INFRASTRUCTURE & INTEGRATION TESTER

The Infrastructure & Integration tester is a technical tester that focuses on the integration of systems and how the systems interact with the environment.

The infrastructure discipline involves having knowledge of physical components like network topologies, virtualization and architecture designs and how this environment interacts with the system under test.

The integration aspect focuses on ensure the system components integrate with each other irrespective of environment influence.

Core Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Understands the different integration techniques: Top-down, Bottom-Up and Big Bang and when to apply it Concept					
Understands what infrastructure is deployed in the environment and how that impacts the system under test Concept					
Understands what systems are deployed in the environment and how that impacts the system under test Concept					
Understands the concepts of configuration management Concept					
Understands the production administration teams are users of the system and that the developed solution is usable for them. Concept					
Understands difference backup strategies and their uses Concept					
Understands component configuration based on technologies selected by developers Concept					
Can apply fuzz testing for the purpose of robustness testing Technique					
Factor the impacts of platforms have on test outcomes during test design and prepare cases that exercise the different browsers, virtual machines and operating systems Responsibility					
Factor the impacts of environments on test outcomes during test design and prepare cases that determine the impact of relevant hardware, peripherals and virtualisation. Responsibility					

Core Skills	Level 1	Level 2	Level 3	Level 4	Level 5
Analyse the system architecture in order to determine test scenarios Responsibility					
Perform administration of infrastructure components for the purposes of test execution and data creation Responsibility					
Communicates with system administration teams to get their perspective on the system within the environment. Responsibility					
Verify organisational system operations can monitor system behaviour and can respond accordingly to system failure. Responsibility					

Aspects	Level 1	Level 2	Level 3	Level 4	Level 5
Deployment – *focuses on the act of deployment whether via user installation or desktop rollout techniques. Deployment testing looks at whether the application can be installed easily with no manual steps, the workflow for installation is correct and that the application works on all supported hardware and software configurations.*					
Failover – *Failover is the capability for infrastructure components to switch over to secondary components in the event of failure. Testing looks at whether the setup failover configuration works correctly by running failover scenarios.*					
Backup & Recovery – *tests the backup and recovery procedures for an application or infrastructure component. Backup & Recovery should investigate all components that can fail and ensure that the system can be restored in the event of failure.*					
Disaster Recovery – *tests the disaster recovery plan. Simulate disasters by disabling infrastructure in a controlled way and then following the plan to ensure that it works.*					
Integration – *tests the integration of system components with respect to a project or change. Includes both project delivered components, existing components, 3rd party components (browser upgrades, security patches, etc) and 'Off the Shelf' products*					
Instrumentation – *Instrumentation is data provided by an application during runtime. This can be used to diagnose or identify developing issues. Instrumentation testing focuses on ensuring the content is correct and written at appropriate times.*					
Robustness – *Looks at how well a system handles invalid or incorrect configuration. Robustness of input is covered in User Interaction or Business & Functionality Testing so we focus exclusively on the interactions between the system and its components or the system and the environment.*					
Interoperability – *focuses on testing system components can be communicated with using different technologies. For example interoperability testing can be done on a service to ensure it can be consumed by both .NET and Java clients.*					