AUTOMATION TESTER

Automation can be used in two contexts; for the facilitation of testing and for the verification of behaviour. Testers who are not dedicated automation testers can leverage the facilitation of testing aspect whilst dedicated automation testers will be proficient in both.

The dedicated automation tester usually takes one of two roles; Interface testing or UI automation. The former can be subdivided into the discrete application components (batch processes, services, APIs, etc). As a group they are defined by the tester's direct interaction with the application component interface to test it. The latter is primarily used for creating build verification tests and regression suites.

The automated discipline is also responsible for developing new testing tools or using automation techniques to simplify or automate process.

The automated skill set it broken into three lists; common skills and then skills associated with interface testers and UI automation testers.

Common Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Has a solid understanding of the concepts of software development and can apply them in development automation test cases or a automation test framework Concept					
Understands when automation is the ideal approach for behaviour verification and when it should not be used at all Concept					
Understands how change impacts automation tests Concept					
Understands the value of automation with respect to repeated execution against changing environment states Concept					
Understand the difference between identity and anonymous data and the impacts it has on automated data creation and automated test execution Concept					
Understanding the difference between data dependant and data independent tests and when to use each one Concept					
Understands the difference between testing and checking Concept					

Common Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Can communicate with other infrastructure components programmatically to assist in testing. E.g. calling a database to verify state data Technique					
Understands mocks and how they can be used for isolating systems under test. Technique					
Understands stubs and how they can be used for isolating systems under test Technique					
Write automation tests that are resilient against change Responsibility					
Create both state and input data for use within automated tests Responsibility					
Use programming languages associated with the infrastructure to assist in testing. Responsibility					
Use programming languages associated with the automation tool. Responsibility					
Debug automation test code during development and fix broken automation scripts. Responsibility					
Use the organisation's automated testing tool set to execute automated tests Responsibility					
Writing new automation tools Responsibility					
Evaluate difference automation tools to identify which one is best for the organisation given specific criterion Responsibility					
Define processes for automation code reviews Responsibility					

Interface Testing Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Understanding the role the component plays in the organisation and the impact that has on testing Concept					
Understands the difference between RPC, Message and Message Bus invocation mechanisms and how that impacts testing. Concept					
Understands component configuration based on technologies selected by developers Concept					
Understands the Law of Leaky Abstractions and how that can manifest in implied dependencies in the interface. Concept					
Understands the difference between SOAP, WCF and RESTful services and how they impact service testing Concept, Specialised					
Understands service discovery and how that impacts service integration testing. Concept, Specialised					
Understands the difference between service orchestration and service choreography and how they impact service integration tests Concept, Specialised					
The analysis of component contracts to identify design issues Technique					
The analysis of component documentation to identify design issues Technique					
Analyse the component contract to create test scenarios Responsibility					
Analyse component documentation to create test scenarios Responsibility					
Identify a change in component contract and update test accordingly Responsibility					
Automation the configuration of a component for testing Responsibility					
Configuring the component tool to work with non-trivial component configurations Responsibility					
Understands the standards associated with component development (SOAP, WCF standards, etc) Responsibility					
Communicate effectively with developers to rely potential design or implementation issues Responsibility					

Interface Testing Aspects	Level 1	Level 2	Level 3	Level 4	Level 5
Input Validation – *testing the component contract with varying forms of invalid data. Considers invalid data in terms of the contract (technical) and invalid data in terms of the business (semantic).*					
Functionality – *testing the functionality of the service in isolation*					
Security – *working with the security tester to test the security characteristics of the service from infrastructure and message security to business security implementations.*					
Compliance – *testing the component to ensure it complies with applicable standards.*					
Exception Reporting – *testing the component to ensure that it reports exception information correctly and appropriately.*					
Auditing – *working with the security tester to ensure the component correctly audits requests made to it and that the audit information is secure.*					
Data Persistence Integrity – *ensuring that any information persisted by the component is done in a consistent manner.*					
Negative Testing – *executing a positive test path scenario over an incorrectly configured component to ensure the component gracefully handles the error.*					
Interoperability – *Testing the component so that it can be used with different technologies (Java vs. dotNet, etc)*					

UI Automation Testing Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Understands the complexities associated with writing checks for different UI technologies and how that impacts check authoring. Concept					
Understands the impacts of change on UI automation checks and how to alleviate these issues. Concept					
Understands the purpose of UI automation and that it is not always focused on identifying bugs that exist now Concept					
Understands the additional complexities identity data places on UI automation checks especially when it results in custom workflows Concept					
Is able to identify when to use UI automation to create data and when to use non UI based techniques. Technique					
Liaise with the other test disciplines to devise appropriate automation scenarios Responsibility					
Develop automation scripts that are resistant to changes in the content and layout of the user interface Responsibility					
Work with performance testers to develop automation performance tests that with the user interface Responsibility					

There are no aspects associated with UI Automation as it is covered under the User Interaction tester or under the Behavioural and Functionality Tester.