

**Support Services Policies  
and Procedures Document**

---

# Support Services Change Control Policy and Procedures

Release Version 1.0

All trade names, trademarks, or registered trademarks are trade names, trademarks, or registered trademarks of their respective companies.

The filename for this document is:

c:\my documents\processdcox\changecontrol.doc

The owner of this document is: Mike Tarrani

This document expires <expiration date will be 6 months after date of release copy>

### Document Classification: Company Confidential

Physical and Administrative Controls	Reproduction	Distribution	Destruction/Disposal
Each employee is responsible for controlling usage and access on a need-to-know basis. This policy also applies to contractors.	Limited copies may be made only to employees or contractors who have signed a non-disclosure agreement.	Internal: Use company envelope whenever possible. External: Use sealed envelope.	Use document destruction bins. Shred or erase magnetically recorded documents if unable to recycle.

For additional copies of this document, please contact:

## Document Revision

[illegible]

## Document Change/Correction Request

Your feedback is valuable. Anyone who uses this document is invited to use this form to submit requests for changes or corrections to this document. Please complete all of the sections and mail to:

## Document Identification

Document title	Support Services Change Control Policy And Procedures
Version	Baseline
Date of Document	22 November 2005
Master File Location	c:\my documents\processsdox\changecontrol.doc

## Requester Identification

Name	
Organization	
Phone Number	( ) -
E-mail address	

## Request

☐ The following change is requested

Please describe the change and reason(s) why the change is requested.

☐ Correction

Section \_\_\_\_, page \_\_\_\_ contains the following erroneous information:

This information should be changed to:

Attach additional sheets if necessary to fully describe the change or correction

# Table of Contents

<b>Document Roadmap.....</b>	<b>1</b>
<b>Introduction .....</b>	<b>2</b>
1. Purpose.....	2
1.1 Scope .....	2
1.2 Audience.....	2
1.3 Glossary of Terms .....	3
<b>Policy .....</b>	<b>8</b>
2. Statement of Policy .....	8
<b>Process.....</b>	<b>9</b>
3. Process Description.....	9
3.1 Entry Criteria .....	9
3.2 Tasks .....	10
3.3 Validation .....	11
3.4 Exit Criteria .....	12
3.5 Roles and Responsibilities .....	12
<b>Procedures.....</b>	<b>15</b>
4. Overview.....	15
4.1 Verifying and Validating Entry Criteria .....	15
4.2 Initiating a Change Request.....	17
4.3 Performing an Impact Analysis .....	18
4.3.1 Dependency Analysis .....	18
4.3.2 Risk Assessment .....	19
4.3.3 Additional Factors.....	20
4.4 Developing a Planning Package .....	21
4.4.1 Implementation Plan of Action and Milestones .....	21
4.4.2 Roll-Back Plan .....	22
4.4.3 Escalation Plan.....	27
4.4.4 Communications Plan.....	28
4.5 Reviewing the Change Request Package.....	28
4.6 Opening Change Request with COMPANY GCCB .....	29
4.7 Implementation Results.....	30
4.8 Performing a Post-Implementation Validation .....	32
4.8.1 Process Summary.....	32
4.8.2 Controls .....	32
Entry criteria .....	32
4.8.4 Tasks .....	33
4.8.5 Validation .....	33
4.8.6 Exit criteria.....	34
4.8.7 Constraints .....	34
4.8.8 Critical Success Factors/Key Performance Indicators .....	35
4.9 Post-Implementation Review.....	36
4.10 Root Cause Analysis and Process Improvement.....	37
<b>Attachments.....</b>	<b>39</b>
Support Services Change Control Request & Implementation Plan.....	39
Post-Implementation Validation (PIV) Worksheet.....	39



# Document Roadmap

Who should read what:

