

User Story Done Criteria

1	Overview	2
2	Required Done Criteria	2
2.1	Story Type: Preliminary Design	2
2.2	Story Type: Detailed Design	2
2.3	Story type: CUT	3
2.4	Story type: Integration	3

1 Overview

A user story is deemed complete after it meets its done criteria and its acceptance criteria. This document provides requirements to follow for identifying these criteria.

2 Required Done Criteria

The following sections describe the definition of done for each user story type. All stories of each story type have the same done criteria.

Each section below has the following format:

1. Story Type: Story type name

High level story done criteria

- a. Required artifact

Detailed “done” checklists for each required artifact are available in separate documents.

Some story types do not have common done criteria. Such stories are deemed completed solely by their acceptance criteria. Stories of this type are indicated below by stating the done criterion is N/A.

2.1 Story Type: Preliminary Design.

Software Component (SWCI) architecture reflects the user story's preliminary design. This is completed by showing artifacts that focus on identifying actors, black box software components and their inter-relationships. The architecture is represented by the following artifacts:

- a. Rhapsody Agile Leaf Subsystem artifact
- b. Actors
- c. Subsystem context diagrams
- d. Leaf subsystem level collaborations
- e. Subsystem collaboration requirements trace diagram
- f. Subsystem collaboration context diagram
- g. Subsystem collaboration system component trace diagram
- h. Subsystem level collaborations' sequence diagrams showing Software Component interactions
- i. Software Component Dependency Diagrams
- j. Boundary Software Component Diagrams
- k. Software component interface diagrams
- l. Software component to system component trace diagrams
- m. Variation management has been applied as needed to the model
- n. Software Model Questions are all addressed.
- o. Level 2 (previously EI&T) test plan is developed or updated

2.2 Story Type: Detailed Design

Detailed design refines the Preliminary Design sufficient for implementation in a programming language. This is done by showing artifacts that focus on identifying relationships between a software

component's architecturally significant classes. The internal implementation details of the classes, e.g., function parameters, private functions and data, are omitted unless they are critical to understanding the design. Furthermore, function names in the model can be notional, i.e., they do not need to match the resulting code. Such implementation details will be found in the code and not duplicated in the model. The Detailed Design is represented by the following artifacts:

- a. Component level collaborations.
- b. Subsystem level collaborations links to component level collaborations
- c. Component level collaboration sequence diagrams
- d. Component level collaboration class diagrams
- e. Classes
- f. Design patterns
- g. Software model questions are all addressed

2.3 Story type: Code & Unit Test (CUT)

CUT includes production code and its level 1 tests and test results. Level 1 tests are done using CppUnitLite for C++ and JUnit for Java. Model updated to reflect design impact of code changes. All implementation artifacts from the story are ready for formal review.

- a. All code written will include Code Headers & Commenting standards
- b. New and Modified New product code units compliant
- c. Each new or modified level 1 test is named to clearly indicate its purpose
- d. Code and level 1 test drivers are in the code repository on the appropriate branch
- e. A subject matter expert has peer reviewed level 1 test procedures to verify sufficiency of preconditions tested and expected results verifications
- f. Any errors in code under test's test report have been allowed as exceptions by technical lead.
- g. Any errors in regression tests report resulting from newly written or modified code have been allowed as exceptions by technical lead.
- h. Level 1 test reports include build log data
- i. Location of test reports referenced in a software code review record
- j. When necessary, software design (preliminary and detailed) has been updated in the models
- k. Level 2 test procedures have been created

2.4 Story type: Integration

Verification and regression tests are complete

- a. All open Level 2 open items have been documented in the test record and in additional software code records and stories.
- b. Updated Level 2 test plans and procedures
- c. Software code record is updated with test results and status
- d. Record issues for newly detected problems