

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
I. Change Management Documentation	To ensure a formally documented change management process exists and is maintained to reflect the current process.	<p>1. Determine if a change management process(es) exists and is formally documented.</p> <p>2. Obtain a copy of the change management procedures and verify that they (at a minimum) include:</p> <ul style="list-style-type: none"> * Accountability for managing and coordinating changes; * The change management flow(s) within the organization; * The change management responsibilities of each organizational function; * The deliverables from each organizational component; * Specific timetables for reviewing and scheduling planned changes; * Specific timetables for the retention of historical records; * The handling of all changes, including change back-outs; * The circumstances when normal change management controls can be waived, and the methodology to be followed in those situations (e.g., emergency). <p>3. Determine the process used to identify and update user/system documentation as a result of the change(s) made.</p> <p>4. Determine if a process exists to maintain the change management procedures. Identify Oracle products and versions in use.</p>		

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
II. Change Initiation and Approval	To ensure change requests are properly initiated and approved.	<p>1. Verify a methodology is used for initiation and approval of changes.</p> <p>2. Ensure the request form includes (at a minimum) the following information:</p> <ul style="list-style-type: none"> * name of requester * phone number and department * requester's signature * reason for change * List of modules that need to be changed * Supervisor's name * Supervisor's approval (changes must be approved by someone other than the requester). <p>3. Determine if priorities are assigned to the change requests.</p> <p>4. Ensure estimated time of completion and budgeted costs are communicated.</p> <p>5. Evaluate the process used to control and monitor change requests (central repository and aging mechanism).</p>		
III. Modification or Development	Ensure code modification/development is performed in a segregated, controlled environment (separate from quality assurance (QA) and production).	<p>1. Ensure all changes are applied to a copy of the latest production version of code.</p> <p>2. Verify code is modified/developed in an area separate from testing/quality assurance, and production.</p> <p>3. Determine if programs can be checked out by more than one programmer simultaneously. Verify a process exists to support concurrent development.</p>		

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
		<p>4. Determine if a version control process exists to ensure the correct module was copied from production.</p> <p>5. Determine how the programmer is made aware of all the modules that need to be changed.</p> <p>6. Ensure history records are kept of code check-ins/outs, and deletions, which are made to the production library. Determine if a work order number is associated with the history record (this should be traceable back to the initial request).</p> <p>7. Verify a process exists that requires Programming Management to review the source code to ensure changes are appropriate and meet the departments programming and documentation standards.</p>		
IV. Testing and Acceptance	To ensure changes made to applications/systems are adequately tested before being placed into a production environment.	<p>1. Verify code is tested in a segregated/controlled environment (a testing/QA region which is separate from development and production).</p> <p>2. Determine how code is moved into the testing/QA environment.</p> <p>3. Determine who moves the code into the testing/QA environment.</p> <p>4. Determine a process exists to "freeze" code once migrated into the testing/quality assurance environment. This ensures no further changes can be made to the code while awaiting</p>		

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
		<p>User acceptance.</p> <p>5. Determine to what extent the User is involved in the testing process (e.g., preparation of tests and data).</p> <p>6. Ensure the test results are reviewed and approved by the User. Verify the method of User acceptance (e.g., verbal, written).</p> <p>7. Verify the existence of back-out procedures. These procedures should outline the process used to back code out of the testing/QA region, in the event the user does not approve the original changes and additional modifications are necessary.</p> <p>8. Ensure a process exists to document problems encountered during this phase of the change methodology. Determine how problems are followed-up and resolved.</p>		
V. Implementation	To ensure only authorized/approved software is moved into production.	<p>1. Verify procedures exist to ensure the approved code from the test environment is the version moved into production.</p> <p>2. Determine who is responsible for migration of code into production.</p> <p>3. Determine how code is implemented to the production environment.</p> <p>4. Verify the existence of back-out procedures. These procedures should outline the process used to back code out of the production.</p> <p>5. Determine if a process exists to reconcile changes. Verify who performs this process and how often the process takes place.</p>		
VI. Non-Emergency	To verify changes are properly authorized and	1. Select a sample of non-emergency changes (application/system) that have occurred during		

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
Changes	adhere to the established change control methodology.	<p>the period of review from the source program library directory.</p> <p>2. Using the sample selected in test #1, verify the following:</p> <ul style="list-style-type: none"> * All changes have been formally initiated, completely documented, and approved by someone other than the requester. * All changes have documentation stating code is ready to be moved from development to testing/QA with the authorized approvals. * All changes have documentation stating that they have been received and reviewed by a QA type function and approved by the User prior to installation into production. * Documentation exists showing a source comparison was performed prior to installation into production. A source comparison will determine if the current production source matches the current load program. <p>3. Obtain a copy of the change reconciliation report. Verify evidence exists for the review and reconciliation of changes.</p>		
VII. Emergency Changes	To ensure a process exists to control and supervise changes made in an emergency situation.	<p>1. Determine if a process exists to control and supervise emergency changes.</p> <p>2. Determine the use of emergency user ids. If emergency changes are made through the use of emergency Ids, ensure a process exists to enable and disable them (at a minimum 2 people should be involved in this process - if it is not automated).</p> <p>3. Ensure an audit trail exists of all emergency Id usage and that it is independently reviewed.</p> <p>4. Ensure emergency changes are approved by appropriate levels of management prior to implementation into production.</p>		

V. Change Management Work Program

Topic	Objective	Audit Procedures and Comments	Results	Workpaper Reference
		<p>5. Determine that procedures require that emergency changes be supported by appropriate documentation (e.g., evidence of management approval, code review) within one business day after the emergency is resolved.</p> <p>6. Verify a list of Business/Operations Management allowed to approve emergency changes exists. Programmers should not be able to initiate emergency changes.</p> <p>7. Determine if the approval of Business/Operations Management is required prior to the implementation of an emergency change.</p> <p>8. Ensure back-out procedures exist. These procedures should outline the process used to back code out of the production.</p>		