มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

Software Requirements Engineering
วิศวกรรมความต้องการซอฟต์แวร์

Suphitcha Chanrueang 🐰

1

มหาวิทยาลัย
ราชภัฏนครปฐม

Chapter 2

Concept to Implementation

แนวคิดสู่การปฏิบัติ

2

Outline

- Types of Requirements
- Challenges in Software Requirements Engineering

3

Objectives of the Lesson

- Understand Different Types of Requirements
- Learn the Requirement Engineering Process
- Recognize and Address Common Challenges
- Apply Knowledge Through Practice

4

Types of Requirements

- Functional Requirements

  What the system must do, such as calculations or data storage

- Non-Functional Requirements
  a) System characteristics, such as speed, security, and usability
  b) Business Requirements: Requirements derived from business goals

5

Types of Requirements

Functional Requirements

1. Data processing
2. Data storage and retrieval
3. User management
4. Sending notifications

6

Types of Requirements

***Functional Requirements* Examples:**

1. The system must accurately calculate the product price after applying discounts.
2. The system must allow users to create new accounts.
3. The system must send a confirmation email to users after completing the registration process.
4. The system shall allow users to browse products by category.
5. The system shall allow users to add products to their shopping cart.
6. The system shall calculate the total cost of items in the shopping cart, including taxes and shipping fees.
7. The system shall allow users to pay for their orders using various payment methods (e.g., credit card, PayPal).
8. The system shall send a confirmation email to the user after an order is placed.
9. The system shall allow administrators to manage product inventory.
10. The system shall allow users to search for books by title, author, or ISBN.

7

Types of Requirements

Non-Functional Requirements

1. Performance: The system must respond to requests within 2 seconds
2. Security: The system must encrypt all user data
3. Reliability: The system must have an uptime of at least 99.9% per month
4. Scalability: The system must support up to 10,000 concurrent users
5. Usability: Users must be able to learn how to use the system within 10 minutes

8

Challenges in Software Requirement Engineering

Discusses common challenges in the requirement engineering process and how to address them

1. Changing Requirements
2. Ambiguity in Requirements
3. Poor Communication

9

Challenges in Software Requirement Engineering

1. **Changing Requirements:** Requirements that evolve over time

   Causes of Changing Requirements

   1) Business Needs Change
   2) User Expectations Evolve
   3) Incomplete Analysis
   4) Technological Changes
   5) Stakeholder Changes

10

Challenges in Software Requirement Engineering

Changing Requirements:

Example 1: E-commerce System

- Initial Requirement: The system should display product catalogs and provide online payment options.
- Change: After testing, users request a "live chat" feature to communicate with support staff.
- Impact: Development time increases to implement and test the new feature, causing delivery delays.

Example 2: Health Monitoring Application

- Initial Requirement: The app should log basic health data like blood pressure and weight.
- Change: Users request an automatic alert when blood pressure exceeds a specified threshold.
- Impact: Developers need to analyze, design, and integrate this new functionality, requiring additional resources.

11

Challenges in Software Requirement Engineering

2. Ambiguity in Requirements: Unclear or vague requirements

Causes of Changing Requirements

1) Unclear Language
2) Incomplete Requirements
3) Lack of Context
4) Poor Communication with Stakeholders

12

มหาวิทยาลัย ราชภัฏนครปฐม

Challenges in Software Requirement Engineering

Examples of Ambiguity in Requirements

- Ambiguous Functional Requirement:
  - *Requirement:* "The system should provide real-time data."
  - *Ambiguity:* What does "real-time" mean? Is it a 1-second, 1-minute, or 1-hour refresh?
  - *Resolution:* Specify the time frame, e.g., "The system should refresh data every 5 seconds."
- Ambiguous Non-Functional Requirement:
  - *Requirement:* "The application should be secure."
  - *Ambiguity:* What level of security is expected? Encryption type? Compliance standards?
  - *Resolution:* Define security standards, e.g., "The application should use AES-256 encryption and comply with GDPR."
- Contradictory Requirement:
  - *Requirement:* "The system should log out inactive users after 5 minutes but maintain their session."
  - *Ambiguity:* Conflicting actions – log out vs. maintain session.
  - *Resolution:* Clarify, e.g., "The system should log out inactive users after 5 minutes and terminate their session."

13

มหาวิทยาลัย ราชภัฏนครปฐม

Challenges in Software Requirement Engineering

3. Poor Communication :

Causes of Poor Communication

1) Lack of Stakeholder Involvement
2) Unclear Channels of Communication
3) Cultural and Language Barriers
4) Use of Jargon or Technical Terms
5) Failure to Clarify Assumptions
6) Lack of Active Listening

14

Challenges in Software Requirement Engineering

Examples of Poor Communication

1. Unclear Requirement:
   - *Scenario:* A stakeholder requests, "The system should be simple to use."
   - *Impact:* The development team delivers a basic interface, but the stakeholder actually wanted a detailed onboarding tutorial.
2. Missed Stakeholder Input:
   - *Scenario:* A project manager consults only upper management and excludes feedback from end-users.
   - *Impact:* The resulting system does not address end-user needs.
3. Confusion Due to Jargon:
   - *Scenario:* Developers discuss "CI/CD pipelines" with non-technical stakeholders without explaining the term.
   - *Impact:* Stakeholders misunderstand the project's progress and timeline.

15

Challenges in Software Requirement Engineering

Prevention Methods of Changing Requirements

1. Engage stakeholders early and continuously.
2. Define a clear project scope.
3. Implement change control processes.
4. Prioritize requirements.
5. Use agile methodologies to manage flexibility.

16

Challenges in Software Requirement Engineering

Prevention Methods of Ambiguity in Requirements

1. Document requirements clearly and thoroughly.
2. Use visual aids like diagrams or prototypes.
3. Conduct reviews to validate understanding.
4. Standardize terminology.
5. Allow open questions to clarify uncertainties.

17

Challenges in Software Requirement Engineering

Prevention Methods of Poor Communication

1. Set up clear communication channels.
2. Keep comprehensive and accessible documentation.
3. Use simple, jargon-free language.
4. Encourage feedback and validation.
5. Provide training for effective communication.
6. Schedule regular meetings for updates and alignment.

18

มหาวิทยาลัย ราชภัฏนครปฐม

Conclusion

By understanding the different types of requirements and the challenges involved in managing them, software development teams can improve their processes and increase the likelihood of delivering successful software projects.

19

มหาวิทยาลัย ราชภัฏนครปฐม

Class Activity

1. Ask students to write examples of Functional and Non-Functional Requirements for a simple application (e.g., a movi ticket booking app)
2. Divide students into groups to analyze problems and propose solutions based on real-world case studies.

20

มหาวิทยาลัย
ราชภัฏนครปฐม

Thank You

21