

---

# ABT Program Testing Strategy Plan

---

June, 2008

## Revision History

<b>Version</b>	<b>Prepared by</b>	<b>Date</b>	<b>Reviewed by</b>	<b>Approved by</b>	<b>Note</b>
<b>Original v.1</b>	<b>Keith Pace- CIBER</b>	<b>May 27, 2008</b>	<b>Lisa Gauthier</b>	<b>Marjorie Mills</b>	
<b>v.2</b>	<b>Lisa Gauthier</b>	<b>June 30, 2008</b>	<b>Marjorie Mills</b>	<b>Manny Ovena, Keith Pace- CIBER</b>	<b>OCM and other changes</b>

# Table of Contents

1.0 Introduction.....	4
2.0 Methodology Overview .....	4
2.1 Testing Levels and Types .....	6
2.1.1 Test Level .....	6
2.1.2 Test Types.....	8
2.2 Testing Approach .....	11
2.3 Testing Tools and Technology .....	16
2.4 Resources, Roles and Responsibilities.....	18
2.5 Testing Environments .....	20
2.6 Instance/Configuration Management .....	22
2.7 Test Issue/Defect/Problem Tracking and Resolution.....	22
2.8 Test Issue/Defect/Problem Identification and Submission.....	23
2.9 Test Issue Process, Escalation and Resolution .....	24
2.10 Assumptions, Risks and Schedule.....	26
2.10.1 Assumptions .....	26
2.10.2 Risks.....	26
2.10.3 Mitigations .....	27
2.10.4 Testing Plan, Schedule and Resources.....	27

## 1.0 Introduction

---

The purpose of this document is to describe the testing strategy designed for the ABT implementation/migration project, and the methods, tools and resource roles required to achieve the desired testing results.

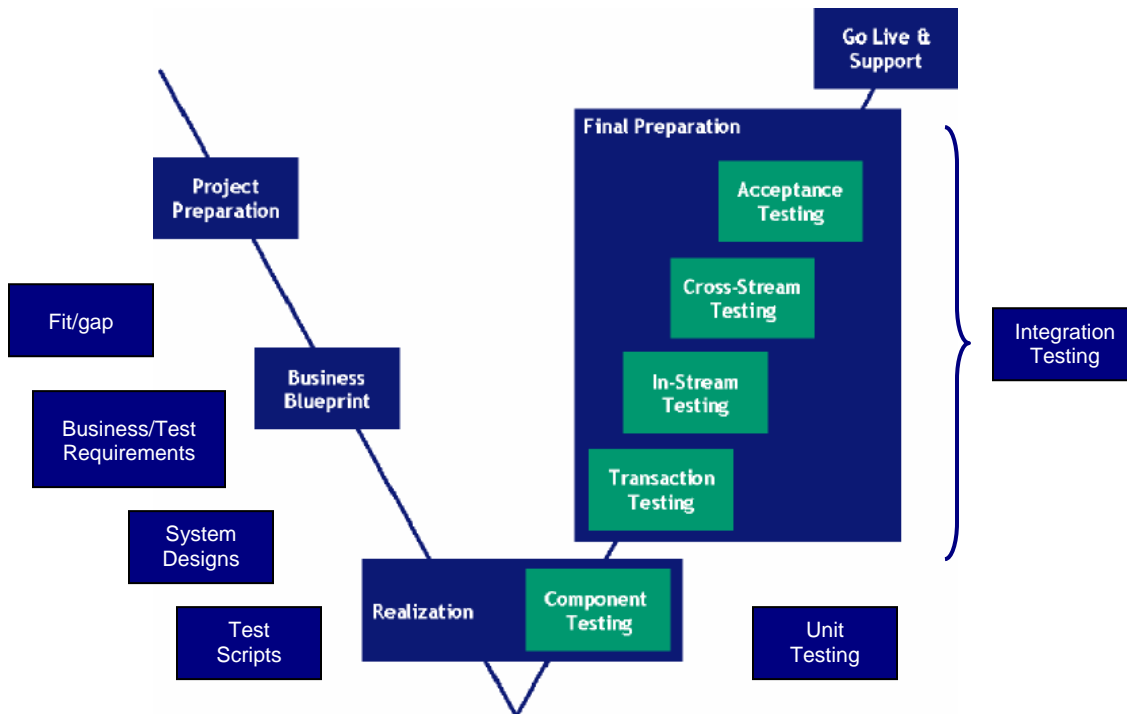
Testing is a crucial element of the ABT implementation as the outcomes become the specific verification and validation of the new Oracle EBS for financials, PeopleSoft Human Capital Management (HCM) for human resources (HR) and payroll and budget system's ability to meet defined business requirements, successfully utilize future-state business processes and ensure system readiness. Moreover, testing supports the knowledge transfer, training and preparation of county users on using and supporting the new systems and processes.

Most of planned testing is functional in nature (i.e. business process related), with the purpose of validating business requirements and confirming business readiness. However, we also plan to conduct technical tests, like system performance tests, to validate system requirements and to confirm system readiness.

## 2.0 Methodology Overview

---

CIBER has developed a specific approach to manage the testing process during the ABT implementation. We plan for each major ABT project phase to follow the general implementation/testing life-cycle (see figure below). Requirements fit/gap and design activities—leading to the development of test scripts (left side of below figure)—are conducted once for the entire King County organization, while the testing lifecycle (right side of below figure) is repeated for each planned phase/transition cycle.



The implementation/testing life-cycle takes this general course:

- Project preparation (initiation) identifies the scope, approach, objectives and related project control features;
- Fit/gap analysis defines business requirements, business processes and conceptual system designs (leading to a system blueprint); alternative designs are reviewed, modeled and refined to arrive at a base-line prototype in order to move into the testing phase (realization phase). Business requirements, test requirements and custom development objects (conversions, interfaces, reports and modifications) are defined, designed, documented, reviewed and approved prior to proceeding to the next phase of testing. This establishes a design baseline—or a testing foundation—prior to the start of testing in order to optimize the testing process and Program Team’s work effort and time. Moreover, this establishes the testing objectives, or acceptance criteria, to ensure a proper definition and measure of testing success.
- Testing begins with component or unit testing, where individual modules are tested in a stand-alone environment; moving progressively through broader and deeper integration testing to fully exercise and test all system modules and components; followed by the final user test to validate system and business readiness.

- System acceptance and approval for production deployment is based on the successful achievement of testing objectives and acceptance criteria. The formal go-live decision will be handled by the ABT PMO; through the standard ABT governance and approval process.

