

DATE &
INITIAL W/P
REF

CHANGE MANAGEMENT

I. Description of Audit Area

Change Management is the process by which changes are planned, scheduled, applied, tested, accepted, distributed, and tracked. The process can consist of the development/conversion of a new system or the modification of existing code. Changes may have many internal and external impacts.

Change Management activities can impact a technology unit's ability to provide critical data processing and information delivery services to the corporation. It is necessary that each change be controlled throughout its life cycle and integrated into the production environment in a systematic and controlled manner.

The primary objective is to maintain the integrity and reliability of the production environment, while making changes for the User community.

	DATE & INITIAL	W/P REF
<p>Objectives</p> <p>To address all activities that will result in changes to the production environment, irrespective of the technology platform or source of the change. Activities include application and system development and modification, telecommunications. Network system (including changes to option settings on file servers), line equipment (including but not limited to changes in options, additions/modifications of addresses) and vendor supplied software modifications/upgrades are also addressed.</p> <p>The following are presented as minimum procedures that can be used to satisfy the attainment of the audit objectives. The bold-type procedures represent <i>core issues</i> that should be included in audit coverage of the Change Management area in every risk cycle. The subsequent indented procedures are suggested steps that may be taken in order to meet the criteria established in the core issues. Additional procedures may be added to this program as deemed necessary. Procedures to review controls over the computer operating environment and associated environmental controls not included in this program.</p> <p>Scope</p> <p>The scope includes:</p> <ul style="list-style-type: none"> • Review of documentation, policies and procedures regarding the change management process(es). • Evaluation of the Change Management Process(es), including, change initiation, development/modification of applications/systems, testing/QA, migration to production, and back-up/recovery. • Information security access restrictions to staging, testing/quality assurance, and production (source/load) libraries <p><u>I. PRELIMINARY DELIVERABLES</u></p> <p>A. Audit Checklist</p> <p>B. Organization Chart</p> <p>C. Process flowcharts and system narratives.</p> <p>Revised: 06/18/03</p>		

	DATE & INITIAL	W/P REF
<p>D. Most recent self-assessment, results and action plans.</p> <p>E. Naming conventions for <u>application production</u> and QA source, executable, command language and copy libraries/directories.</p> <p>F. Naming conventions for <u>system software production</u> and QA source, executable, parameter and command language libraries/directories.</p> <p>G. A report listing the total number of changes during time period under review (e.g., last 6 months):</p> <ol style="list-style-type: none"> 1. application: emergency/non-emergency 2. non-application: emergency/non-emergency. <p>H. Samples of all change management logs and forms.</p> <p>I. Current vendor documentation for any library management or change control software in use.</p> <p>II. <u>DOCUMENTATION, POLICIES AND PROCEDURES</u></p> <p>Objective: To ensure a formally documented change management process exists and is maintained to reflect the current process.</p> <p>Risk/Exposure: Lack of a formal change control process could result in the delivery of inconsistent and unreliable products.</p> <p>Tests:</p> <p>A. Determine if a change management process(es) exists and is formally documented.</p> <p>B. Determine that central applications/systems has a System level owner.</p> <p>C. Determine if change management operations has a current, comprehensive list of system owners.</p> <p>D. Obtain a copy of the change management procedures and verify that they (at a minimum) include:</p>		
<p>Revised: 06/18/03</p>		

DATE & W/P
INITIAL REF

1. Accountability for managing and coordinating changes;
2. The change management flow(s) within the organization;
3. The change management responsibilities of each organizational function;
4. The deliverables from each organizational component;
5. Specific timetables for reviewing and scheduling planned changes;
6. Specific timetables for the retention of historical records;
7. The handling of all changes, including change back-outs;
8. The circumstances when normal change management controls can be waived, and the methodology to be followed in those situations (e.g., emergency).

E. Determine the process used to identify and update user/system documentation as a result of the change(s) made.

F. Determine if a process exists to maintain the change management procedures.

III. CHANGE INITIATION AND APPROVAL

Objective:

To ensure change requests are properly initiated and approved.

Risk/Exposure:

Unauthorized changes could result in unpredictable business solutions that would not meet the users' requirements.

Tests:

A. Verify a methodology is used for initiation and approval of changes.

B. Ensure the request form includes (at a minimum) the following information:

1. Name of requester
2. Phone number and department
3. Requester's signature
4. Reason for change
5. List of modules that need to be changed
6. Supervisor's name

	DATE & INITIAL	W/P REF
<p>7. Supervisor's approval (changes must be approved by someone other than the requester).</p> <p>C. Determine if priorities are assigned to the change requests.</p> <p>D. Ensure estimated time of completion and budgeted costs are communicated.</p> <p>E. Evaluate the process used to control and monitor change requests (central repository and aging mechanism).</p> <p>F. Determine through trend analysis if there are systems that have an unusually high number of changes (which could be suggestive of other issues).</p>		

DATE & W/P
INITIAL REF

IV. MODIFICATION/DEVELOPMENT

Objective:

Ensure code modification/development is performed in a segregated, controlled environment (separate from quality assurance (QA) and production).

Risk/Exposure:

Modification/development of code may adversely affect other business systems if not performed in a segregated, controlled environment.

Tests:

A. Ensure all changes are applied to a copy of the latest production version of code.

B. Verify code is modified/developed in an area separate from testing/quality assurance, and production.

D. Determine if programs can be checked out by more than one programmer simultaneously. Verify a process exists to support concurrent development.

- Does the change management software have a checkout feature?
- Is the feature used?
- If the feature is not used, how are simultaneous checkouts controlled?