Key Role	Introduction	Policy	Process	Procedures
	<b><u>ESSENTIAL - WHO</u></b> Purpose, Scope, Audience and Glossary of Terms	<b><u>ESSENTIAL - WHY</u></b> Statement of Policy and Enforcement	<b><u>OVERVIEW - WHAT</u></b> High Level Description of the Change Control Process	<b><u>DETAILS - HOW</u></b> Detailed Description of Procedures
<b>Support Services Manager</b>	Responsible	Responsible	Responsible	Responsible
<b>Support Services Change Control Coordinator</b> <i>(Normally assigned as a collateral duty to the Application Support Coordinator)</i>	Responsible	Responsible	Responsible	Responsible
<b>Application Support Manager</b>	Responsible	Responsible	Responsible	Responsible
<b>Application Support Analyst (ASA)</b>	Responsible	Responsible	Responsible	Responsible
<b>Business System Manager (BSM) [Some tasks may be assigned to a Business Systems Analyst (BSA)]</b>	Responsible	Responsible	Responsible	Responsible
<b>Application Owner</b>	Responsible	Responsible	Optional	Optional
<b>Global Change Control Board Coordinator</b>	Responsible	Responsible	Responsible	Responsible
<b>SME/Technical Domain Owner</b>	Optional	Responsible	Optional	Optional

# **Introduction**

## **1. Purpose**

This policy and procedures document sets forth <ORGANIZATION NAME> policy for making changes to production systems under the cognizance of Support Services, and explains the process and procedures for controlling changes in accordance with the policy. The immediate objective is to institute the policy in the Support Services domain and establish processes and procedures in support of the policy. The overall objective is to consolidate the policy and procedures with change management initiatives and processes that are already in place in other functional areas within <ORGANIZATION NAME>.

### **1.1 Scope**

Change management policy and procedures are applicable to the <ORGANIZATION NAME> Support Services domain.

### **1.2 Audience**

Support Services Staff - sets policy for <ORGANIZATION NAME> Support Services staff with respect to making changes to production systems and describes the change control process and procedures that are used to conform to the policy.

Business Systems Managers and Business Systems Analysts - provides guidelines for initiating change requests and developing implementation and roll-back plans.

Application owners and end users of supported applications - describes how to initiate a change request and gives an overview of the change control process.

Technical domains within <ORGANIZATION NAME> - provides entry and notification criteria for making changes to any system or application under the cognizance of the Support Services domain. This includes vendors and contractors sponsored by, or performing services for, these technical domains.

Support Services sponsored vendors and contractors - all vendors and contractors sponsored by, or performing services in behalf of, the Support Services domain are required to adhere to the policy and follow the process and procedures set forth in this document.



### 1.3 Glossary of Terms

<b>application</b>	A system that provides a specific set of functions and/or services to end users in support of business objectives. The term is commonly associated with a specific software application, such as an accounting system. Within the context of this policy and procedures document an application can also include integrated components, such as back-end server processes, desktop client, middleware (such as a transaction process monitor) and a database management system to form a cohesive set of functions and services.
<b>application owner</b>	The business owner of an application. Commonly called <i>business process owner</i> outside of the <ORGANIZATION NAME> environment.
<b>application support analyst</b>	Abbreviated: ASA. A member of the Support Services team who is responsible for providing issue management, application maintenance and tier-2 (and sometimes tier-3) support for a specific application.
<b>build</b>	Synonyms: <i>develop, development, construct</i> . Milestone in the system development life cycle during which the design is translated into a product or application, or enhancements or fixes are added to an existing product or application. See <i>life cycle</i> .
<b>build analysis</b>	Verification that the correct software configuration items were included in a software build, and that the configuration items successfully passed unit and integration testing in development prior to the build. A build is the compiling and linking of all software configuration items for a specific application.
<b>business system analyst</b>	Abbreviated: BSA. Supports the BSM (see below) and serves as a subject matter expert in required business and/or technical disciplines in support of a specific or group of applications and/or systems. Also functions as project manager, systems analyst and business analyst.
<b>business system manager</b>	Abbreviated: BSM. The technical manager of a specific application or system. The BSM interfaces and has direct communication with the application owner in all matters pertaining to the support of the application, including emergent requirements, service level attainment and changes to the application in the production environment.
<b>design</b>	Milestone in the system development life cycle where requirements are translated into specifications and design documents. See <i>life cycle</i> .
<b>emergency maintenance</b>	Maintenance that is required to resolve a severity 1 or severity 2 issue.
<b>ETVX</b>	An abbreviation for Entry-Task-Validation-Exit model. Processes set forth in this document have been designed in accordance with the ETVX model. Specific components are: <b>Entry criteria</b> - defines what must be provided or accomplished before the process can proceed. <b>Tasks</b> is the sequence of steps to meet process objectives. <b>Validation</b> identifies the quality checkpoints in the process, and <b>Exit criteria</b> describes the conditions that must be met before the process can be successfully terminated.
<b>hardware configuration item</b>	Abbreviated: HCI, also abbreviated as CHCI (computer hardware configuration item). Any component of a hardware platform, such as memory, mass storage, physical interfaces, etc.
<b>impact analysis</b>	An examination of inter- and intra-system dependencies and the effects that will result from making a change. Impact analysis takes into consideration operational requirements (service level objectives, business operations, support, etc.), and how the change will affect other internal or external systems, subsystems or components, etc. An impact analysis of software configuration items used in a specific release or patch is called a <i>build analysis</i> .
<b>life cycle</b>	Synonym: <i>System development life cycle, SDLC</i> . The life of a system measured from inception through retirement. A typical life cycle will have the following milestones: requirements analysis, design, build, test, roll-out, sustainment and retirement. Each milestone may be iterative because systems usually undergo continuous refinement. As such a system at the sustainment milestone will have many iterations of requirements-design-build-test-implement as it is upgraded to add new functions and features and fix defects.

