Test Levels

Attributes of test levels

	Component Testing	Integration Testing	System Testing	Acceptance Testing (UAT, OAT, CAT, RAT, alpha and beta testing)
	focuses on components that are separately testable	focuses on interactions between **components** or **systems**	both focus on the behavior and capabilities of a whole system or product	
Objectives	- Reducing risk - Verifying if the functional and non-functional behaviors of the component are as designed and specified - Building confidence in the component quality - Finding defects in the component - Preventing defects from escaping to higher test levels	- Reducing risk - Verifying if the functional and non-functional behaviors of the interfaces are as designed and specified - Building confidence in the quality of the interfaces - Finding defects (which may be in the interfaces themselves or within the component or system) - Preventing defects from escaping to higher test levels	- Reducing risk - Verifying if the functional and non-functional behaviors of the system are as designed and specified - Validating that the system is complete and will work as expected - Building confidence in the quality of the system as a whole - Finding defects - Preventing defects from escaping to higher test levels or production	- Establishing confidence in the quality of the system as a whole - Validating that the system is complete and will work as expected - Verifying that the functional and non-functional behaviors of the system are as designed and specified
Test basis	- Detailed design - Code - Data model - Component specification	- Software and system design - Sequence diagrams - Interface and communication protocol specifications - Use cases - Architecture at component or system level - Workflows - External interface definitions	- System and software requirements specification (functional and non-functional) - Risk analysis reports - Use cases - Epics and user stories - Models of system behavior - State diagram - System and user manuals	- Business processes - User or business requirements - Regulations, legal contracts and standards - Use cases and/or user stories - System requirements - System or user documentation - Installation procedures - Risk analysis reports
Test objects	- Components, units or modules - Code and data structures - Classes - Database modules	- Subsystems - Databases - Infrastructure - Interfaces - APIs - Microservices	- Applications - Hardware/software systems - Operating systems - System under test (SUT) - System configuration and configuration data	- System under test - System configuration and configuration data - Business processes for a fully integrated system - Recovery systems and hot sites - Operational and maintenance processes - Forms - Reports - Existing and converted production data
Typical defects and failures	- Incorrect functionality (not as described in design specification) - Data flow problems - Incorrect code and logic	- Incorrect data, missing data, or incorrect data encoding - Inconsistent message structures between systems - Incorrect sequencing or timing of interface calls - Interface mismatch - Failures in communication between components / systems - Unhandled or improperly handled communication failures between components / systems - Incorrect assumption about the meaning, units, or boundaries of the data being passed between components / systems - Failure to comply with mandatory security regulations	- Incorrect data calculations - Incorrect or unexpected system functional or non-functional behavior - Incorrect control and/or data flows within the system - Failure to properly and completely carry out end-to-end functional tasks - Failure of the system to work properly in the system environment(s) - Failure of the system to work as described in system and user manuals	- System workflows do not meet business or user requirements - Business rules are not implemented correctly - System does not satisfy contractual or regulatory requirements - Non-functional failures such as security vulnerabilities, inadequate performance efficiency under high loads, or improper operation on a supported platform

There are also **specific approaches and responsibilities** for each of the above test levels.