Requirement Engineering
Prioritize and Validate

References: C1.Ebook +John Vu (CMU)

Requirements Prioritization & Validation

- Engineers always work under certain constrained conditions (time, cost, quality), and can not implement all requirements as stated.
- Requirements must be prioritized and validated to ensure that they provide an accurate account of stakeholder requirements.
- Requirements validation is difficult because it has to solve the question of truth and what is knowable, as well as reaching agreement among different stakeholders with conflicting goals.
- These activities require intensive communication and negotiation between stakeholders and project teams.

Constraints

- Expectations are high but time and resources are short, Software Engineers must make sure the project delivers the final product on time, within cost and still satisfy stakeholders:
  - With these Requirements
  - With limited Budgets
  - With strict Schedules
  - Within these Constraints
- How can a development team build a system that meets the business objectives and satisfy stakeholders?
  - Answer: Prioritize requirements by high priorities and eliminate or defer lower priorities to a later release.

Prioritization Technique 1

- Review all requirements with stakeholders and have them prioritize by **Vote** (Yes, No) to identify:
  - Must have (Essential - High)
  - Should have (Desirable- Medium)
  - Nice to have (Optional - Low)
- **Must Have**: Without these functions, system is NOT a system, business problem will NOT be solved.
- **Should Have**: Very important features that distinguish a system from others, significant to users and business.
- **Nice to Have**: Other features that could enhance the system but not significant enough.

Activities

- Group discussion (10 mins):
  - List your CDIO project requirements
  - Identify, which requirements:
    - Must have
    - Should have
    - Nice to have

Prioritization Technique 2

- Requirements Prioritization Based on Importance and Urgency

	Important	Not Important
Urgent	High Priority	Don't do these!
Not Urgent	Medium Priority	Low Priority

Prioritization Technique 3

- Estimate the relative value and cost of each requirement.
- Priority: requirements that have the largest fraction of the total product value at the smallest fraction of the cost.
  - Have stakeholders estimate the relative benefit that each requirement would provide value, on a scale from 1 to 9 where 9 is the highest.
  - Have project teams estimate the relative cost to implement each requirement using the same scale.
  - Have project team estimate the relative risk to implement each requirement (same scale).

Prioritization Technique 3

- $$\frac{\text{value \%}}{(\text{cost \%} * \text{cost weight}) + (\text{risk \%} * \text{risk weight})}$$
- The typical participants in the prioritization process include:
  - The project manager, who leads the process, arbitrates conflicts, and adjusts input from the other participants if necessary
  - Customer representatives, such as product champions or marketing staff, who supply the benefit and penalty ratings
  - Development representatives, such as team technical leads, who provide the cost and risk ratings

