มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

Software Requirements Engineering
วิศวกรรมความต้องการซอฟต์แวร์

Suphitcha Chanrueang 🐰

Chapter 4

Non-Functional Requirements

ความต้องการที่ไม่ใช่เชิงฟังก์ชัน

Outline

- Understanding Non-Functional Requirements
- Detailed Exploration of Non-Functional Requirement Categories
- Challenges in Defining Non-Functional Requirements

3

Understanding Non-Functional Requirements

- **What Are Non-Functional Requirements?**

Imagine a car. Functional requirements would describe what the car does (move from point A to B), while non-functional requirements explain how well it does it. These are the critical quality attributes that define the system's behavior, performance, and overall user experience.

4

Understanding Non-Functional Requirements

Core Characteristics of Non-Functional Requirements

Key Dimensions

1. Performance
2. Security
3. Usability
4. Reliability
5. Scalability
6. Maintainability

Detailed Exploration of Non-Functional Requirement Categories

1. Performance Requirements

What They Define

- System response times
- Processing speed
- Throughput capacity
- Resource utilization

Example:

"The system must process 1000 transactions per second"

"Maximum response time should be under 2 seconds"

Detailed Exploration of Non-Functional Requirement Categories

2. Security Requirements

Critical Considerations

- Data protection mechanisms
- Authentication protocols
- Access control systems
- Encryption standards
- Compliance with regulatory frameworks

- **Example**:
- "User passwords must be encrypted using AES-256 standard"
- "System must support multi-factor authentication"

7

Detailed Exploration of Non-Functional Requirement Categories

4. Reliability Requirements

System Dependability

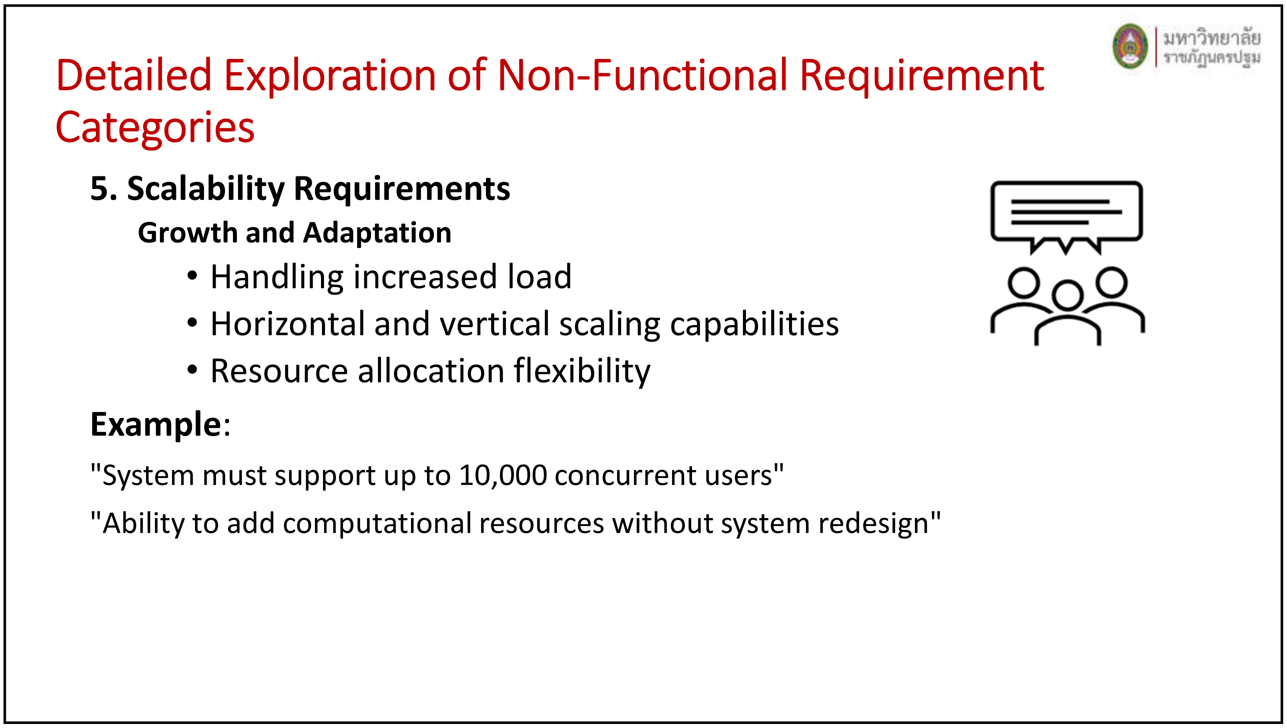
- Mean time between failures
- Error handling capabilities
- System recovery mechanisms
- Fault tolerance

Example:

"System must maintain 99.99% uptime"

"Automatic failover within 30 seconds of primary system failure"