<b><i>maintenance window</i></b>	Time set aside to perform normal system maintenance, such as back-ups, preventative mechanical maintenance, upgrades, etc. Maintenance windows are defined in service level agreements (SLAs) as a function of availability. For example, a service level objective for a business-critical system might state that the system will be available on a 24x7 basis for twenty consecutive days, with a four-hour period set aside on every 21 <sup>st</sup> day to perform maintenance. This four-hour period is the maintenance window.
----------------------------------	--

*Glossary of Terms (continued)*

<b>operational level objective</b>	Abbreviated: OLA. (1) An agreement between two or more inter-dependent systems or infrastructure owners that defines an aggregate maintenance window that is scheduled in such a manner that the service level objectives for each of the systems can be met using a common maintenance window that does not exceed what is specified in any of the systems' SLAs. (2) Notification criteria that specifies how much advance notice a system or infrastructure owner must give to owners of inter-dependent system prior to performing maintenance that would affect the other systems. (3) Entry or exit criteria that must be met by the owner of a system that will affect inter-dependent systems.
<b>planned maintenance</b>	(1) Recurring maintenance, such as back-ups, that are scheduled to occur during a maintenance window. (2) Maintenance requirements, such as upgrades and patches, that are scheduled in accordance with an implementation plan to occur during a maintenance window.
<b>post implementation validation</b>	Abbreviated PIV. Synonyms: <i>cycle 0 testing</i> , <i>sanity check</i> , <i>sanity test</i> . A series of tests that take place after a change is made to the production environment, but prior to formal release to production. The goal of PIV is to exercise all major system or application subsystems and interfaces and to observe the system or application's stability and performance in the production environment prior to formally releasing the system or application into production (sustainment). The observation period with a real end users operating in the production environment is typically one hour. If no severity 1 or 2 issues occur during the observation period the implementation is deemed to be successful and the system or application is transferred to sustainment. If severity 1 or 2 issues do occur during PIV the change is rolled-back.
<b>pre-production environment</b>	Synonyms: <i>pre-production test environment</i> , <i>pre-production</i> , <i>staging environment</i> . A test environment that is configured identically to the production system. The purpose of this environment is to test patches and minor releases for defects prior to implementing them into the production environment. A secondary purpose, when applicable, is to verify and validate functions and features that were included in a minor release or patch. This type of testing is normally performed in the product test/UAT environment; however, minor releases and patches bypass that environment, requiring the verification and validation of functions and features in the pre-production environment. Successful test results from the pre-production environment comprises one of the entry criteria for the change control process.
<b>product</b>	Any hardware or software system designed to provide specific services or functions. A product can be an application, an add-on module or feature set to a hardware or software system or subsystem, an operating system, database management system or infrastructure component such as a router or switch.
<b>product test</b>	See <i>user acceptance testing</i> .
<b>quality control</b>	Abbreviation: QA. Synonyms: <i>quality assurance</i> (not preferred because of the similarity to the term <i>Software Quality Assurance</i> ), <i>testing</i> , <i>pre-production testing</i> . A term used to describe the user acceptance or pre-production test functions, which are milestones in the system development life cycle. Often incorrectly called software quality assurance (SQA). Software quality assurance is a proactive, metrics-based approach to software quality that monitors critical indicators at each stage of the systems development life cycle, while quality control is a reactive, inspection-based approach to either hardware or software quality that occurs after the build milestone, but before the implementation (roll-out) milestone in the systems development life cycle.
<b>regression testing</b>	A quality assurance function that re-runs a full suite of test cases against any system that has been changed by a patch or upgrade. The purpose of regression testing is to ensure that there are no undocumented or non-obvious dependencies between the existing system and the subsystems, modules or components that have been changed.
<b>release</b>	The promotion of a change into the production environment and hand-off from development and implementation to sustainment. This is a critical milestone that occurs after a change has been implemented and has successfully passed the PIV checkpoint, and has been accepted by the BSM and application owner.



