STMI@NUS

STRATEGIC TECHNOLOGY MANAGEMENT INSTITUTE

SINGAPORE WORKFORCE SKILLS QUALIFICATIONS

NATIONAL INFOCOMM COMPETENCY FRAMEWORK

Jointly developed by IDA and WDA

NICF-Testing Methodologies: Delivering Quality in Application Lifecycle

Course Overview

About of Quality in Application Lifecycle

Enterprises across the globe are under pressure to deliver more with less. The current global economy is driving the need to reduce cost and free up capital that is in short supply. Today's applications have a much broader reach, extending directly out to our customers and shaping their experience with us as a company. We need to ensure high levels of application quality – functional, performance and security – to deliver a positive customer or user experience, as these applications are ultimately a reflection of our brand. A poor customer experience can be very damaging to customer brand loyalty.

However, far too often, software quality is equated to just testing and the responsibility for software quality is relegated to the testing team. Because of this skewed focus, software quality has not really improved over the last couple of decades despite the advancements and innovations in software development processes and technology. According to Wikipedia, software quality refers to two related but distinct notions that exist wherever quality is defined in a business context:

- Software functional quality reflects how well it complies with or conforms to a given design, based on functional requirements or specifications. That attribute can also be described as the fitness for purpose of a piece of software or how it compares to competitors in the marketplace as a worthwhile product;
- Software structural quality refers to how it meets non-functional requirements that support the delivery of the functional requirements, such as robustness or maintainability, the degree to which the software was produced correctly.

Also, ITIL v3 Service Transition defines Service Validation and Testing as the testing of services to ensure that new or changed services are fit for purpose and fit for use. Therefore, delivering quality in software system is more than just testing, but a shared responsibility of different personas across the entire application lifecycle. While the core of this course will still articulate the key aspects of software testing, which is the final gate before the real world, the objective is to also provide a practical and holistic approach to considering and building quality from the definition of the requirements to the deployment of these requirements into production as an IT service and through to the retirement of the IT service.

Duration

3 days

Venue

STMI@NUS
ICube, Level3
21 Heng Mui Keng Terrace
Singapore 119613
stmi@nus.edu.sg

Contact Us

Tel: +65 6601 1040
Fax: +65 6776 2856
Email: stmi@nus.edu.sg

About this course

The 3-day course, "Testing Methodologies" has been developed to cover the following Competency Units of the National IT Competency Framework (jointly developed by the Singapore Workforce Development Agency and the Infocomm Development Authority of Singapore):

- Develop detailed test plan
- Manage the testing process
- Optimise test processes
- Design project QA governance and processes for software development
- Assess developed software
- Develop quality assurance process for business solution

Programme Benefits

After completing this course, you should be able to discuss and apply:

- Understanding the basics and definitions of software testing and principles
- Different testing models
- How to test to satisfy business requirements
- Risk management and how it applies to software testing
- How to utilize reporting and Key Performance Indicators (KPI) to measure success
- Understanding of Test Automation
- Understanding of Non-Functional Testing
- Improving Software Quality through Center of Excellence

Programme Outline

I. Introduction to Quality in Application Lifecycle Management

- Software Quality and Testing Fundamentals
- Trichotomy of Software Quality
- Application Lifecycle Management and ITIL Service Transition
- Software Development Models

II. Software Quality Management

- Requirements Management
- Test Strategy and Practices
- Test Planning and Development
- Risk-based Testing
- Regression and Test Automation
- Non-Functional Testing
- Governance and Measuring Success

III. Automating Software Quality Management

- Software Tools Overview
- The Right Tool for the Task
- Tools Introduction Considerations

IV. Organizing to Deliver Software Quality

- Capability and Competency
- The Human Factor
- Centralized, Distributed or Outsourced

V. Software Quality Process Improvement

- Introduction to Standard and Models
- Introduction to Center of Excellence

Who Should Attend

IT Managers, IT Application Project Manager, Software Development Engineers, Business/System Analysts, Software Testing/Quality Professionals, and IT professionals who are involved or concerned with Application Quality.

Course Leader

Mr. Arthur Lim
ALM Practice Lead for Hewlett-Packard Software Professional Services

Arthur Lim is currently the ALM Practice Lead for Hewlett-Packard Software Professional Services responsible for providing consulting to and partnering with many HP customers in the region in their journey toward IT service excellence, specifically in the domain of Software Quality and Testing in the Application Lifecycle Management and Center of Excellence. Arthur also represented HP in the Technical Committee for the development of the National Infocomm Competency Framework (NICF) for building competencies of IT professionals in Singapore.

Arthur holds a bachelor's degree in Computer Science and Information Systems from the National University of Singapore and is ITIL v3 certified with more than 20 years of experience in diverse areas of IT ranging from software development, consulting, software product sales and support, and software quality implementation and management. More recently prior to HP, Arthur was the Director of QA (Singapore) in Savi Technology, a wholly-owned subsidiary of Lockheed Martin, responsible for architecting and leading a cross-organization QA practice, infrastructure and processes.