

Leading Practice: Test Strategy and Approach in Agile Projects



Abstract

This document provides best practices on how to strategize testing CA Project and Portfolio Management (CA PPM) in an agile project. The document does not include specific test cases; the list of test cases and steps for each test case are provided in a separate document.

This document should be used by the agile project team that is planning the testing activities, and by end users who perform user acceptance testing (UAT).

Concepts

Concept	Description
Test Approach	Defines testing strategy, roles and responsibilities of various team members, and test types.
Testing Environments	Outlines which testing is carried out in which environment.
Testing Automation and Tools	Addresses test management and automation tools required for test execution.
Risk Analysis	Defines the approach for risk identification and plans to mitigate risks as well as a contingency plan.
Test Planning and Execution	Defines the approach to plan the test cases, test scripts, and execution.
Review and Approval	Lists individuals who should review, approve and sign off on test results.

Test Approach

The test approach defines testing strategy, roles and responsibilities of various team members, and the test types.

The first step is to define the testing strategy. It should describe how and when the testing will be conducted, who will do the testing, the type of testing being conducted, features being tested, environment(s) where the testing takes place, what testing tools are used, and how are defects tracked and managed. The testing strategy should be prepared by the agile core team. In an agile project:

- Feature testing occurs throughout each sprint with the final user acceptance testing (UAT) conducted in the testing sprint before going live.
- During the release planning meeting, the team should capture acceptance criteria and immediately add them as logical test cases linked to the product backlog item.
- In each sprint planning meeting, a sprint test plan should be created and reviewed.
- As the testing is done in each sprint, the results should be tracked along with which features have been successfully tested and which ones have defects. These results should be passed back to the CA consultant who can fix the defects.
- Care should be taken to distinguish defects and changes in requirements. If the team requires a change in requirement, this should be added to the product backlog and prioritized.

Test Actors

To efficiently test the various rights assigned to the various security groups, the team should create a mini test organization by following these steps:

- Identify and create the various actors ('test actors') whose CA PPM access rights must be validated. Usually this would represent one actor for each CA PPM security group.
- If a reporting organizational business structure (OBS) is used in the system, create the appropriate reporting relationship for the test actors.
- Validate the CA PPM access rights for each of the security groups by logging in as the test actor and execute each test case.
- When the system goes live, the test actors should be de-activated. Whenever there is a need to conduct use case or system validation, test actors can be re-activated.

In this sample organization, the following actors could be required:

- Project manager or scrum master (PM1, PM2)
- Idea manager
- Project team member (TM1, TM2)
- Resource or capacity manager
- Scrum product owner (presuming the organization uses this role)
- Project stakeholders/executives (persons interested in the project, but is not assigned any specific tasks)
- PMO/IT requestor
- IT Finance team member
- Governance board member
- Project accounting team member

Depending upon the test cases, these actors may need to be set up for each business unit.

Types of Testing

Different types of testing occur at different points within the project implementation lifecycle. Each type of testing serves a different purpose, but all build up to final user acceptance testing before the system goes live.

Unit Testing

Unit tests are conducted to confirm an operational baseline install of a software component or to confirm connectivity to the CA PPM system.

Unit tests are required whenever a software component is installed into an environment. While testing of the installation in the development environment would be adequate to validate the installation procedure prescribed for the solution, any installation into any subsequent environment must be appropriately validated. This testing is conducted by the agile core team members, along with the help of selected technical support staff; for example, single sign-on.

Note: The CA On Demand team performs unit tests on the environment at the time of environment setup, and whenever a scheduled update is performed.

Feature Testing

Feature tests confirm the desired functionality of the delivered solution, and are specifically designed to validate that the solution addresses requirements as closely as possible. A key purpose of testing is to provide timely information about the quality of the system being built, and this testing is conducted by the agile core team members. Acceptance criteria should be clearly defined for each feature.

Feature testing should be spread evenly throughout the sprint. Component tests can start as each feature is configured or built; these are the building blocks for finalizing the acceptance tests towards the end of the sprint, or before go live. Work on each requirement continues until any defects are fixed. Any change in requirements or any enhancements discovered during the testing process should be added to the product backlog and prioritized.

The testing should focus on the configuration of the system and any customization, as well as the processes built for the customer. Testing the out-of-the-box features would not be needed as those have already been validated by CA prior to releasing the product.

Functional Testing

Functional testing confirms the desired functionality of the delivered solution and is commonly based on the use cases defined for the solution. Specifically, functional tests are designed to simulate project management processes used by the customer, as closely as possible. Test cases step through the project management process through a variety of scenarios. For example, some scenarios will take a change from the initial idea to a new project created in CA PPM. Others will simulate the idea being rejected. The test script should prescribe any pre-requisites and a set of steps or actions that each actor will execute during the test. The testers should record the test results, and if the expected results were met.

Acceptance criteria should be clearly defined for each use case, and business process. Functional testing is conducted by the agile core team members.

