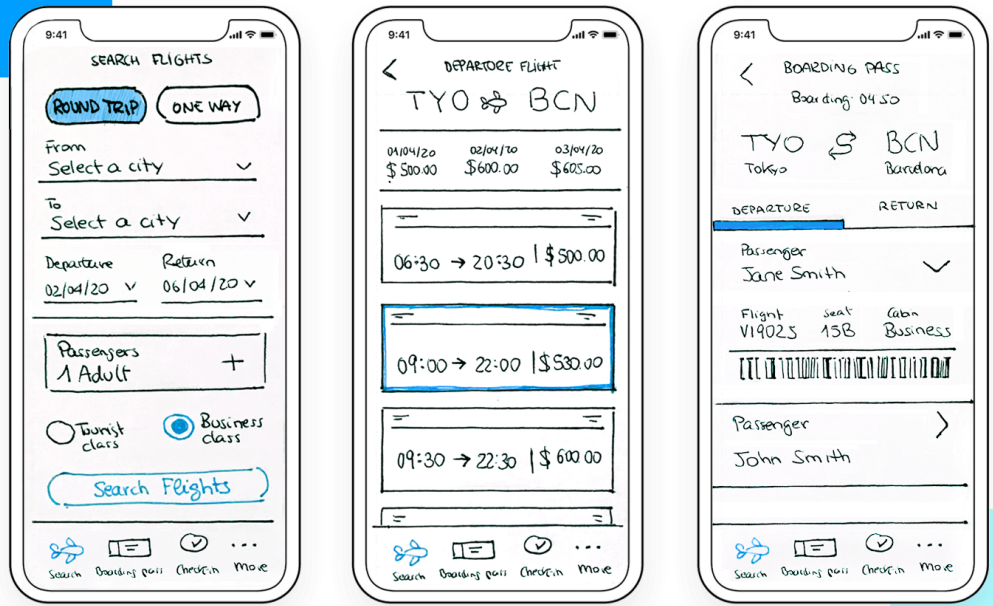
- ❑ Are the requirements organized in a logical and accessible way?
- ❑ Are all cross-references to other requirements and documents correct?
- ❑ Are all requirements written at a consistent and appropriate level of detail?
- ❑ Is each requirement uniquely and correctly labeled?
- ❑ Is each functional requirement traced back to its origin (e.g., system requirement, business rule)?

Other Issues

- ❑ Are any use cases or process flows missing?
- ❑ Are any alternative flows, exceptions, or other information missing from use cases?
- ❑ Are all of the business rules identified?
- ❑ Are there any missing visual models that would provide clarity or completeness?
- ❑ Are all necessary report specifications present and complete?

Software Requirements. Third Edition. Wiegers and Beatty.