Validating the Requirements

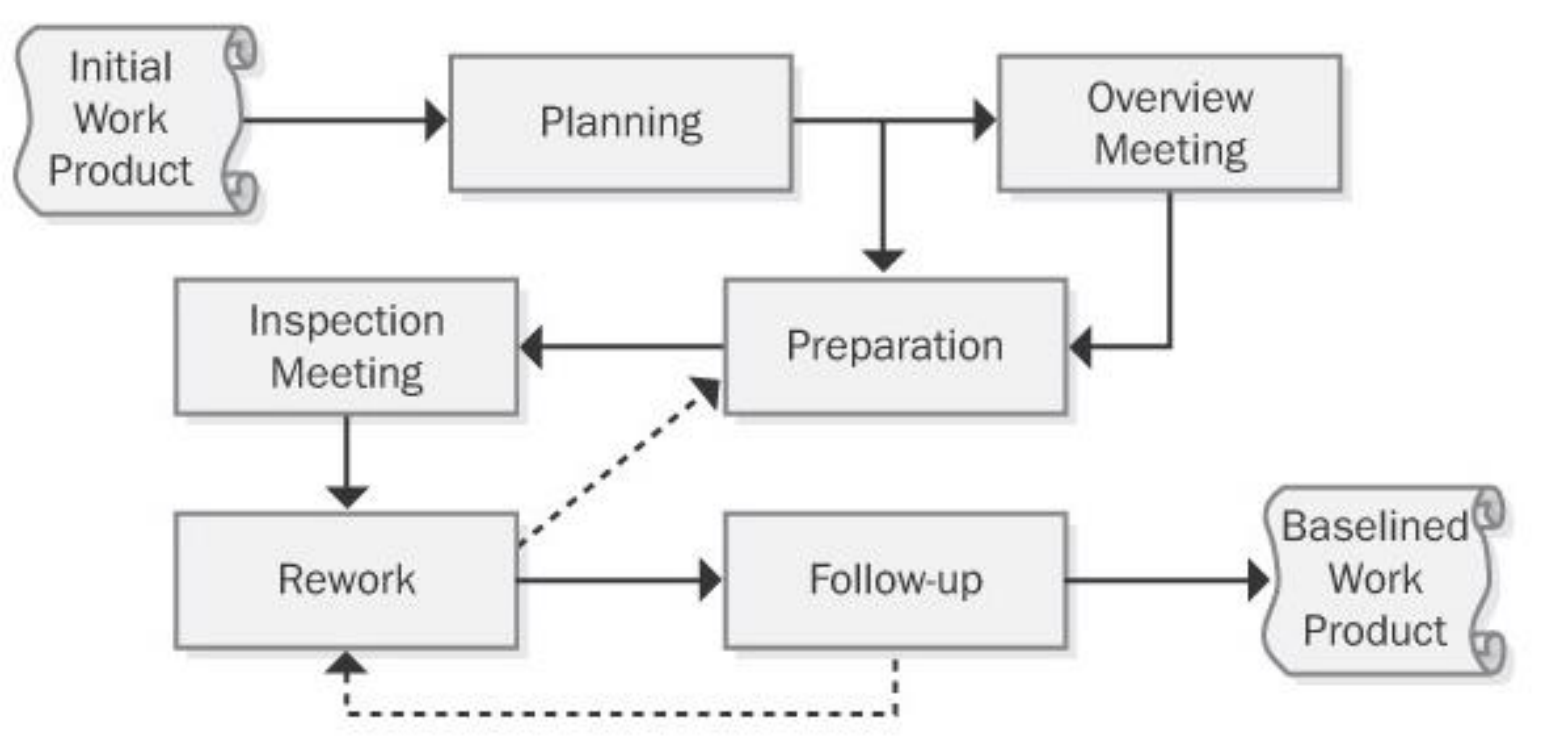
- Reviewing Requirements
- Testing the Requirements