## **2.1 Testing Levels and Types**

### **2.1.1 Test Level**

**Unit (Component) Tests** are individual, stand-alone functional or technical tests of low-level components (e.g. entering an invoice or validating data conversions). Unit tests will validate that the individual pieces of the software functionality as a standalone unit, that the module configuration is sound and free of defects, the business process performs as expected and the user and data security is as designed.

Unit testing is the initial test of configurations, custom code and process models with a requirement of meeting strict exit criteria—in order to move from unit testing into integration testing—to prevent as many defects as possible and to ensure quality in integration testing.

Unit testing ranges from tests of individual functions to testing stand-alone modules (e.g. PeopleSoft Payroll). For the purposes of this test strategy plan, unit testing includes functional testing.

**Integration Tests** are complete functional and technical tests of the system as a whole (e.g. procure to pay process including electronic submission of purchase order and electronic payment of vendor invoice).

**Entry and Exit Criteria** – ABT Program Team will establish formal entrance and exit criteria for integration testing during the initial stage of the ABT implementation. Established entrance criteria must be met in order for an integration test cycle to formally begin: e.g. configuration and designs reviewed and approved; unit tests completed and reviewed; critical issues/defects resolved and non-critical issues/defects addressed; test requirements and test scenarios documented; environments built and validated; necessary code migrated; data staged and/or converted (HR sensitive data handling) and validated; and test plan/schedule baselined and approved. Exit criteria must be met in order for an integration test cycle to formally conclude: completed tests script execution including ABT PMO and/or peer review of actual results; issues/defects documented; critical issues/defects resolved; non-critical issues/defects addressed; and test plan/schedule fulfilled.

Note: Entrance and exit criteria apply to ABT Program, county organization and 3<sup>rd</sup> party inputs and outputs (e.g. an interface between ABT and county organization or external vendor).

There are five basic levels of integration testing: Transaction, In-stream, Cross-stream, Regression and User Testing.

- **Transaction Testing** is the testing of a single transaction (e.g. Create Sales Order) to confirm the operation of transactions and related configuration is sound. There are “Positive” and “Negative” transaction tests. Positive testing is testing that attempts to show that a given transaction/module does what it is supposed to do (e.g. enter an invoice for \$100 and a resulting revenue entry to my cost center is posted for \$100). Negative testing is testing that attempts to show that the transaction/module does not do anything that it is not supposed to do (enter an invoice that posts revenue to my cost center only, will not to post to other’s cost centers). Negative/positive testing is used to test the designed user/data security and access.
- **In-Stream Testing** covers chains of transactions that flow together and which reflect important business process, scenarios and user security/access (e.g. within the Procure to Pay process).
- **Cross-Stream Testing** is end-to-end testing of integrated processes through execution of business processes (within the system and with third-party and legacy systems).
- **Regression Testing** is similar to cross-stream testing, but is focused on testing software functionality for unintended consequence resulting from off-cycle program changes (e.g. major configuration/code change or software patch application that requires testing/validation of software that previously worked as desired, but may have stopped working or no longer work in the same way that was previously tested). Also, regression testing may be used for other unplanned ABT implementation activities that are not in the normal, planned ABT testing cycles.
- **User Tests** are in-stream or cross-stream tests by the user community with the objective of formal validation of the system design, data conversion and business readiness. This is the final test prior to go-live. Throughout each phase of the implementation/ testing life-cycle, ABT stakeholders will approve system tests.

Integration tests will validate that the software operates as designed, across all business processes, including modifications, data conversions, interfaces, workflows, security roles and reports. Ciber utilizes a structured approach to testing, based on the defined test scenarios developed in the fit/gap and business process design activities, using the structural decomposition of high-level business process, sub processes and activities. This

will allow tests to build upon each other to validate the desired system-level behavior. Therefore, integration tests are more than testing PeopleSoft HCM or Oracle EBS software, we are testing business processes.

## 2.1.2 Test Types

Within the broader levels of unit or integration testing, there are various types of testing: user testing, system testing, load testing and parallel testing. Each has a unique design and purpose and will be planned to be strategically positioned in the testing life-cycle to ensure thorough testing results.

Note: CIBER plans to complete all testing types described below during the ABT implementation, with testing plan details to be developed in early stage of ABT project.

- **User Testing** – The primary purpose of user testing is to provide users the opportunity to test the software to ensure it meets the defined business requirements. Secondly, early user testing helps to develop test scenarios/scripts and to facilitate knowledge transfer from the System Integrator (SI) Application Specialist to King County functional team leads.

User testing begins early in the project where ABT and System Integrator team leads work together to design solutions and prototype system configuration by modeling and informally testing various solution alternatives in a development environment. Once designs have been finalized and the test prototype has been approved and configured, user testing becomes more formal as testing progresses through the project's testing life-cycle. User testing concludes with a final user test that formally tests and accepts the system prior to go-live.

**System Testing** – The primary purpose of system testing is to ensure the system as a whole (i.e. servers, security, applications, custom applications, workflows, data and interfaces) works together as designed and meets the defined business requirements and processes. ABT stakeholders responsible for the various components of the system must be involved in system tests in order to validate the system and data.

System testing becomes more comprehensive, more controlled (with lower tolerances for defects) and broader in scope and user population as it progresses through the testing life-cycle. We plan for the following general approach to system test progression during the ABT implementation (plan will be updated during ABT test planning phase):

- **Test Cycle 1** – Testing performed by ABT Program Team only. Test Oracle EBS, PeopleSoft HR/Payroll and Budget applications with limited (70%) of



data conversion, interfaces, modifications and custom reports, on development hardware. Expectation is to successfully pass 80% of test scripts with less than five critical issues remaining.