*Glossary of Terms (continued)*

<b>release notes</b>	Software turnover documentation. Typically contains: list of deliverable items contained in the release (special instructions, updated documentation, media, scripts, executables, etc.), description of release specifications, list of fixes/features, (by issue number in release), installation requirements and impacts, test results with remaining open issues and closed issues, and any additional information needed to implement and support the product
<b>requirements</b>	Initial stage in the system development life cycle during which functional and technical requirements are gathered. Requirements are the basis for design. See <i>life cycle</i> .
<b>risk assessment</b>	Identification of risks, their impact and methods or strategy to eliminate or mitigate the risks or their impact
<b>roll-back</b>	Synonym: <i>back-out</i> . To uninstall or remove a change and restore a system to its previous state.
<b>roll-out</b>	Synonym: <i>Implementation and release</i> . Milestone in the system development life cycle that occurs after successful completion of user acceptance and quality control testing but prior to post-implementation verification. See entries for <i>life cycle and release</i> .
<b>service level objective</b>	Abbreviated: SLO. An objective or goal in support of providing services. Typical service level objectives are: availability, key transaction performance, uptime, etc. Service level objectives are the foundation of service level agreements (SLAs)
<b>severity</b>	The degree of impact an problem has on business operations. Examples: <ol style="list-style-type: none"> <li>1. <b>Severity One</b> Loss of application, or critical performance degradation, with no workaround. Incident affects an entire workgroup.</li> <li>2. <b>Severity Two</b> Moderate application degradation incidents. Severity One workaround. Incident affects several customers.</li> <li>3. <b>Severity Three</b> Minor application degradation incidents. Incident or request has medium to high impact on single customer's ability to work.</li> <li>4. <b>Severity Four</b> Incident or request has a low impact on single customer's ability to work..</li> </ol>
<b>SME</b>	An abbreviation for Subject Matter Expert.
<b>software configuration item</b>	Abbreviated: SCI; also abbreviated as CSCI (computer software configuration item).
<b>staging environment</b>	See: pre-production environment
<b>sustainment</b>	Synonyms: <i>operations, operational system, maintenance milestone, production system</i> . (1) Milestone in a systems life cycle. (2) Support provided by IT to maintain systems that are in operation in support of business processes. (3) Systems used by end users to support business processes
<b>system</b>	A collection of subsystems and components that comprises an integrated environment that provides services or functions. Systems are typically comprised of hardware and software, and many also include databases, middleware and communications facilities
<b>technical domain</b>	A functional area defined by related technical specialties. For example, database management is a technical domain that includes data architects, database administrators and data stewards (application owners who control the content and quality of their application's data)
<b>tiered support</b>	The support hierarchy that starts with tier-1 (help desk services), which opens, logs and assigns issues. Issues that cannot be resolved by tier-1 are elevated [assigned] to tier-2, which is typically an application support analyst (see definition). If the issue can only be resolved by a developer or third-party vendor it is elevated to tier-3.
<b>user acceptance test</b>	Abbreviated: UAT. Synonyms: <i>product test, functional test</i> . A test environment that supports the next release planned for the production environment. The purpose of the user acceptance test environment and supporting activities is to verify and validate features and functions included in the product being tested. Verification and validation is performed against contract specifications, functional requirements specifications, technical specifications and design documentation.

## Policy

### 2. Statement of Policy

It is the policy of <ORGANIZATION NAME> to manage the life cycle of all information systems supporting its business and technical objectives. As such, the processes and procedures for change control set forth in this policy document governs change, and release management. The scope of this policy is the management of changes to the production environment. Specifically:

Before any change to a system or a baseline, the proposed change will be evaluated and approved by the <ORGANIZATION NAME> Support Services Change Control Board.