Integration Testing

Integration testing tests the interfaces between CA PPM and other external systems such as a financial, human resources, or enterprise resource planning systems. Integration tests are required whenever a software component is installed into an environment and integrated with the other software components.

Integration testing should be performed when all the required components have been configured or built, and the required interfaces have been installed. Each interface can be tested individually and test results signed off. This testing is conducted by the agile core team members, and with the functional administrators of interfacing systems.

User Acceptance Testing

This testing is performed by end users before the system can be certified as ready to go live. It is conducted in the last sprint before go live, and by the end users who would use the system after it is live. Users must be given brief training in the requirements, how to use CA PPM, where to find the various features, and how the system has been configured. Acceptance criteria must be clearly defined for the system functions required to go live.

Regression Testing

This testing should be carried out on any systems which are part of the solution, and have been upgraded or patched during the lifetime of the solution. An appropriate regression test would involve appropriate unit, feature and integration testing related to the affected software product, and a specific test or set of tests related to any feature that is impacted by the upgraded or patched software.

The regression testing should not be so extensive that an entire execution of the solution test plan is required. The regression test should only focus on the function specifically provided by the upgraded or patched component of the software product and affected interfaces.

Collecting a good regression test set is important. Some teams re-run every test within each sprint. This is time consuming and, typically is not worth the effort. Having a clear understanding of which tests to execute during regression testing raises the return of investment of the testing effort and provides more time to specify and execute test cases for the functionalities implemented during the current sprint.

Regression testing is conducted by the CA PPM administrator along with the people responsible for the interfacing systems depending upon what systems and components are being tested.

Performance Testing

Performance testing evaluates system performance for speed and stability under load and stress. Tests should be conducted through emulations in a test environment before the full go live to all the end users. Since testing requires use of a separate automated testing tool, organizations should enlist help from their technical staff. For CA PPM, tests can be coordinated with the CA On Demand Support team and your agile project team by opening a ticket on the CA On Demand Support web site.

Production Validation

Validation testing should be carried out after the build has been released to the production environment to ensure the functionality has been deployed correctly. Typically, production validation is executed during off hours using a sample data set in production. Proper and complete production verification requires knowledge of the solution and experience with common build issues. If issues are found, the agile team should determine if the functionality was deployed correctly or whether roll-back procedures must be implemented. Once completed, the test data sets should be removed or deactivated within the environment. Production validation is conducted by the agile core team members, CA PPM administrator and end users.

Training Validation

The purpose of training validation is to help ensure trainees will experience the same behavior/functionality that will be deployed in the production environment. This testing is conducted by the trainer or training administrator. In addition to validating functionality, trainers should ensure the performance of the application is sufficient for the number of users being trained. This testing is conducted by the trainer or training administrator.

Validation of the training environment should be carried out after:

- The CA PPM build has been completed and deployed to the training environment or any other environment where the training will be conducted, such as the test environment
- Training content and training data sets have been developed
- Backup of the training environment has been requested. Note if multiple sessions are to be conducted this provides a systematic image to refresh the environment
- Trainer has been educated on the business processes and the current CA PPM that will be deployed

Testing Environments

The following chart shows which types of testing are conducted in each environment. Whenever the changes are migrated from one environment to another, it is advisable to revalidate testing conducted in the prior environment in the new environment.

	Development Environment	Testing and Training Environments	Production Environment
Unit Testing	Yes		
Feature Testing	Yes		
Functional Testing	Yes		
Integration Testing	Yes		
Regression Testing		Yes	
User Acceptance Testing		Yes	
Production Validation			Yes
Training Validation		Yes	

Test Automation and Testing Tools

Automated testing tools can help to create and link test cases to backlog items. This reduces the possibility for human error during regression testing and iterative tests. As noted above, these types of tools are likely required to conduct performance testing.

The project team should define test management and automation tools required for test execution. If no automated testing tools are available on-site, test cases and results should be tracked separately, such as in a spreadsheet.

Risk Analysis

The agile core team should develop a comprehensive list of all potential risks. These types of risks may include:

- Resource Availability
- Financial
- Technical
- Scope
- Business Process
- Training/User Adoption
- Communication

These risks should be prioritized based on likelihood and potential impact. This information can be used by the agile project team to focus on developing plans to mitigate these risks and to develop contingencies. Just like on any other project, if a risk management methodology is readily available, it should be used.

Test Planning and Execution

The steps involved in planning and executing testing are:

- Use cases/user stories have been clearly defined
- Build test cases and test scripts based on use cases
- Define acceptance criteria for each feature and each process
- Conduct testing
- Review and approve test results
- Migrate changes to production or for the next level of testing (such as user acceptance testing)

Review and Approval

- The agile core team and the project leader should be responsible for validating and approving all test results.
- The functional administrators of the interfacing systems should also sign off on integration test results.
- The end users and their authorized representative(s) should validate and sign off on the User Acceptance Test results and production validation.
- The trainer should validate and sign-off on the training content and training validation.

Authored by CA Services