KICT TEACHING AND LEARNING EXHIBITION 2025

TAWHIDIC EPISTEMOLOGY LEADING THE WAY UMMATIC EXCELLENCE LEADING THE WORLD
KHALĪFAH · AMĀNAH · IQRA' · RAHMATAN LIL-ĀLAMĪN

KULLIYYAH OF INFORMATION & COMMUNICATION TECHNOLOGY

TAWHIDIC PARADIGM IN TEACHING & LEARNING: REINTEGRATING FAITH AND KNOWLEDGE

NORMI SHAM AWANG ABU BAKAR
DEPT. OF COMPUTER SCIENCE
EMAIL: NSHAM@IIUM.EDU.MY

Thinking Like a Tester: Role Playing in Course Delivery

Abstract

Software testing education requires more than theoretical understanding, it demands the development of a tester's mindset. Integrating role-playing activities into course delivery encourages students to “think like testers,” enhancing analytical reasoning, communication, and problem-solving skills. This article discusses how role-playing can be effectively applied in software testing education to simulate real-world contexts, promote critical thinking, and prepare students for professional testing environments.

Keywords:

Software Testing, Tester Mindset, Role-Playing, Experiential Learning, Quality Assurance, Active Learning.

Introduction

In software engineering education, students often focus on coding, algorithms, and design, while testing is treated as a secondary phase. However, quality assurance and defect prevention depend heavily on the tester's ability to think critically and anticipate potential failures. The phrase "thinking like a tester" refers to adopting a mindset that questions assumptions, explores edge cases, and values evidence over intuition. Teaching this mindset effectively requires active, experiential learning methods—one of which is role-playing.

Thinking Like a Tester

To think like a tester is to view software from multiple perspectives users, developers, clients, and security experts. A tester's mindset is curious, skeptical, and analytical. It focuses on discovering what could go wrong rather than proving what works. Testers ask questions such as:

- What happens if the user enters invalid data?
- How does the system behave under extreme conditions?
- Are there hidden assumptions in the requirements?

Cultivating this mindset in students requires more than lectures. It requires experiential learning where they confront ambiguity, explore systems, and learn from failures.

Role-Playing as a Pedagogical Strategy

Role-playing transforms passive learning into active exploration. In a software testing course, role-playing can simulate real-life testing scenarios. For example:

- Tester–Developer Interaction: One group plays testers who identify bugs, while another plays developers defending their code. This encourages negotiation, technical reasoning, and communication.
- Client Acceptance Testing: Students act as clients evaluating whether a software product meets their expectations. This teaches the importance of understanding requirements and user satisfaction.
- Agile Sprint Reviews: Students role-play as QA engineers within Agile teams, practicing test planning, execution, and reporting in iterative cycles.

Benefits of Role-Playing in Software Testing Education

The integration of role-playing fosters multiple learning outcomes:

- Critical Thinking: Students learn to analyze requirements and anticipate risks.
- Communication Skills: Testers must convey defects effectively and diplomatically.
- Collaboration: Role-playing encourages teamwork across functional boundaries.
- Authentic Assessment: Instructors can evaluate students based on observed behaviors, decision-making, and applied reasoning rather than rote memorization.
- Motivation and Engagement: Simulated professional contexts make learning interactive and relevant.