- **Test Cycle 2** – Testing performed by ABT Program Team and county users. Test same as prior test cycle with additional (85%) data conversions, interfaces, modifications and custom reports, on development hardware. Expectation is to successfully pass 85% of test scripts with two or less critical issues remaining.
- **Test Cycle 3** – Testing performed by ABT Program Team and county users. Test same as prior cycle with all (100%) data conversions, interfaces, modifications and custom reports, on production hardware. Expectation is to successfully pass 95% of test scripts with zero critical issues remaining.
- **Test Cycle 4** – Testing performed by ABT Program Team and county users. Test same as prior cycle with broader county perspective. Also, test the system build process as a dress rehearsal of the production cut-over on production hardware. Expectation is to successfully pass 99% of test scripts with zero critical issues remaining.
- **User Testing** – Testing performed by county users. Test the same as prior test cycle in an abbreviated test cycle, conducted by users for the purpose of user validation and ABT acceptance prior to go-live. Expectation is to successfully pass 99% of test scripts with zero critical issues remaining.
- **Load Testing** – The purpose of load testing is to ensure the system performs adequately and as expected under a heavy user load (e.g. simulating peak load of users logged into the system simultaneously). This test validates peak user load and load balancing, e.g. hundreds of employees simultaneously logging into the self-service PeopleSoft system during open enrollment. Load testing will be performed at least once prior to major go-live. *Load testing may require third<sup>d</sup> party application and/or services.*
- **Stress Testing** – The purpose of stress testing is to ensure the system performs under heavy stress and over extended periods of time (e.g. simulating multiple or very large system processes running long and simultaneously). This test validates the application's performance, stability and reliability under peak load of system processes (and users), e.g. month-end processing. Stress testing will be performed at least once prior to major go-live. *Stress testing may require third party application and/or services.*
- **Parallel Testing** – The purpose of parallel testing is to compare results from the new system to results from the existing (old) system in a side-by-side test with the same inputs and processes. A parallel test requires that data be entered in both systems.

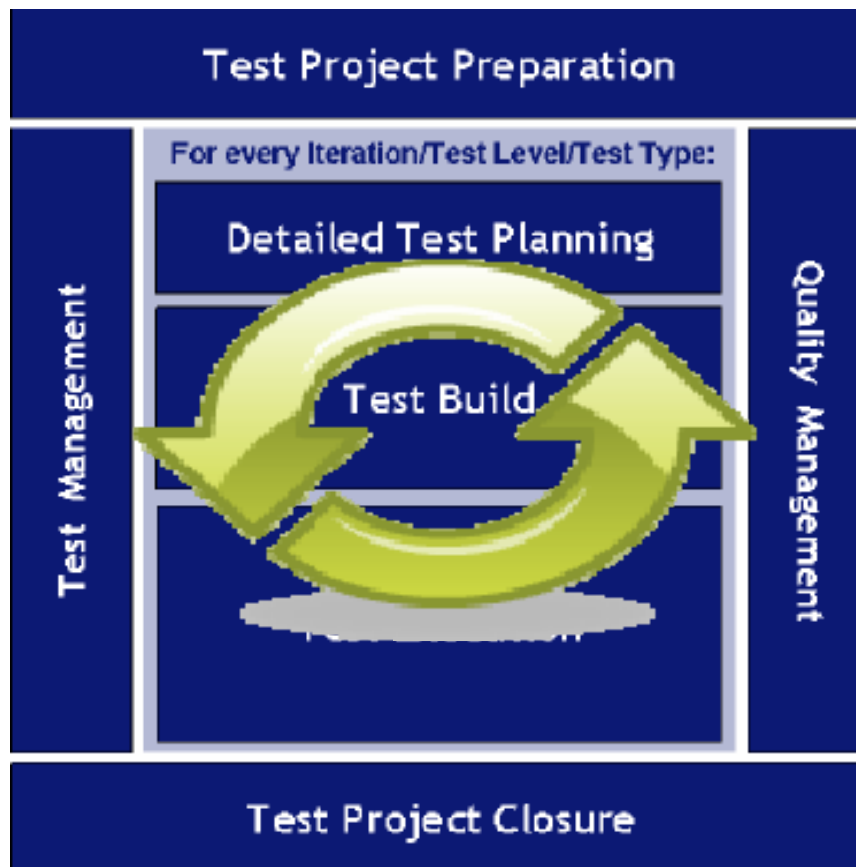
Often times parallel tests use different chart of account code structures and results will require mapping to these different structures for validation purposes. We expect to run three parallel tests at King County for payroll testing of all employees' paychecks included in the current deployment (to compare paychecks between old and new systems). Parallel testing requires real payroll data to run in a true test. Test results variances will be allowed under the condition of known acceptable differences that and have been communicated.

- **Security Testing** – The purpose of security testing is to validate data protection and planned functionality is used as intended. The seven basic security concepts that need to be covered by security testing are: confidentiality, integrity, authentication, authorization, availability, non-repudiation and intrusion. Security testing will be scheduled at least once for each major release of the core enterprise systems.
  - Confidentiality: A security measure which protects against the disclosure of information to parties other than the intended recipient that is by no means the only way of ensuring confidentiality.
  - Integrity: A measure intended to allow the receiver to determine that the information which it receives has not been altered in transit or by other than the originator of the information. Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.
  - Authentication: A measure designed to establish the validity of a transmission, message, or originator. Allows a receiver to have confidence that information it receives originated from a specific known source.
  - Authorization: The process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.
  - Availability: Assuring information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.
  - Non-repudiation: A measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp.
  - Intrusion/Penetration: Assuring unintended or malicious system access. Intrusion testing will follow King County standard procedures.

## 2.2 Testing Approach

Summary of testing types utilized in the testing approach: Unit Testing → System (Integration) Testing → Load/Stress Testing and Security Testing.

This section describes the testing approach (life-cycle) recommended for the ABT implementation.



**Test Project Preparation** - We will prepare for testing in the early stages of the ABT implementation in order to build, conduct, execute and assess valid and effective tests. ABT Program Team will establish test requirements in advance to ensure adequate time and effort are spent on building a stable testing infrastructure.

Test preparation activities will result in:

- Infrastructure (servers, disk storage, network, applications, system environments/instances, database and user access).

- Test Tools (test software, email, SharePoint sites for document storage, etc. See section on Testing Tools and Technology in this document for more details.)
- Test Facilities (e.g. conference room(s) where ABT Program Team members can co-locate to perform testing).

**Detailed Test Planning** – We will plan testing activities, inputs, processes and outputs in the early stages of the ABT implementation to ensure a comprehensive and structured test plan. Testing is such a critical element for user involvement, knowledge transfer, requirements/process validation, business readiness and user validation that all activities and aspects of testing will be fully planned and communicated to the ABT stakeholder community.

Two key planning activities are: 1) the design review and approval process, and 2) the development of test scenarios and scripts.

