Functional Test Strategy

An application's functional test plan defines how functional testing will be completed to ensure that input produces expected outputs. The plan is only one part of a complete application testing strategy. Functional test planning should begin as early as during the application design of the application development.

Cynergy Functional Test Strategy
Author: Tony Lombardo
Created on: 05/08/2009
Last Modified on: 05/13/2009

Testing Goals
Team Member Assignments
General Project Timeline
Scope
Testing Approach & Tools
Strategies
Test Plans Identified
Test Case Repository
Requirements & Test Case Design
Defect Tracking System
Executing Test Cases
Test Status Reporting

Testing Goals

The following are the objectives of functional testing.

The Cynergy application is new and will have high visibility on campus. It is critical we take steps to ensure its functional quality. All available components of the Cynergy application are not enabled in the current installation. This phase of functional testing will be limited to components that are in use at this time. A core set of functional tests representing basic application functions will be designed with future regression testing in mind. These features will be tested to ensure all functions provide the expected result. Testing will complete when all identified features have passed all associated test cases. Exceptions may be made for functional requirements that are non-critical.

Team Member Assignments

The following resources will be completely or partially dedicated to the testing effort. The roles each will play in the testing phase.

<table>
<thead>
<tr>
<th>Name</th>
<th>High-level Testing Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aaron Godert</td>
<td>Cynergy Product Owner and Team Lead</td>
</tr>
<tr>
<td>Mark Anbinder</td>
<td>Cynergy Testing Coordinator</td>
</tr>
<tr>
<td>Tony Lombardo</td>
<td>Testing SME</td>
</tr>
</tbody>
</table>

Table 1. Testing Team Member Assignments

<table>
<thead>
<tr>
<th>Name</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>TBD</td>
<td>TBD</td>
</tr>
<tr>
<td>TBD</td>
<td>TBD</td>
</tr>
</tbody>
</table>

General Project Timeline

Beginning roughly - ending roughly

Scope

This section details the features that will be included in the functional testing phase(s) and those that will be excluded.

Detail the features that will be tested in Table 2 and those that will be excluded in Table 3. Define the scope as specifically as desired (i.e., leave vague or list specific requirements).

Table 2. Features Included in Testing
### Feature ID

<table>
<thead>
<tr>
<th>Feature ID</th>
<th>Name / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dashboard</td>
</tr>
<tr>
<td>2</td>
<td>General Purpose Workflow Dashboard</td>
</tr>
<tr>
<td>3</td>
<td>Portable Document Format (PDF)</td>
</tr>
</tbody>
</table>

**Table 3. Features Not Included in Testing**

<table>
<thead>
<tr>
<th>Name / Description</th>
<th>Reason for Exclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Testing</td>
<td>Separate Cynergy team project</td>
</tr>
<tr>
<td>Integration Testing</td>
<td>Separate Cynergy team project</td>
</tr>
<tr>
<td>Usability Testing</td>
<td>Done at the Kuali Foundation level</td>
</tr>
<tr>
<td>Load/Performance Testing</td>
<td>Separate CIT IS project</td>
</tr>
</tbody>
</table>

### Testing Approach & Tools

**The following approach and tools will be used to test the application.**

A decision on which automation tool should be used. If test automation is a future consideration, HP Quality Center is probably best. There may be some licensing issues, as well as competing with other Kuali testing projects. If automated testing is not a future consideration TestLink may provide an adequate test case repository.

Example: Testing will be executed with the aid of HP Quality Center, an automated functional testing tool. Test cases will be developed and maintained in the tool. A template will need to be agreed to for creating test cases.

Section [#Requirements & Test Case Design](#) defines the basic guidelines for test case design.

### Strategies

**Several strategies are employed in the plan in order to manage risk and get maximum value from the time available for test preparation and execution.**

**General Test Objectives:**

1. To find any bugs that have not been found in unit and integration testing performed by development team
2. To ensure that all requirements have been met Positive test cases designed to test for correct functions will be supplemented with negative test cases designed to find problems and to test for correct error and exception handling.

