USC Marshall Center for Technology Commercialization

Technology Readiness Levels

**Readiness: Software Development**
(Adapted from CECOM’s Software Technology Readiness Levels)

**Level 1: Basic principles observed and reported.**
Lowest level of software readiness. Basic research begins to be translated into applied research and development. Examples may include a concept that can be implemented in software or analytic studies of an algorithm’s basic properties.

**Level 2: Technology concept and/or application formulated.**
Invention begins. Once basic principles are observed, practical applications can be postulated. The application is speculative and there is no proof or detailed analysis to support the assumptions. Examples are still limited to analytical studies.

**Level 3: Analytical and experimental critical function and/or characteristic proof of concept.**
Active research and development is initiated. This included analytical studies to produce code that validates analytical predictions of separate software elements of the technology. Examples include software components that are not yet integrated or representative but satisfy an operational need. Algorithms run on a surrogate processor in a laboratory environment.

**Level 4: Technology component and/or basic technology sub-system validation in laboratory environment.**
Basic software components are integrated to establish that they will work together. They are relatively primitive with regard to efficiency and reliability compared to the eventual system. System software architecture development initiated to include interoperability, reliability, maintainability, extensibility, scalability, and security issues. Software integrated with simulated current/legacy elements as appropriate.

**Level 5: Technology component and/or basic sub-system validation in relevant environment.**
Reliability of software ensemble increases significantly. The basic software components are integrated with reasonably realistic supporting elements so that it can be tested in a simulated environment. Examples include “high fidelity” laboratory integration of software components.
System software architecture established. Algorithms run on a processor(s) with characteristics expected in the operational environment. Software releases are “Alpha versions and configuration control is initiated. Verification, Validation, and Accreditation initiated.

Level 6: Technology system/subsystem model or prototype demonstration in a relevant environment.
Representative model or prototype system, which is well beyond that of level 5, is tested in a relevant environment. Represents a major step up in software demonstrated readiness. Examples include testing a prototype in a live/virtual experiment or in a simulated operational environment. Algorithms run on processor of the operational environment are integrated with actual external entities. Software releases are “Beta” versions and configuration controlled. Software support structure is in development. Verification, Validation and Accreditation is in progress.

Level 7: System prototype demonstration in an operational environment.
Represents a major step up from Level 6, requiring the demonstration of an actual system prototype in an operational environment. Algorithms run on processor of the operational environment are integrated with actual external entities. Software support structure is in place. Software releases are in distinct versions. Frequency and severity of software deficiency reports do not significantly degrade functionality or performance. Verification, Validation and Accreditation completed.

Level 8: Actual system completed and qualified through test and demonstration.
Software has been proven to work in its final form and under expected conditions. In most cases, this level represents the end of true system development. Examples include test and evaluation of the software in its intended system to determine it meets design specifications. Software releases are production versions and configuration controlled, in a secured environment. Software deficiencies are rapidly resolved through support infrastructure.

Level 9: Technology System proven through successful operations.
Application of the software in its final form and under usage conditions, such as those encountered in operational test, evaluation and reliability trials. In almost all case, this is the end of the last “bug fixing” aspects of the system development. Examples include using the system under operational conditions. Software releases are production versions and configuration controlled. Frequency and severity of software deficiencies are at a minimum.