1. **Design Review and Approval.** Business requirements, future-state processes, configurations and custom code designs must be completed, reviewed and approved, as a planned activity, prior to the start of a planned test cycle in order to ensure quality and to optimize performance by reducing defects and re-work. The ABT Project Manager is responsible for design reviews/approvals, through peer-review, technical review or other review mechanism, e.g. conference room pilot-like code reviews.
2. **Test Scenarios and Scripts.** Test scenarios/scripts result from the collection of business requirements and business processes developed during the fit/gap and business process re-design activities. For example, the fit/gap and/or business process re-design activities will result in information about the business process of “Entering a Purchase Requisition in Oracle iProcurement”. This becomes a numbered test scenario with the same title. This test scenario will have a dozen or so executable steps and will be scripted in a document or test tool to officially become a “test script”, which can be combined with all other test scripts and utilized and repeated through multiple test cycles. A test script includes the input variables (e.g. entered and configuration data: supplier name, product #, production name, product category, amount, price, etc.), the business processing variables (negative/positive testing, required approvals, variable conditions (on-hand/not on-hand), etc.) and the expected results (e.g. purchase order is generated with expected vendor, product & quantity). Testers will follow the test scripts while conducting tests and will record actual test results and/or issues or defects uncovered while testing. The ABT Test Manager is responsible to ensure a complete set of test scripts that cover all King County business requirements and processes.

Note: This test strategy document assumes the creation of manual test scripts (documented test steps). However, in the event King County purchases and uses an

automated test tool (e.g. HP Quality Center), test scripts will be developed in the test tool by testing specialists for automated scripting based on test requirements, designs and delivered code.

Test planning activities will result in:

- Detailed Test and Data Requirements (and supporting documentation)
- Detailed Test Plan (with detailed entrance and exit criteria for ABT, county and third parties)
- Test Documentation and Deliverables
- Communication Plan
- Test Schedule and Milestones
- Named Resource Requirements by Team and Roles & Responsibilities
- Test Environment/Instance Plan and Physical Test Environment
- Test Scenarios (test cases) and Test Scripts (documents)
- Design Review/Approval and Test Acceptance Criteria

**Test Build** – We will build and configure the test environment once business requirements, processes and system designs are sufficiently complete (meet test entrance criteria). The build activity creates the physical environment for testing, including the necessary configurations, data, custom objects (interfaces, modifications, reports and workflows), and establishes the materials required to manage the test, e.g. a new and comprehensive set of test scripts stored in a fresh network directory (or within a test tool) for test execution.

Test build activities will result in:

- System (prototype) Configurations
- Converted Data
- System Interfaces
- Workflows and Other Custom Objects
- Reports (standard and custom)
- Testing and Tracking Tools (complete with test plan, schedule and scripts)

**Test Execution** – This is where we do the heavy lifting. Having a fresh test environment prepared, a fresh set of test scripts, a test plan and schedule, appropriate team members and conference room(s) booked, and test tools prepared, we can begin testing. The ABT PMO will assign responsibility for the role of “Test Manager”, who will organize, facilitate and

manage the testing activities according to the test plan. Testing is essentially the structured execution of defined test scripts and resolution of test issues.

Test Execution activities will result in:

- Completed Test Scripts/Scenarios (documenting expected and actual results)
- Validated Data (note: sensitive HR data will be validated in a secure environment and be modified in normal test environments to protect employee's privacy)
- Validated System Configurations
- Validated System Interfaces
- Validated Workflows and Other Custom Objects
- Validated Reports (standard and custom)
- Validated System Performance (stress and load tests)
- Third Party Applications (as applicable)
- Validated Security
- Design/Process Updates
- Open Test Issues and Defects (by priority and type; requiring resolution)

It is important to note that our test approach plans for 3 to 4 iterations (cycles) of testing. This means the Detailed Test Planning, Test Build and Test Execution activities will occur multiple times, in cycles, allowing for refinement and improvement on prior cycles, and allowing multiple iterations to practice or exercise the necessary steps to prepare for the final production cycle.

**Test Management** – We will manage the testing process with discipline and rigor as this activity is one of the most crucial elements of the entire ABT implementation. As mentioned, the ABT PMO will assign the full-time responsibility of “Test Manager” to an individual to manage the test plan, monitor the test process and ensure the validity and quality of test results.

Test Management activities will manage:

- Test Requirements, Scripts and Acceptance Criteria
- Test Schedules (by process group, organization, test scenario, location, etc)
- Resource Management (the right people are in the right place at right time)

- Test Logistics (test room, PCs, projectors, test materials, etc)
- Monitor, Manage and Communicate Test Performance, Progress and Status
- Track and Manage Test Issues/Defects (with established triage process based on severity/priority)
- Corrective Action (as necessary...for example, where we determine that an integration test has been compromised due to a significant software defect, custom object defect or data incompatibility, the test may be rolled-back to a prior version or backup and re-started)

**Quality Management** – As final user validation depends on the quality of the test results, quality is something we plan and manage into the test cycle. Quality is not something we *hope* will happen, but is something we *plan* to happen. Thus, a key responsibility of the entire ABT Program Team is to ensure quality testing and test results.

Quality Management pre-determines testing objectives, acceptance criteria, metrics and required activities (by owner and dates) to review and measure actual testing results/deliverables against planned results/deliverables. This process will ensure the testing results meet planned expectations for quality and completeness.

Quality management activities will provide oversight in:

- Business Requirements, Test Requirements and Processes
- Test Scripts and Scenarios with “Expected Results” (prior to start of testing and stored in central location for the purpose of managing tests)
- Review and Approve Custom Code and System Designs
- Validate Completed Tests (audit completed test scripts to ensure actual results match the expected results and that tests meet quality specifications for completeness, depth, breadth and accuracy)
- Assist with Tracking, Managing and Reporting Test Issues/Defects (per test plan)

**Test Project Closure** – Given that testing in an iterative process, we need to close each test cycle by ensuring test results and lessons learned are understood and documented in order for this valuable information to be used in future test cycles.

Test Project Closure activities will result in:

- Completed Test Documentation and Test Summary at the conclusion of each test cycle for ABT PMO/PM, program team and key stakeholders (stored in project repository)

- Lessons Learned
- Updates to System Designs, Business Processes, System Configurations and Other Key Project Documentation

## ***2.3 Testing Tools and Technology***

Testing is made up of a complex set of data and activities that must be carefully planned, executed, managed. Therefore, the use of technology and tools can be useful in managing the testing process. There are automated test tools available for purchase and use, like HP Quality Center (aka Mercury Interactive). There are also common tools available for use, like SharePoint, Excel and standard Helpdesk software for issue/defect tracking (Remedy/BMC, HEAT, Computer Associates and others).