- **Product Quality Standard**
  - **Release Criteria**
    1. Execution of all documented test cases is attempted and their status recorded.
    2. Application will not proceed to a Production system with unresolved software problems of critical severity (defined as severity = 1 or 2) without prior approval.
    3. All non-critical severity problems not resolved in the current release will be reviewed and workarounds identified where possible.

- **Testing Quality Standard**
  - **Goals**
    1. All test cases are clearly documented.
    2. Entrance Criteria, as defined in the Test Plan, will be met before entering a Test Phase.
    3. Exit Criteria, as defined in the Test Plan, will be met before exiting a Test Phase.

- A list of Entrance Criteria to be completed is required before starting a Test Phase. A list of Exit Criteria to be completed is required before exiting a Test Phase. These lists will be detailed in the Test Plan.
  - Entrance criteria allows for the proper preparedness in order to start testing.
  - If Exit criteria is not met for a specific Test Phase, then the software will not proceed to the next Test Phase.
    - The Entrance & Exit Criteria will list the most important items and will not be an exhaustive list of tasks that is usually completed to enter or exit a Test Phase.
    - Entrance & Exit Criteria is usually associated to the Test Phase, but if needed, can also be specific to a particular software migration by either date, or other identifying aspect.

### Test Plans Identified

The following Test Plans will be created for this project.

Cynergy Functional area to be tested
Test Case Repository

- Identify application where the test cases will be electronically stored.
- If applicable, the directory path on the Test Case Repository will be identified in the Test Plan document. Access is granted to team members as appropriate.
- Available applications can be a Word document file, Spreadsheet file or Database application.
- Preference should be placed on an Excel file due to the ability to export all fields into HP Quality Center. (For Business Area's planning to utilize HP Quality Center.)

Requirements & Test Case Design

It is standard testing methodology to create test cases to validate the requirements of the system. The following represents specific methodology that will be utilized for the this project as it pertains to requirements and test case design.

- For each business process requirement, one or more test cases will be created to validate the business process.
- Test cases can also be created to verify software feature requirements.
- In the absence of business process requirements, the 'Test Case Summary' definition will become the requirement.
- Test cases can be created outside the scope of a pre-existing requirement. In these cases, the 'Test Case Summary' will be considered the business process or software requirement.
- A Test Case review can be initiated by either the business area or CIT to help ensure testing coverage & accuracy. This activity is recommended but not currently a mandated activity for this release.

The following will be the minimal requirements to create a test case.

<table>
<thead>
<tr>
<th>Test Case Number</th>
<th>Test Case Summary</th>
<th>Test Case Description</th>
<th>Test Case Steps (1-x)</th>
<th>Test Case Expected Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(One line)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Optional per test case - but highly recommended)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(A minimum of one step indicating user action)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Defect Tracking System

- Identify the system that will be used for defect tracking to submit problems identified during testing.
- Every issue must have an appropriate severity level assigned as determined by both the functional and technical representatives.
- Severity levels of problems can change if there is agreement between the functional and technical representatives, once opened, due to various reasons. If agreement can not be reached, escalation to the project governance structure will be used to reach a decision.
- Severity levels for a particular issue may be downgraded to a lower severity if a workaround has been identified or it is later found to be a lesser problem or enhancement.
- Severity levels can be upgraded in severity due to finding an increase level of importance to resolve the problem in the current release.