Prototyping



https://www.justinmind.com/prototyping/paper-prototype

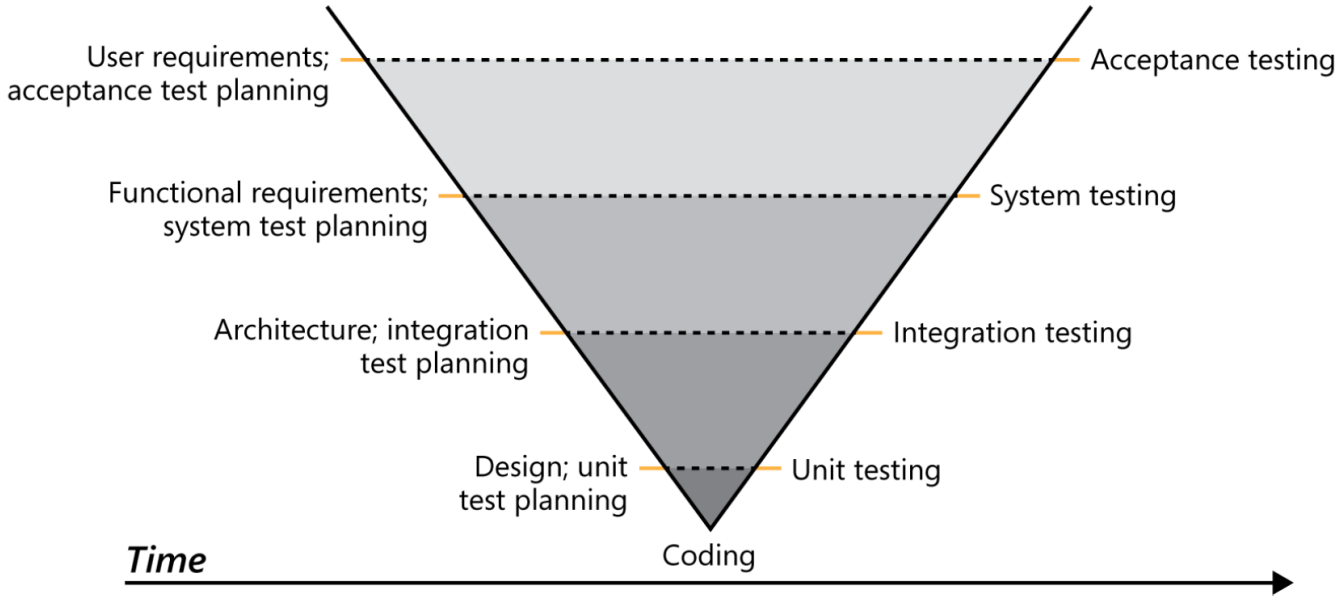
Feasibility Study

- Does the system contribute to the **overall objectives** of the organization?
- Can the system be implemented **within schedule and budget** using current technology?
- Can the system be **integrated with other systems** that are used?

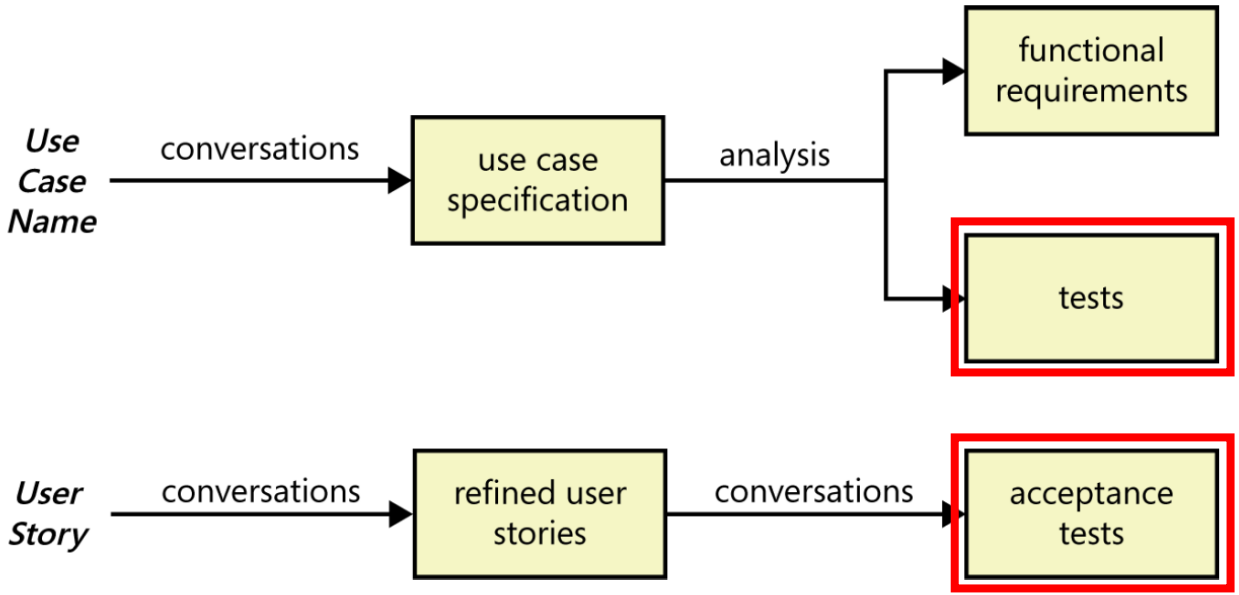
Requirements Validation and Testing

Image generated by Gemini

V-Model and Testing



Specification and Validation: Use Cases and User Stories



"Software Requirements". Third Edition. Wiegers and Beatty

Summary and Key Points

- Informal reviews
- Inspections
- Prototypes
- Feasibility studies
- Testable requirements