- HP Quality Center (Mercury) is an industry standard for system testing, with cross-team functionality and collaboration, and has multiple testing products. Some of the HP Quality Center tools are:
  - WinRunner/QuickTest software is used to automate testing (i.e. to create automatic test scripts and run test scripts automatically without or with less user involvement).
  - TestDirector software is used to gather requirements, plan & schedule tests, analyze results, manage test issues/defects and support cross-team communication and collaboration.
  - Performance Validation software and services are used to perform professional assisted load/stress testing.

Note: HP Quality Center requires costs for licenses, hardware and professional services for installation, configuration, training and support.

- Conventional Tools:
  - PSSD's RFW or HEAT applications that currently exists at King County.
  - VSS application (need to consider cost of adding licenses)
  - Remedy/BMC, Computer Associates and other software vendors develop standard helpdesk type issue tracking tools that may be utilized in tracking test issues, but may be cumbersome in coordinating the use between the regular helpdesk and ABT Program Team.
  - Shareware software, like Bugzilla, can be downloaded for the specific use of issue/defect tracking under the guidelines of the shareware with no vendor support. Further evaluation is required to determine suitability and costs.



- SharePoint software can be easily adapted to develop a repository for test requirements and scripts, scheduling test activities, capturing/tracking fundamental issues/defects and integrating with common business applications and email systems.

During the Detailed Implementation Plan (DIP) analysis, we determined the use of HP Quality Center is an attractive alternative, but likely to be cost prohibitive as the potential return on investment is likely to be years beyond the end of ABT. Therefore, this test strategy assumes the use of more common tool sets that can satisfy the same results with more manual involvement, and less cost.

Core test tool requirements:

- Central repository to store testing artifacts with secure access for ABT Program Team (requirements documents, test plans, test scripts, contact lists, etc.).
- Web enabled issue/defect tracking tool.
- Web enabled calendar and scheduling tool with email functionality.

CIBER recommends the use of our proprietary software testing solution, using SharePoint technology, called PMRx to manage the testing process. The use of PMRx is free to ABT as a client of CIBER. Alternatively, if King County decides against the use of PMRx, other software with similar features and functionality is acceptable.

PMRx is a Web-based project/test management tool used to keep King County and the ABT Program Team informed on the latest project and testing information and status. PMRx is particularly useful on projects with multiple phases such as the ABT implementation. PMRx can be setup as a permanent or interim solution depending on ABT's long-term testing strategy.

Highlights of features delivered with PMRx include:

- **Project Information Areas**—Stores regular Program Team documentation like team listing, project plans, test scripts, status reports, schedules, testing events, etc.
- **Documentation Warehousing**—Allows access to ABT Program documents in a central repository.
- **Security**—Security logons are setup for access to the FTP processes and accessing tools areas. Security can be setup at any level desired, and is part of the initial setup process.

- **InSession Scheduling Tool**—permits the Program Team to schedule testing sessions and identify objectives/attendees, etc. InSession has search, reporting, drilldown, email capabilities and is web enabled.
- **FastTrack Issues Tracking Tool**—allows the team to log and track the progress of testing issues, including reporting and email capabilities.
- **IMPACT Discovery Tool**—is a Web-enabled tool allows clients and consultants to log and track the progress of program or testing requirements, test scripts or test scenarios.

Note, CIBER has recommended PMRx as the testing tool, because it is free of charge and we are familiar with its use and benefits. However, if King County chooses another tool, we can simply replace the term “PMRx” with the name of the other tool that has similar and acceptable features and functionality. Moreover, other third party testing tools or services may be evaluated and procured for integration and/or load/stress testing.

## 2.4 Resources, Roles and Responsibilities

The Testing Team’s main responsibility is to ensure that all functional and technical requirements have been successfully tested and validated at each process, sub-process and activity level (including modifications, data, interfaces, security roles, workflows, and reports).

The following table describes the resource and role recommendations:

Role / Resource	Responsibilities
King County Business Process Owners  <i>(1 person per major process group, e.g. Procure to Pay, Time &amp; Labor, etc.)</i>	<ul style="list-style-type: none"> <li>• Responsible to approve test scenarios, test cases and acceptance criteria</li> <li>• Participate in discussions regarding King County test planning and scheduling</li> </ul>
ABT Program Manager	<ul style="list-style-type: none"> <li>• Ensure testing meets the ABT Program objectives</li> <li>• Manage issue escalation process</li> <li>• Monitor progress and take corrective action as necessary</li> </ul>
ABT Project Manager <i>(2 people: 1 for EBS and 1 for PeopleSoft)</i>  or ABT Test Manager	<ul style="list-style-type: none"> <li>• Manage, coordinate and communicate testing activities (with primary responsibilities for King County user testing)</li> <li>• Review and approve configuration and code designs</li> <li>• Organize and schedule testing for King County resources</li> <li>• Manage issues and defects; escalate issues as required</li> </ul>

Role / Resource	Responsibilities
	<ul style="list-style-type: none"> <li>• Ensure effective test session objectives are established and followed</li> <li>• Facilitate testing sessions</li> <li>• Approves configuration and code migration and modifications and patch application</li> <li>• Collect performance data and report testing status and results from Test Manager</li> </ul>
ABT Organizational Change Manager	<ul style="list-style-type: none"> <li>• Coordinate with Test Manager and monitor issues log to understand, assess and manage implementation and organizational change impacts</li> </ul>
<p>King County Functional Team Leads</p> <p><i>(1 person per module, e.g. Payroll, General Ledger, etc.)</i></p> <p><i>(50% dedicated to testing)</i></p>	<ul style="list-style-type: none"> <li>• Define business and test requirements</li> <li>• Provide details of King County business requirements and processes into the development of test scripts, scenarios and cases</li> <li>• Manage and/or resolve test issues &amp; defects</li> <li>• Support the analysis and development of test scripts, scenarios and cases</li> <li>• Primary participation in functional testing and issue/defect resolution</li> </ul>
<p>King County Functional Users</p> <p><i>(as many as required to cover all business requirements/processes)</i></p> <p><i>(20-40% dedicated to testing)</i></p>	<ul style="list-style-type: none"> <li>• Provide details of King County business requirements and processes into the development of test scripts, scenarios and cases</li> <li>• Support the analysis for developing test scripts, scenarios and cases</li> <li>• Participate in functional testing and issue/defect resolution as necessary</li> </ul>
King County Departments	<ul style="list-style-type: none"> <li>• Work with ABT team to define business and test requirements, data and interface specifications for interfaces to/from ABT environments</li> </ul>
ABT Technical Leads	<ul style="list-style-type: none"> <li>• Support testing activities as required where technical (infrastructure, database, data conversions, interfaces, reports or other custom development) is concerned</li> </ul>
<p>System Integrator Project Manager (<b>Test Manager</b>)</p> <p><i>(1 person)</i></p>	<ul style="list-style-type: none"> <li>• <b>Primary responsibilities for testing scope and activities, quality, results and overall success</b></li> <li>• Organize and schedule testing</li> <li>• Review and approve configuration and code designs</li> <li>• Manage the creation and achievement of test acceptance criteria</li> <li>• Ensure effective test session objectives are established and followed</li> <li>• Manage test issues and defects; escalate issues as required</li> <li>• Facilitate testing sessions</li> <li>• Communicate testing session results and solicit</li> </ul>

