**Software Systems Testing - Project Topics**

## **Project Requirements**

* Each topic can be chosen by a maximum of **3 teams**, except:
  + **T12 (Scientific Article Study)** and **T13 (Related Topics in Software Testing)** can be chosen more than 3 times, as long as different articles are used.
* **Deadlines:**
  + **March 23, 23:59** – Team formation and topic selection.
  + **April 13, 23:59** – **1/3 implementation** (upload to repository + lab presentation).
  + **May 4, 23:59** – **2/3 implementation** (upload to repository + lab presentation).
  + **May 16, 23:59** – **Final submission** (upload to repository).

### **Repository Requirements**

* **Must include:**
  + **Source code** of the application and tests.
  + **README or wiki**, including:
    - Documentation (strategies, diagrams, hardware/software configurations, VM usage, tool versions, code snippets, screenshots of test execution/results, tool comparisons, interpretations, references).
    - Diagrams created using professional tools (e.g., **diagrams.net, Lucidchart, Microsoft Visio**). **Hand-drawn or scanned images are NOT accepted**.
  + **Presentation** (PowerPoint, Keynote, Markdown, etc., max **10 slides** summarizing documentation).
  + **Demo videos** (e.g., YouTube, Microsoft Stream).
  + **AI Tool Report**: Compare AI-generated test cases (e.g., GitHub Copilot, ChatGPT) with manually written ones. Include screenshots, prompts, interpretations, and references.

## **Project Topics**

### **T1: Unit Testing in Python**

* Use a **Python unit testing framework** to test class functionalities.
* Demonstrate **test generation strategies** (equivalence partitioning, boundary value analysis, code coverage, mutation testing).

### **T2: Unit Testing in C#**

* Use a **C# unit testing framework** to test class functionalities.
* Apply **test generation strategies** with team-created examples.

### **T3: Unit Testing in Java**

* Use a **Java unit testing framework** to test class functionalities.
* Illustrate various test generation strategies on custom examples.

### **T4: Unit Testing in JavaScript**

* Use a **JavaScript testing framework** to test components.

### **T5: Unit Testing in PHP**

* Implement unit tests for a **PHP backend**.
* Cover different service functionalities.

### **T6: Web UI Testing**

* Compare **at least two UI testing frameworks**.
* Provide code examples demonstrating differences.
* Implement **functional, navigation, form validation, and cross-browser compatibility tests**.

### **T7: Mobile Application Testing**

* Compare **at least two mobile testing frameworks**.
* Implement **functionality, UI, performance, security, and cross-platform compatibility tests**.

### **T8: Blockchain Network Testing**

* Implement **unit, integration, performance, and security tests** for a blockchain network.

### **T9: Automated Testing with Robots**

* Develop scripts for **regression, performance, and security tests** run by robots.

### **T10: Enhancing Unit Testing with AI**

* Use AI to improve existing unit tests and increase **code coverage**.
* Implement a system to **identify critical code areas** and prioritize tests.

### **T11: Game Strategy Testing**

* Develop a **decision-based game simulation** using game theory.
* Implement and test game strategies.

### **T12: Scientific Article Study**

* Choose a **software testing article published in the last 4 years**.
* Summarize concepts and methods.
* Discuss with the professor for approval.
* Provide a demo illustrating key concepts **using original examples**.

### **T13: Related Topics in Software Testing**

* Teams may propose **new topics** related to testing strategies, tools, automation, CI testing, security testing, cloud testing, or testing software developed in other courses.
* Requires **professor approval**.

This is a detailed English translation of the original document. Let me know if you need any modifications!