No approved change will be implemented without:

1. Entry criteria needed to initiate the change control process.
2. An approved plan of action with milestones for implementation, that provides a sequence of events or steps for implementing and releasing the change into the production environment, a roll-back plan, assigned roles and responsibilities and post implementation validation (PIV) test plan.
3. A completed test plan showing the results of testing the change in a pre-production or staging environment.
4. Approval from the application owner(s) affected by the change and the business systems manager responsible for the application or system being changed.
5. A formal review by the <ORGANIZATION NAME> Support Services Change Control Board to ensure that all entry criteria for the change have been met.

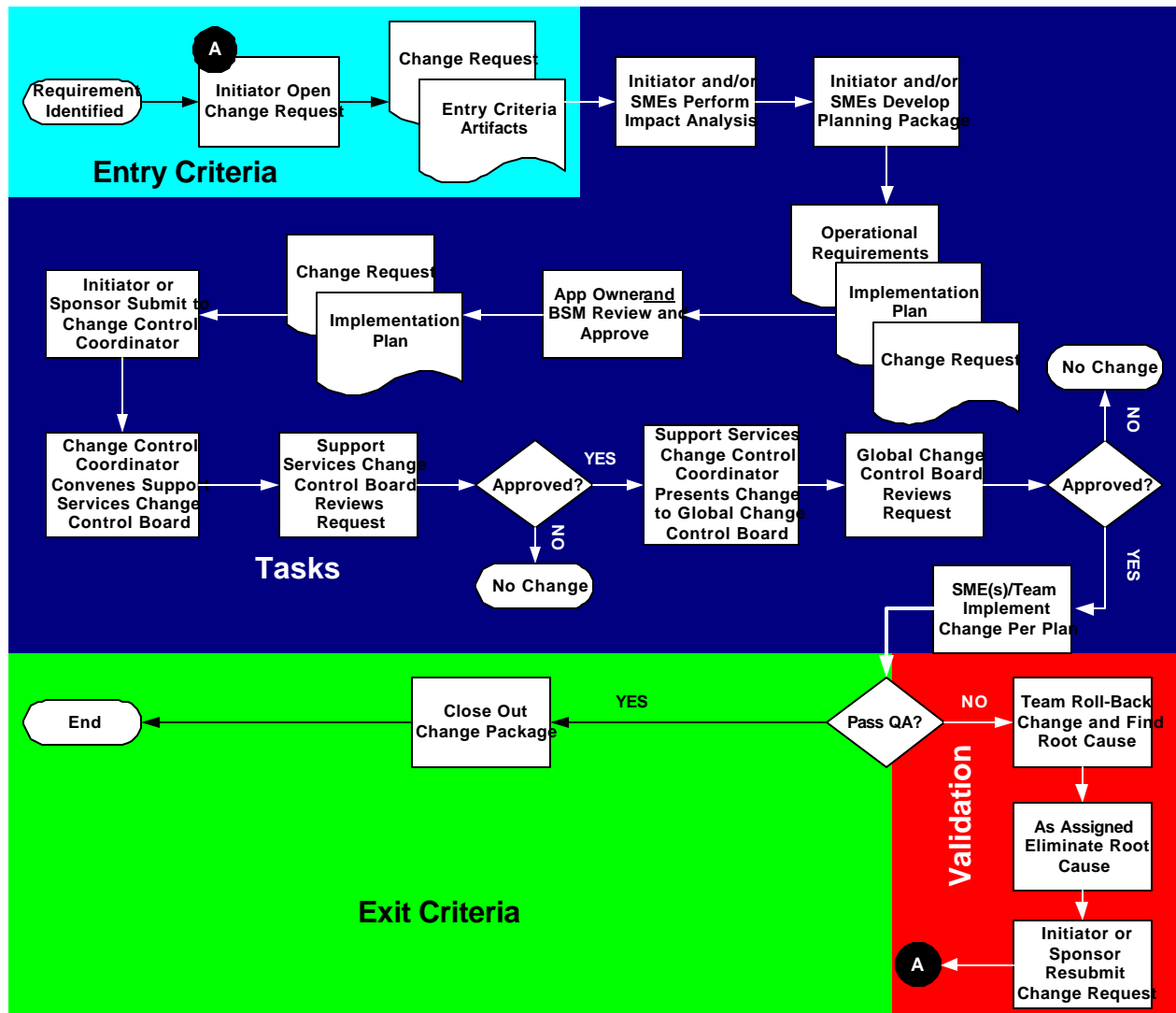
Any system or application failure or defect traced to a change made to a <ORGANIZATION NAME> system or application that was not made in accordance with this policy, process and procedures will result in disciplinary action. Specifically:

1. The error will be communicated to all stakeholders of the affected system and/or application.
2. Individual(s) making the unauthorized change will be required to develop an action plan specifying which measures will be taken to avoid a future occurrence of the failure or defect.
3. The action plan will be reviewed and approved by the individual's management chain and posted in a public place for review.

## Process

### 3. Process Description

The change control process follows the entry-task-validation-exit (ETVX) model, and is depicted in the following diagram:



#### 3.1 Entry Criteria

The change control process is initiated when there is a requirement to make a change. Change is defined as any of the following:

1. New system - application, operating system, database, hardware platform or infrastructure.

2. Major upgrade to an existing system - version release, new or upgraded components and/or subsystems (hardware or software), database schema reorganization, etc.
3. Minor upgrades to an existing system - patches, modifications to existing scripts or additional scripts (batch, shell, SQL, etc.), minor database schema reorganization (dropping columns, adding or modifying constraints, triggers and stored procedures, etc.) and infrastructure changes that are transparent to end users (i.e., upgrading IOS in a Cisco router, etc.).
4. Changes to service level objectives - permanent maintenance window changes, changes to problem management response times, mean-time-to-repair metrics, availability commitments, etc.
5. Maintenance to any system that has dependencies with the system being managed - in this special case the BSM or BSA will open a change request to document the maintenance being performed on the inter-dependent system even though the BSM or BSA has no direct control over, or responsibility for, the system. For example, if a particular application exchanges data with an application that is managed and supported by different BSMs, BSAs and ASAs, and is owned by a different application owner a dependency exists. The BSM (or BSA if applicable) for the external application are responsible for initiating change control. However, since the change will affect the second application, that application's BSM will open a change request as well. This provision will ensure that the scope of the required impact analysis will extend to all systems that are affected by the change. It will also ensure that each BSM, BSA and ASA remain cognizant of any change or maintenance activity that affects his or her system.

The following are the minimum entry criteria that must be met before the process can move to the task stage:

1. Release notes, build analyses, installation manuals and any other documentation that is needed to correctly test and install the product (hardware or software).
2. Test results from QA (product test/UAT and/or pre-production/staging).
3. Operational requirements, such as special training, maintenance window considerations, help desk entry criteria, spare parts, etc.

### **3.2 Tasks**

1. Perform an impact analysis. Deliverable: completed impact analysis.
2. Develop planning package. Deliverable: description of change and why change is being made (including benefits and how the change will create value for the users), how the change will affect users during the implementation (scheduled start and



end time, impact on maintenance window and service level objectives)  
implementation plan, roll-back plan, roles and responsibilities, notifications, quality assurance plan.

3. Provide operational requirements, implementation plan and change request to application owner and BSM for review and approval.
4. Application owner approve change.
5. Business systems manager approve change.
6. Submit change control package to change control coordinator.
7. Support Services Change Control Coordinator reviews package for completeness and presents at the next convened change control board.
8. Support Services change control board reviews and approves the change request.
9. Support Services Change Control Coordinator presents change at Global Change Control Board.
10. Global Change Control Board reviews and approves the change request.
11. Change is implemented in accordance with implementation plan.
12. Change action is closed out as complete.

### **3.3 Validation**

The following are checkpoints in the change control process:

1. All entry criteria will be checked for accuracy and completeness by the Support Services Change Control Coordinator.
2. Application owner will review and approve the change request before proceeding.
3. Business systems manager will review and approve the change request before proceeding.
4. The change control coordinator will review the implementation plan and change request for accuracy and completeness before including the change as an agenda item at the next scheduled change control board.
5. The change will successfully pass all post implementation validation test checkpoints before the change is released into production, else the change will be rolled-back.