Role / Resource	Responsibilities
	<ul style="list-style-type: none"> <li>feedback</li> <li>• Responsible for SI's scope and test summaries</li> </ul>
System Integrator Organizational Change Manager	<ul style="list-style-type: none"> <li>• Coordinate with Test Manager and monitor issues log to understand, assess and manage implementation and organizational change impacts</li> </ul>
System Integrator Functional Leads <i>(1 person per module, e.g. Payroll, General Ledger, etc.)</i>  <i>(90% dedicated to testing)</i>	<ul style="list-style-type: none"> <li>• Define business and test requirements</li> <li>• Provide details application functionality into the development of test scripts, scenarios and cases</li> <li>• Manage and/or resolve test issues &amp; defects</li> <li>• Support the analysis and development of test scripts, scenarios and cases</li> <li>• Primary participation in functional testing and issue/defect resolution</li> </ul>
System Integrator Development Leads <i>(as required to cover all development activities)</i>	<ul style="list-style-type: none"> <li>• Support testing activities as required where technical (data conversions, interfaces, reports or other custom development) is concerned</li> <li>• Provide input to and review test scripts</li> </ul>

Notes:

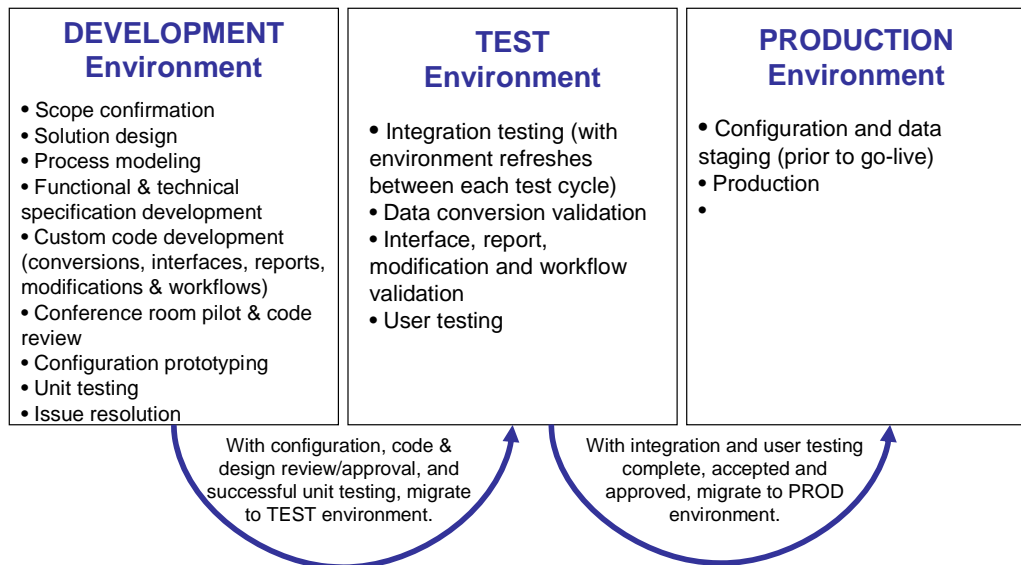
- 1) The System Integrator's Project Managers (aka "Test Manager") has primary responsibilities for all testing activities, with the ABT Project Manager, or ABT Test Manager assisting. However, the ABT Project Managers have primary responsibility for final user testing, where King County must formally accept the system and approve the go-live, with the SI Project Managers assisting.
- 2) ABT Program Team will develop a detailed resource matrix in the test planning phase for named resources by phase, timeframe, role, department and responsibility.

## 2.5 Testing Environments

The testing will be performed in separate environments and servers across test cycles to ensure the separation, security and integrity of testing and the refinement of test execution and system configuration over time. CIBER recommends a standard three environment model where tests progressively migrate through the environments depending on the stage (level and type) of testing being performed.

## Sample Test Environments:

# Test Environment Strategy



## Sample Test Flow:

- 1) **Development Environment** – Component/Unit tests for custom development and system configuration prototyping are performed in the Development environment. Testing in the Development environment is loosely controlled. Once designs, configuration, custom development and component/unit testing have been adequately modeled, tested and reviewed, we move to the next level/environment. (Specific and detailed migration criteria will be established during the ABT implementation.) We move to the next level of testing (by moving to a new test cycle or a new test environment within a test cycle) by copying a clean “staging” environment to a new environment, which reduces the build/setup time, reduces human error in re-keying setups/data and allows prior validation of setups/data to be used in future test cycles and environments.
- 2) **Test Environment** – Integration tests are performed in the Test environment. Testing in the Test environment is more tightly managed and controlled. Objects and configurations being introduced or migrated to Test environment are under strict configuration and change control. Configurations and custom code versions will be documented at the outset of a test cycle, and tracked and updated throughout the

test cycle to document the specific configuration and code versions utilized during testing. Note: the resolution of critical issues and/or defects discovered during integration testing—in the Test environment—may be moved back to the Development environment if required to ensure the integrity of the Test environment. Once integration testing is complete, based on the specific test criteria, we migrate objects (e.g. configurations and custom objects) to the production or staging environment to prepare for the next cycle of testing (or ultimately to production cut-over).

- 3) Production/Staging Environment – Validated and approved configurations are staged in (configured/migrated) the Production environment in order to reduce manual setup steps and/or other preparations in future test cycles, and becomes the basis for future test cycles and ultimately for the production cut-over. This environment is tightly controlled.

