



มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

Software Requirements Engineering

วิศวกรรมความต้องการซอฟต์แวร์

Suphitcha Chanrueang 🐰

1



มหาวิทยาลัย
ราชภัฏนครปฐม

Chapter 10

Requirements Management

การจัดการความต้องการ

2

Outline

- Introduction to Requirements Management
- Stages of Requirements Lifecycle
- Requirements Change Management
- Requirements Traceability

3

Objectives

- Understand the importance and role of requirements management in project success.
- Learn the stages of the requirements lifecycle and their contributions to the project.
- Master the process of managing changes in requirements effectively.
- Understand the concept and importance of tracking requirements throughout the project.

4

Introduction to Requirements Management

Requirements Management is the process of documenting, analyzing, tracking, prioritizing, and controlling changes to requirements.

Importance: Ensures that all stakeholders have a common understanding of the requirements, and that the final system meets business needs.

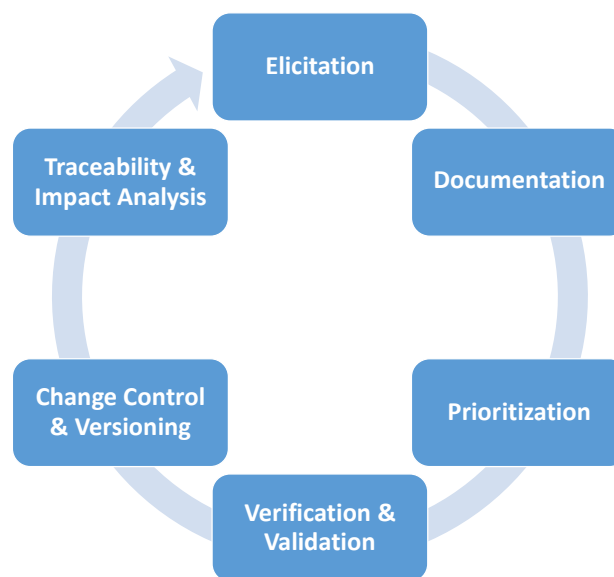
Common Challenges:

- Scope Creep (uncontrolled requirement expansion)
- Poorly defined requirements
- Inefficient communication between stakeholders
- Difficulty in tracking changes



5

Stages of Requirements Lifecycle



6

Requirements Change Management

Why Do Requirements Change?

- Evolving business needs
- Regulatory updates
- User feedback and testing



Change Control Process

- Change Request Submission – Identify what needs modification
- Impact Analysis – Evaluate how the change affects the system
- Approval/Rejection – Decision-making process by stakeholders
- Implementation & Documentation – Updating the requirement repository

7

Requirements Traceability

Types of Traceability:

1. Forward Traceability
2. Backward Traceability
3. Bidirectional Traceability

8

Requirements Traceability

Example of Forward Traceability

Scenario:

1. Requirement: "The system should allow users to reset their password using an email verification link."
2. Design: The system's design includes a "Forgot Password" page where users can enter their email to receive a verification link.
3. Development: Developers implement the "Forgot Password" functionality that sends an email with a reset link to the user.
4. Testing: Test cases are created to verify that the "Forgot Password" feature works correctly by checking the email functionality, link validity, and password reset.

9

Requirements Traceability

Example of Backward Traceability

Imagine a requirement for a mobile banking app that states: The user must be able to view their account balance in real-time.

1. Requirement: The user should see their account balance in real-time.
2. Design: The design documents specify an interface that fetches the balance from the server every minute and displays it to the user.
3. Implementation: The developers create the functionality where the app sends a request to the server every 60 seconds to retrieve the balance.
4. Testing: Test cases are created to verify that the account balance is displayed correctly and is updated in real-time every minute.

10

Requirements Traceability

Bidirectional Traceability Example

Scenario: You are working on a project to develop an online shopping platform.

