

# CTFL - Agile Extension

1. The exit criteria that a component or system must satisfy in order to be accepted by a user, customer or other authorized entity. A. Unit Test Framework
2. A statement on the value that underpins Agile software development B. Test Strategy
3. A group of software development methodologies based on iterative incremental development where requirements and solutions evolve through collaboration between self-organizing cross-functional teams C. Software Lifecycle
4. A set of automated tests which validates the integrity of each new build and verifies its key and core functionality, stability, and test-ability. It is an industry practice when a high frequency of build releases occurs (e.g. agile projects) and it is run on every new build before the build is rela D. Configuration Management
5. An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process E. Test Estimation
6. A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements F. Test Execution Automation
7. An informal test design technique where the tester actively controls the decision of the tests as those tests are performed and uses information gained while testing to design new and better tests G. Iterative Development Model
8. A development life cycle where a project is broken into a series of increments, each of which delivers a portion of the functionality in the overall project requirements. The RQ's are prioritized and delivered in priority order in the appropriate increment. In some, but not all, versions of this life cycle model, each subproject follows a mini V-Model with its own design, coding & testing phases H. Performance Testing

9. A development lifecycle where project is broken into a usually large number of iterations.
10. A complete development loop resulting in a release (internal or external) of an executable product, a subset of the final product under development which grows from iteration to become the final product
11. Testing to determine the performance of a software product
12. A risk directly related to the test object
13. A product risk related to quality attribute
14. Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.
15. The period of time that begins when a software product is conceived & ends when the software is no longer available for use. Typically includes a concept phase, RQ's phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and sometimes, a retirement phase. Phases may overlap or be performed iteratively
16. The implementation of the test strategy for a specific project. Typically includes the decisions made that follow based on the test project's goal and risk assessment carried out, starting points regarding the test process, the test design techniques to be applied, exit criteria and test types to be performed
17. The use of software to perform or support test activities, e.g. test design, test management, test execution, and results checking
18. All documents from which the RQ's of a component or system can be inferred. The documentation on which the test cases are based. If a document can be amended only by way of formal amendment procedure, then the test basis is called a frozen test basis
- I. Agile Manifesto
- J. Quality Risk
- K. Exploratory Testing
- L. Incremental Development Model
- M. Product Risk
- N. Test Approach
- O. Test Basis
- P. User Story
- Q. Test Charter
- R. Iteration

19. A statement of test objectives, and possibly test ideas about how to test. Used in exploratory testing.
20. The calculated approximation of a result related to various aspects of testing (e.g. effort spent, completion date, costs involved, number of test cases, etc.) which is usable even if input data may be incomplete, uncertain or noisy
21. The use of software, e.g. capture, playback tools, to control the execution of tests, the comparison of actual results, the setting up of test preconditions, and other test control & reporting functions
22. A source to determine expected results to compare with the actual result of the software under test. It may be the existing system (for a benchmark), other software, a user manual, or an individuals specialized knowledge, but it should not be the code
23. A high level description of the test levels to be performed and the testing within those levels for an organization or programmer (one or more projects)
24. A way of developing software where the test cases are developed and often automated, before the software is developed to run those test cases
25. A tool that provides an environment for unit or component testing in which a component can be tested in isolation or with suitable stubs and drivers. It also provides other support for the developer, such as debugging capabilities
26. A high level user or business RQ commonly used in agile software development, typically consisting of one sentence in the everyday or business language capturing what functionality a user needs and the reason behind this, any non-functional criteria, and it also includes the acceptance criteria
- S. Acceptance Criteria
- T. Test Oracle
- U. Configuration Item
- V. Build Verification Test
- W. Regression Testing
- X. Test Automation
- Y. Test Driven Development
- Z. Agile Software Development