## ***2.6 Instance/Configuration Management***

ABT Program Team will develop (early in the ABT implementation) and manage a configuration management and instance management plan to control the configuration, code migration and patch application of the development, test, training (UPK) and production staging environments. This will control configuration/code versioning and ensure stability and the roll-back functions as required by the dynamic nature of testing, resolving issues and fixing problems during a normal test cycle.

## ***2.7 Test Issue/Defect/Problem Tracking and Resolution***

Whereas issues, defects and problems are expected during the test cycle, issue management becomes an important aspect of test management and is fundamental to the success of the test cycle and the overall project. Issue resolution is a greater concern of the overall project, and is covered in greater detail in the DIP deliverable titled “Issue Management Plan”. It is worth noting the distinct differentiation between issue management in the ABT testing cycle and general issue management in the ABT program, because we plan to find problems in testing in order to fix them in the testing cycle and plan to avoid issues/problems at the program level. Therefore, issues, defects and problem related to testing will be treated with special process and be separated from problems/issues related to the ABT Program.

An issue is an activity, decision or other conflicting variable that requires some action in order for planned progress to return to a normal state. A defect is a short coming, bug or other type of system inadequacy that must be remedied in order for planned scope of functionality to return to a normal state. A problem is a challenge, constraint or other

limiting factor/event that must be fixed in order for planned progress/scope to return to a normal state. Issues, defects and problems are unanticipated and require external forces to bring them to resolution.

As mentioned, we fully expect to identify issues and uncover problems during testing, like software bugs and/or design, configuration and data issues. When issues, problems or defects are found during testing, they will be identified, documented on a Project Issues Form—or entered directly into an issue tracking system—and recorded in a central Issues Log to be monitored and managed by the Test Manager and/or Project Manager and ultimately resolved by the ABT Program Team.

## **2.8 Test Issue/Defect/Problem Identification and Submission**

Test issues will be submitted by the functional team lead. Test defects are submitted by testers as they uncover them. This helps ensure the appropriate person who knows and understands the details behind each issue/defect are the ones logging it. Issues and defects are managed similarly but separately. Key to documenting testing issues and defects is to categorize them by severity and priority.

**Defect Severity Levels** (established by Test Manager, Development Manager or Development Lead):

- **Critical** – significant impact on the project. Causes delay, scope change or increase in cost. The system test cannot proceed or be put into production until issue resolved. Testing stops.
- **Moderate** – moderate impact on the project. May cause delay or scope change. The system may be put into production without resolution only with the understanding this issue will be resolved as soon as possible. May create a significant work-around, which must be fully documented if required in production.
- **Low** – limited impact on the project. The issue will be placed on a list by relative importance for eventual resolution. The system may be put into production and operate with this issue unresolved.

**Issue/Defect Priority Levels** (established by Project Manager or Functional Lead):

- **Critical** – significant priority to the project. The system test cannot proceed or be put into production until issue resolved. Testing stops.
- **Moderate** – moderate priority to the project. The system may be put into production without resolution only with the understanding this issue will be resolved as soon as

possible. May create a significant work-around, which must be fully documented if required in production.

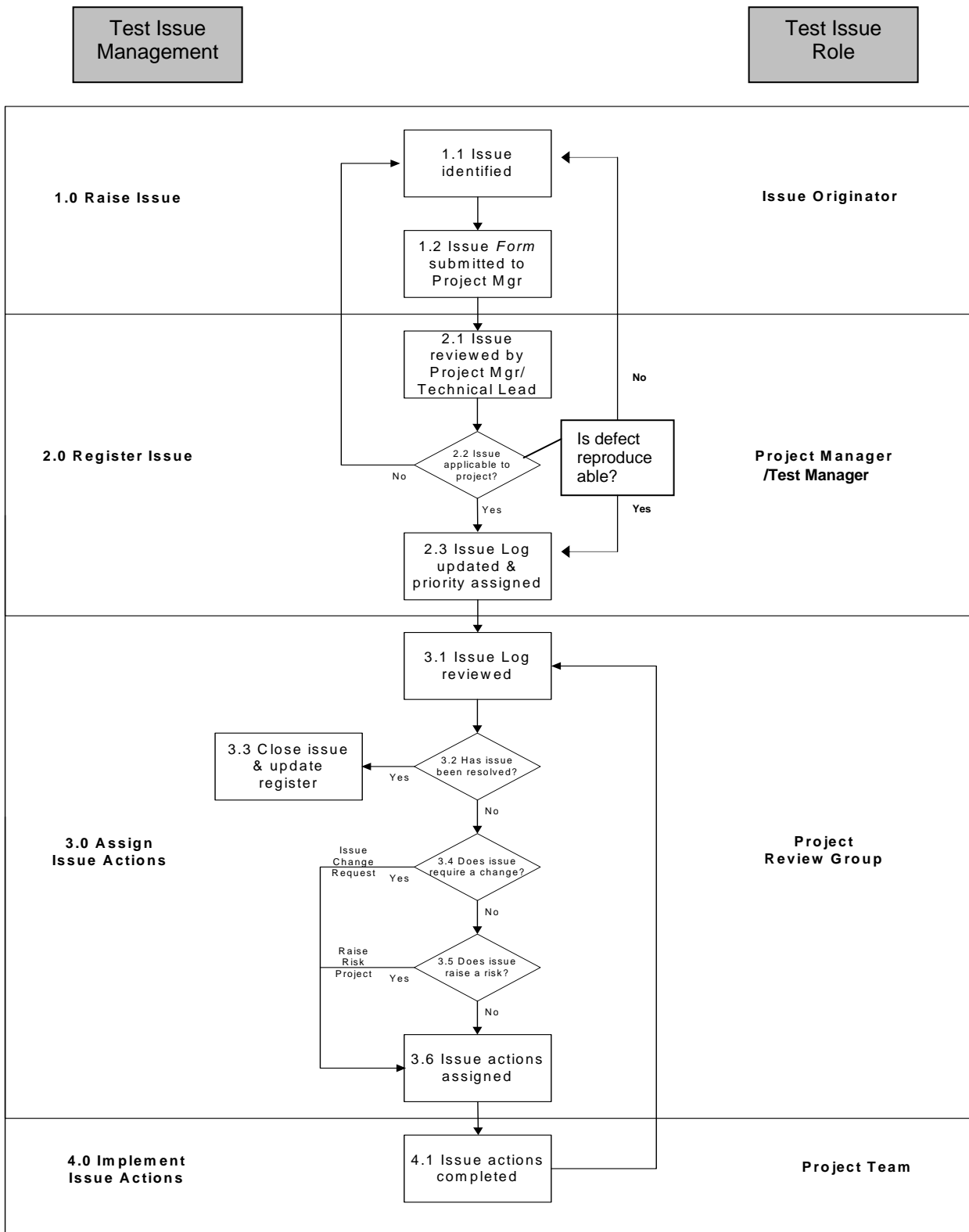
- **Low** – limited priority to the project. The issue will be placed on a list by relative importance for eventual resolution. The system may be put into production and operate with this issue unresolved.