1. Forward Traceability (Requirement to Test Case):

1. Requirement: "The system must allow users to add items to their shopping cart."
2. Test Case: "Verify that users can add a product to their shopping cart successfully."

2. In this case, you can trace the requirement (add items to the shopping cart) to the test case (ensuring the functionality works), ensuring that the requirement is tested.

3. Backward Traceability (Test Case to Requirement):

1. Test Case: "Verify that users can add a product to their shopping cart successfully."
2. Requirement: "The system must allow users to add items to their shopping cart."

4. In this case, you can trace the test case back to the original requirement, ensuring the test is aligned with the requirement and no unnecessary or unimportant test cases exist.

11

Requirements Traceability Matrix (RTM)

Requirement ID	Requirement Description	Test Case ID	Test Case Description	Test Status	Comments
REQ-001	The system must allow users to register	TC-001	Test user registration functionality	Passed	-
REQ-002	The system must allow users to log in	TC-002	Test user login functionality	Failed	Login function needs to be fixed
REQ-003	The system must allow users to reset passwords	TC-003	Test password reset functionality	Passed	-

12

Benefits of Traceability

Ensures complete coverage of requirements

Helps manage changes efficiently

Reduces project risks by preventing missed requirements

13

Impact Analysis

Why Impact Analysis is Critical?

- Identifies consequences of changing a requirement
- Helps assess risks and resource needs

Steps in Impact Analysis:

- Identify impacted components
- Evaluate cost and effort required
- Communicate with stakeholders

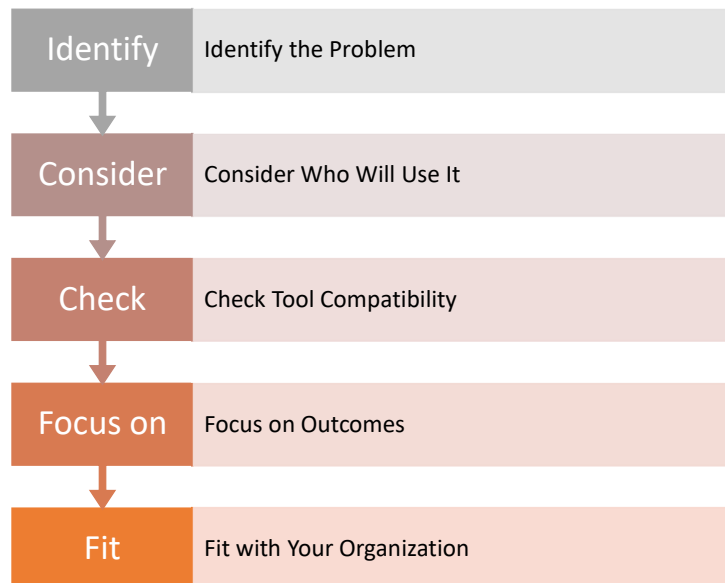
14

Tools for Requirements Management



15

How to Choose Requirements Management Software



16

Case Study – Healthcare.gov Failure

Issues:

1. Over 100 changes to requirements made last-minute
2. Poor coordination between contractors
3. Unclear system performance requirements



Lessons Learned:

1. Establish a well-defined change management process
2. Prioritize load testing and scalability requirements

17

Best Practices in Requirements Management



Maintain a central repository for requirements



Use version control for tracking changes



Conduct regular stakeholder reviews



Keep requirements clear, concise, and testable



Continuously validate and refine requirements



Avoid ambiguities and assumptions

18

Conclusion

Requirements Management in Software Development

Importance: Effectively managing clear and organized requirements enables development teams to deliver products that meet user expectations.

Monitoring and Controlling: Helps minimize risks and prevent errors during development and system deployment.

Tools and Techniques: Such as identifying and prioritizing requirements, verifying and validating requirements, and managing changes.

Outcomes: Strong requirements management leads to projects being completed on time, within budget, and in line with user expectations.

19

Thank You



20