The following table below lists guidelines to assist in determining the severity level of a problem that was identified from the execution of a test case.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1=High</td>
<td>System had an outage or significant parts of the system were inoperable as a result of executing the test. There is no workaround. Severity level 1 issues must be addressed before the software exits any Test Phase.</td>
</tr>
<tr>
<td>2=Serious</td>
<td>One of more primary business requirements could not be met with the system. There might be a temporary workaround but no easily apparent or readily acceptable viable workarounds are in place. Performance, functionality, or usability is seriously degraded. At times, a problem is raised to this severity level - from a lower severity - due to the customer's expressed importance to have the issue resolved in the current release. The goal is to have all severity 2 issues resolved before exiting a specific Test Phase but must be resolved before entering Production.</td>
</tr>
<tr>
<td>3=Moderate</td>
<td>Business requirements can be met with the system. Any needed workarounds are apparent. Performance, functionality, or usability is not seriously degraded. Level 3 severity defects may, or may not, be fixed in the current release.</td>
</tr>
<tr>
<td>4=Minor</td>
<td>These issues are very minor in nature, yet still are valid problems. An example would be a typo in the documentation or an irregularity in the user interface. These problems usually do not affect overall release accuracy or usability in any significant way. Level 4 severity defects are less likely to be fixed in the current release and are usually deferred to a later release.</td>
</tr>
</tbody>
</table>
5=Enhancement
Issue is an enhancement and not a bug. Usually this issue addresses items outside of the scope of present business requirements. This is usually fixed in a future software release.

Executing Test Cases

Note whether test cases will be automated, executed manually, or a combination

It is the comparison of Test Case expected results to Test Case actual results (obtained from the test execution run) that will determine whether the test has a 'Pass' or 'Fail' status.

Test Case status definitions are:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passed (P)</td>
<td>Test run-result matches the expected-result.</td>
</tr>
<tr>
<td>Failed (F)</td>
<td>Test run-result did not match expected-result. In some cases, test run-result did match expected-result but caused another problem. A defect must be logged and referenced for all failed test cases.</td>
</tr>
<tr>
<td>Not Run (NR)</td>
<td>Test has not yet been executed. In a test case database, all tests start from a default status of 'NR'.</td>
</tr>
<tr>
<td>In Progress (IP)</td>
<td>Test has been started but not all of the test steps have been completed.</td>
</tr>
<tr>
<td>Investigating (I)</td>
<td>Test has been run but investigating on whether to declare as a passed or failed test. This may be due to the difficulty of the problem investigation or investigation has not yet started on a particular test that has been run. A defect must be logged and referenced when requesting assistance in the investigation.</td>
</tr>
<tr>
<td>Blocked (B)</td>
<td>Test cannot be executed due to a blocking issue. An example would be a particular test cannot run until a hardware problem is resolved. Sometimes it is a toss-up in deciding whether to leave a test in 'Not Run' status or assign it a 'Blocked' status. Comments should be provided indicating the issue causing the block. A defect should be logged and referenced assistance is required to resolve the issue causing the block.</td>
</tr>
<tr>
<td>Deferred (D)</td>
<td>Test has been approved to be deferred and will not be executed in the current release.</td>
</tr>
</tbody>
</table>

Test Status Reporting

Identify reporting and communications plan

Example:
Functional Area Testing Status Report - this is a report of testing activities from a specific functional area to the project manager. This report is created by each Functional Team Lead.

Functional Area Testing Status Report
The following will be the minimal requirements of the Test Report are:

1. Current Testing Phase
   a. Listing of Test Suites Planned to be Executed During This Phase
      i. Per Test Suite planned in this phase,
      ii. Total count of Test Cases in the Test Suite
      iii. Total count of Test Cases Not Run
      iv. Total count of Test Cases Run
      v. Total count of Test Cases Passed
      vi. Total count of Test Cases Failed
      vii. Total count of Test Cases Investigating
      viii. Total count of Test Cases Blocked
      ix. Total count of Test Cases Deferred

- For all Test Cases in Failed state, provide total count of problems in each of the severity levels (1-5).
- For all Test Cases in Failed state, provide a description of each open problem with a severity level of 1 or 2 and current status.
- For all Test Cases in Investigating, Blocked, or Deferred, provide comments (indicating RT ticket, if applicable, reason for block or reason for deferment).