E. Determine if a version control process exists to ensure the correct module was copied from production.

F. Determine how the programmer is made aware of all the modules that need to be changed.

G. 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).

H. Verify a process exists that requires Programming Management to review the source documentation or code [if applicable] to ensure changes are appropriate and meet the departments programming and documentation standards.

	DATE & INITIAL	W/P REF

DATE & W/P
INITIAL REF

V. TESTING AND ACCEPTANCE

Objective:

To ensure changes made to applications/systems are adequately tested before being placed into a production environment.

Risk/Exposure:

Lack of (or inadequate) testing could result in the migration of unauthorized of code into production.

Tests:

- A. Verify code is tested in a segregated/controlled environment (a testing/QA region which is separate from development and production).
- B. Determine how code is moved into the testing/QA environment.
- C. Determine who moves the code into the testing/QA environment.
- D. 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 User acceptance.
- E. Determine to what extent the User is involved in the testing process (e.g., preparation of tests and data).
- E. Ensure the test results are reviewed and approved by the User. Verify the method of User acceptance (e.g., verbal, written).
- F. Determine that any changes resulting from user testing triggers a complete re-testing of the system.
- G. 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.
- H. Ensure a process exists to document problems encountered during this phase of the change methodology. Determine how problems are followed-up and resolved.

DATE & W/P
INITIAL REF

VI. IMPLEMENTATION

Objective:

To ensure only authorized/approved software is moved into production.

Risk/Exposure:

Unauthorized software migrated into production could adversely impact the production environment.

Tests:

- A. Verify procedures exist to ensure the approved code from the test environment is the version moved into production.
- B. Determine who is responsible for migration of code into production.
- C. Determine how code is implemented into the production environment.
- D. Verify the existence of back-out procedures. These procedures should outline the process used to back code out of production and reinstall the most recent version of the code.
- E. Determine if a process exists to reconcile changes scheduled for implementation to those changes actually implemented. Verify who performs this process and how often the process takes place.

VII. NON-EMERGENCY CHANGE MANAGEMENT COMPLIANCE TESTING

Objective:

To verify changes are properly authorized and adhere to the established change control methodology.

Risk/Exposure:

Lack of a change control process could result in un-tested/unauthorized migration of code into production. This could result in delays in production processing, hence, customer dissatisfaction. Also, this could result in unauthorized changes adversely affecting application processing to produce unintended results.

Tests:

DATE & W/P
INITIAL REF

A. Select a sample of non-emergency changes (application/system) that have occurred during the period of review from the source program library directory.

B. Using the sample selected in test #1, verify the following:

1. All changes have been formally initiated, completely documented, and approved by the system owner.
2. All changes have documentation stating code is ready to be moved from development to testing/QA with the authorized approvals.
3. 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.
4. Review User test documentation for adequacy and proper signoff.
5. Documentation exists showing a source comparison was performed prior to installation into production ensuring consistency between source and object code (if applicable).

VIII. EMERGENCY CHANGE MANAGEMENT

Objective:

To ensure a process exists to control and supervise changes made in an emergency situation.

Risk/Exposure:

Lack of an emergency change process could result in the unauthorized migration of code into production. This may result in delays in production processing, hence, customer dissatisfaction.

Tests:

A. Determine if a process exists to control and supervise emergency changes.

B. 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).

	DATE & INITIAL	W/P REF
<p>C. Ensure an audit trail exists of all emergency ID usage and that it is independently reviewed.</p> <p>D. Ensure emergency changes are approved by appropriate levels of management, prior to implementation into production.</p> <p>E. 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>F. Verify a list of Business/Operations Management allowed to approve emergency changes exists. Programmers should not be able to initiate emergency changes.</p> <p>G. Determine if the approval of Business/Operations Management is required prior to the implementation of an emergency change.</p> <p>H. Ensure back-out procedures exist. These procedures should outline the process used to back code out of the production.</p> <p>I. Determine the number of emergency changes made during the audit period under review. Analyze the volume of emergency access requests and determine if it appears to be excessive.</p> <p>J. Determine if emergency fixes are closed out in a reasonable amount of time.</p>		
<p><u>IX. EMERGENCY CHANGE MANAGEMENT AUDIT COMPLIANCE TESTING</u></p>		
<p>Objective: To ensure a process exists to control and supervise changes made in an emergency situation.</p>		
<p>Risk/Exposure: Lack of a process to control emergency changes could result in unauthorized changes being moved into production. This may adversely affect production processing, and result in customer dissatisfaction.</p>		
<p>Revised: 06/18/03</p>		

	DATE & INITIAL	W/P REF
<p>Tests:</p> <p>A. Select a sample of emergency changes that have occurred during the audit period under review. Determine if any of the changes should have gone through the non-emergency change process.</p> <p>B. Using the sample selected in test #1, determine if the changes have been made in compliance with the established procedures.</p> <p>C. Using the sample selected in test #1, verify that the date on the approval documentation is not more than one day after the date on the executable module.</p> <p>X. <u>SECURITY</u></p> <p>Objective: Ensure access to change management libraries is restricted to authorized personnel.</p> <p>Risk/Exposure: Unauthorized access could result in the intentional or inadvertent modification and/or destruction of application or system software.</p> <p>Tests:</p> <p>A. Obtain a list of the application and system, production and test/QA source, executable libraries/directories.</p> <p>B. Review security rules to ensure access has been restricted to authorized individuals.</p> <p>C. Determine that access to Acceptance Libraries is properly restricted to Users, Production Control and Information Security staff.</p>		