Reviewing Requirements

- Informal
  - 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

Reviewing Requirements

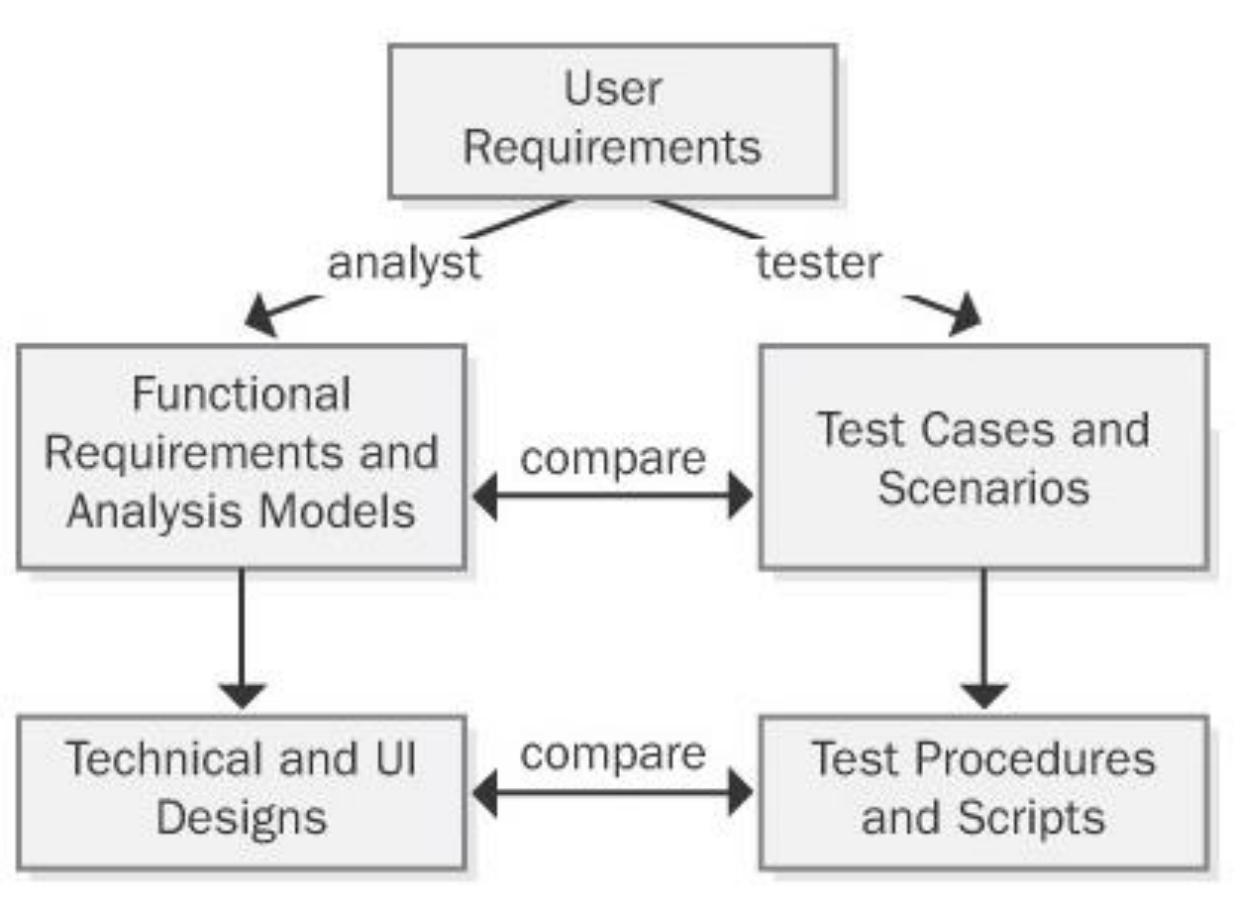
- Formal - Inspection



Testing the Requirements

- The simple act of designing test cases will reveal many problems with the requirements even if you don't execute the tests on an operational system (Beizer 1990).
- If you begin to develop test cases as soon as portions of the requirements stabilize, you can find problems while it's still possible to correct them inexpensively.

Testing the Requirements



Activity

- Design test case for the key UseCase in CDIO projects
  - Sample data

Requirements Validation

1. Validation with Stakeholders
   - To formally verify requirements to make sure they meet their needs.
2. Validation with Management:
   - To provide confidence that the requirements are reasonable and in alignment with business goals and objectives.
   - Sometimes attended by both management and stakeholders.
3. Validation with Project Team:
   - To clarify some quality attributes or find defects.

Validation with Stakeholders

- A list of Requirements sorted by attributes:
  - Priority
  - Cost
  - Risk
  - Volatility
  - Dependencies (Relationships between requirements)
- Stakeholders could change priority order.
- Resolve conflicts among requirements between different stakeholders, if possible.
- The approved fully-attributed requirements will serve as a **baseline for future changes**.

Validation by Management

- Ensure requirements meet business needs.
- Ensure requirements are in alignment with business goals and objectives.
- Verify "business case" of requirements.
- Clarify that all requirements are documented correctly

Validation By QA

- Quality Assurance reviews requirements to:
  - Identify any standard non-compliance.
  - Ensure it follows organizational templates and guidelines.
  - Ensure it is documented, well written, clear, and complete.
  - Ensure it can be used by 'intended readership'.
  - Check before baselined by Configuration management.

Approved Requirements

- A good requirements specification is:
  - Documented
  - Clear and concise
  - Understood
  - Testable
  - Usable
  - Traceable
  - Verifiable

Check list

- Are the requirements complete?
- Are all requirements uniquely identifiable?
- Are the requirements clearly and appropriately prioritized?
- Are the requirements consistent? (no contradictions)
- Does the set of requirements adequately address all appropriate exception conditions?
- Does the set of requirements adequately address boundary conditions?
- Are the requirements feasible? (a solution exists)
- Can the requirements be implemented within known constraints?
- Are the requirements sufficient? (i.e., they could be sent to software development team and have a reasonable probability of implementing the product that was desired)

Check list

- Are requirements explicitly stated?
- Do the set of requirements meet the stakeholder's needs?
- Are all cross-references to other requirements correct?
- Have functional and non-functional requirements been considered?
- Is the requirement precise and unambiguous?
- Is the requirement stated simply as possible?
- Is the requirement testable/verifiable?
- Is the requirement correct?
- Is the requirement in scope? (i.e., the system will be considered incomplete if even one requirement is left out)
- Is the statement of the requirement expressed only in terms of *what* and *why*, rather than *how*?

Check list

- Does the requirement meet a stated stakeholder need?
- Is the requirement both necessary and sufficient?
- Is the requirement understandable without having to analyze the meaning of words?
- Does the requirement have a unique interpretation?
- Do all stakeholders interpret the requirement in the same way?
- Is the requirement redundant?
- Does the requirement conflict with others?
- Does the requirement contain errors of fact?
- Is it physically possible to meet the requirement using existing technologies?
- Can the requirement be met within the approved schedule and budget?