8



Detailed Exploration of Non-Functional Requirement Categories

5. Scalability Requirements

Growth and Adaptation

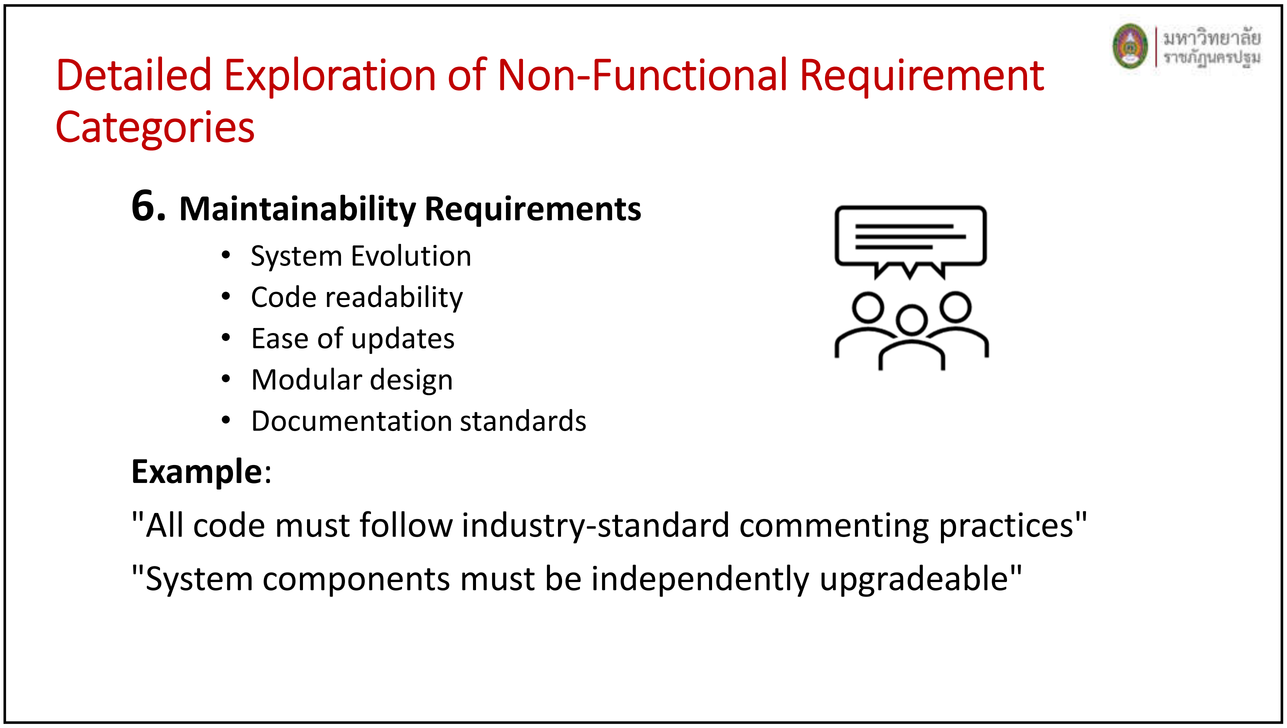
- Handling increased load
- Horizontal and vertical scaling capabilities
- Resource allocation flexibility

Example:

"System must support up to 10,000 concurrent users"

"Ability to add computational resources without system redesign"

9



Detailed Exploration of Non-Functional Requirement Categories

6. Maintainability Requirements

- System Evolution
- Code readability
- Ease of updates
- Modular design
- Documentation standards

Example:

"All code must follow industry-standard commenting practices"

"System components must be independently upgradeable"

10

Non-Functional Requirements Examples

1. The system shall respond to user requests within 2 seconds under normal load conditions. (Performance: This specifies speed and responsiveness.)
2. The system shall protect user data from unauthorized access, including but not limited to, SQL injection, cross-site scripting, and brute-force attacks. (Security: This addresses data protection.)
3. The system shall be intuitive and easy to navigate for users with minimal technical experience. (Usability: Focuses on user-friendliness.)
4. The system shall be available 99.9% of the time, excluding scheduled maintenance. (Reliability: Specifies uptime and availability.)
5. The system shall be designed in a modular fashion to facilitate easy updates and bug fixes with minimal downtime. (Maintainability: Concerns ease of modification.)

11

Non-Functional Requirements Examples

6. The system shall be able to handle a 50% increase in user traffic within the next 6 months without significant performance degradation. (Scalability: Addresses future growth.)
7. The system shall be compatible with the latest versions of Chrome, Firefox, and Safari web browsers. (Portability: Specifies platform compatibility.)
8. The system shall comply with WCAG 2.1 Level AA accessibility guidelines to ensure usability for people with disabilities. (Accessibility: Addresses inclusivity.)
9. The system shall support multiple languages, including English, Spanish, and French, with appropriate date and currency formatting. (Localization: Concerns internationalization.)
10. The system shall comply with all applicable data privacy regulations, including GDPR and CCPA. (Legal and Regulatory: Addresses legal obligations.)

12

Challenges in Defining Non-Functional Requirements

Common Difficulties

- Measuring abstract qualities
- Balancing competing requirements
- Translating business needs into technical specifications
- Avoiding overly restrictive or vague definitions

13

Challenges in Defining Non-Functional Requirements

Best Practices for Specification

Effective Specification Strategies

- Use Measurable Metrics
- Involve Multiple Stakeholders
- Prioritize Requirements
- Create Testable Specifications
- Consider Long-Term System Evolution

14

Importance in Software Development

Strategic Significance

Non-functional requirements are not optional luxuries but critical determinants of

- User satisfaction
- System viability
- Competitive advantage
- Long-term operational effectiveness

15

Conclusion

Non-functional requirements transform software from mere functional tools into robust, user-friendly, and strategically aligned solutions. They represent the invisible architecture that determines a system's true quality and potential for success.

By meticulously defining and implementing non-functional requirements, organizations create systems that are not just functional, but exceptional.

16

มหาวิทยาลัย
ราชภัฏนครปฐม

Thank You

17