## ***2.9 Test Issue Process, Escalation and Resolution***

Once an Issue has been identified and established, the Test Manager will manage the process to triage issues and defects and assign to the responsible team member for action.

See diagram below to illustrate the issues resolution process:





Note: Test issues and defects are handled separately.

## **2.10 Assumptions, Risks and Schedule**

### **2.10.1 Assumptions**

- As defined in the Business Process Design Strategy Plan, business requirements will be linked to business process, sub process and activity combination, which becomes a use case, or in this circumstance a “test case”. Therefore use case and test case become synonymous for testing.
- Business requirements (and test requirements) will be tracked as a use/test case, which links it to testing, and also to custom development, business process redesign and training. This referenceability and traceability is useful in user testing and user training.
- Test script templates will be provided by CIBER and completed jointly by the CIBER and county functional team leads.
- Test scripts will be based on real King County data.
- The tools, resources (including technical support) and infrastructure required for testing will be available as needed.
- King County and System Integration team members will work together to address/resolve issues or problems discovered during testing by actively working the issue through resolution.
- All business, business requirements, processes, configurations, modifications, interfaces, reports, and business cycles have defined test scripts and have been unit tested prior to integration test cycle.
- Development and Test environments will be maintained as close as possible to the Production environment (software versions, application configurations and patches).
- Conversion data and test data may need to be reloaded into the test environment after and environment refresh, depending on the situation.
- The test environment should be maintained as close to production environment as possible, including process scheduler, database, application and web servers for adequate load testing.
- Adequate project staffing for the expected goals and timeline.
- Adequate testing facilities, testing room for team members to coordinate next steps and share integrated testing results.
- Representatives from each testing group are part of the Program Team to facilitate consideration of enterprise-wide performance quality management objectives.

### **2.10.2 Risks**

- If issue/problem remediation is not completed within the specified time period, schedule delay may occur.

- If King County is unable to provide the required users for testing purposes, schedule delay may occur.
- If testing and validation require more effort and/or time than planned, schedule delay may occur.
- If development objects (data conversions, interfaces, reports and modifications) are not designed properly, developed properly or unit tested properly, integration testing will be adversely impacted, may create rework and/or may cause a schedule delay.
- If cross-team collaboration is not effectively managed for issue/defect management, the process may cause inefficiencies and/or schedule delays.

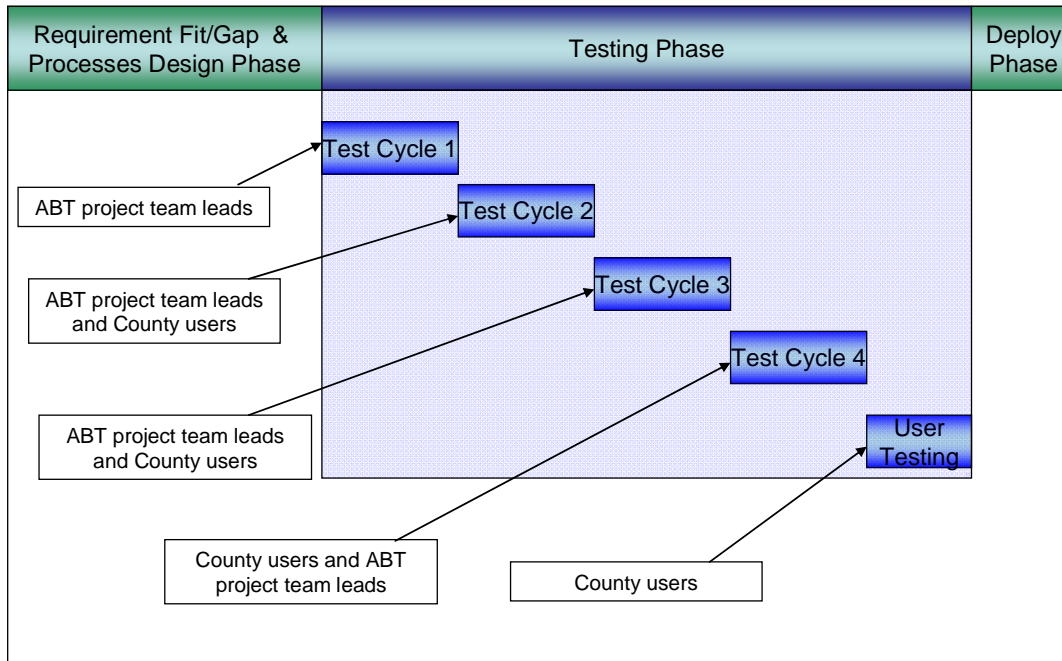
### **2.10.3 Mitigations**

- Closely monitor and manage issues/problems. Escalate early.
- Closely monitor progress of schedule and tasks. Keep Project Manager informed of any slippage and deviation from the project plan that will impact implementation.
- Train other staff to use testing tools and testing methodology to assist with testing as needed– especially to fill in gaps
- Evaluate testing tools to identify efficiencies.
- Prioritize testing efforts, testing critical and major processes as early in the schedule as possible.

### **2.10.4 Testing Plan, Schedule and Resources**

Given that the ABT implementation will have three major tracks (Oracle EBS financials, PeopleSoft HR/Payroll and budget) and multiple phases within each track, test plans will vary by track and phase. Therefore, for demonstration purposes, we will generalize the ABT test plan and resource requirements in the diagram below. (Note: a testing life-cycle may have between two and four test cycles; we've shown four below as an example.)

## Example Test Plan, Schedule and Resource Requirements



### Schedule

- Formal integration testing phase is planned for the second half of the ABT implementation. (The first half being the fit/gap requirements analysis, business process design and development design.)
- Informal component/Unit testing is an integral part of the fit/gap, business process design and development design activities and will occur during the first half of the ABT implementation (and likely into the second half as necessary).
- User testing is planned after integration testing with ample time planned, to ensure testing issues can be completed and issues addressed prior to go-live.
- Integration test cycles will be 4-8 weeks in duration in order to accommodate all testing and time for issue/problem resolution.