6. In the event of a roll-back there will be a root cause analysis performed and responsibility for eliminating the root cause and, when applicable, developing a process improvement plan will be assigned to individual(s) by cognizant authority. The change request will also be cancelled and resubmitted after the root cause has been determined and eliminated.

### 3.4 Exit Criteria

1. The change is successfully released into the production environment or cancelled and resubmitted depending on validation checkpoints above.
2. After a change is successfully released into the production environment the change control coordinator will close out the change request as completed.

### 3.5 Roles and Responsibilities

Change Control Process Owner	Process Owned	Specific Responsibilities
<b>Support Services Manager</b>	Change control policy, process and procedures for the Support Services domain.	<ol style="list-style-type: none"> <li>1. Enforce change control policy within the Support Services domain.</li> <li>2. Ensure that Support Services change control policy, process and procedures remain aligned to COMPANY business and technical objectives by conducting a review of the policy, process and procedures every six months and making any changes required to reflect changes to business and technical objectives.</li> <li>3. Appoint a Support Services Change Control Coordinator.</li> </ol>
<b>Support Services Change Control Coordinator</b> <i>(Normally assigned as a collateral duty to the Application Support Coordinator)</i>	Support Services change control review and approval process.	<ol style="list-style-type: none"> <li>1. Review change control packages for accuracy and completeness.</li> <li>2. Maintain a log of open, pending and closed change requests.</li> <li>3. Coordinate with other technical domains to ensure that any system or subsystem identified in the impact analysis is aware of the proposed change(s) and will participate in the change control board review and approval process.</li> <li>4. Open change control issues in Global Change Control Board tracking system.</li> <li>5. Participate in Global Change Control Board meetings and present application changes to the board.</li> <li>6. Provide status reports to the GCCB and to other functional and management representatives as directed by the Support Services Manager and Application Support Manager.</li> </ol>
<b>Application Support Manager</b>	Support Services Change Control compliance monitoring and escalation processes..	<ol style="list-style-type: none"> <li>1. Attend Mainframe change control board meetings and disseminate information applicable to Support Services/Applications Support to ASAs.</li> <li>2. Ensure compliance with Support Services Change Control policies and procedures within the Application Support domain.</li> <li>3. Function as situation manager for issues and escalations during change implementations.</li> </ol>

*Roles and Responsibilities (continued)*

Change Control Process Owner	Process Owned	Specific Responsibilities
<b>Application Support Analyst (ASA)</b>	Monitor the change process, serve as application SME and perform Post-implementation validation.	<ol style="list-style-type: none"> <li>1. Performs or monitors change implementation.</li> <li>2. Performs post-implementation validation (PIV).</li> <li>3. Performs other duties in support of change request initiation, meeting entry and/or exit criteria, performing change control and implementation tasks, validation, root cause analysis or process improvement as directed by the BSM and/or required by the Support Services Change Control Coordinator.</li> <li>4. Functions as an SME and/or stakeholder when reviewing impact analyses or change requests initiated by technical domain owners outside of Support Services.</li> </ol>
<b>Business System Manager (BSM) [Some tasks may be assigned to a Business Systems Analyst (BSA)]</b>	Change control process for application.	<ol style="list-style-type: none"> <li>1. Responsible and accountable for complying with change control policy and the integrity of applications and systems under his/her cognizance .</li> <li>2. Notifies ASA of change requirements.</li> <li>3. Ensures that all entry criteria associated with change requests have been met.</li> <li>4. Initiates change requests (if the change request is initiated by a BSA it will be under the direction of the BSM).</li> <li>5. Performs impact analysis.</li> <li>6. Develops implementation and roll-back plan.</li> <li>7. Obtains application owner approval for requested changes.</li> <li>8. Reviews and approves impact analysis, planning and post-implementation packages that were developed by BSAs or other SMEs.</li> <li>9. Reviews all entry criteria and proposes to application owner any waivers to test results that may be appropriate with respect to accepting known defects into the production environment.</li> <li>10. Negotiates with the application owner [any] extended maintenance windows or other waivers to service level objectives that may be required to support the implementation of a change.</li> <li>11. Designated technical approving authority for all changes to the applications and systems under his/her cognizance.</li> <li>12. Responsible and accountable for the completion and quality of root cause analysis, root cause elimination and process improvement requirements in connection with rolled-back changes or unauthorized changes that result in application or system failures or defects.</li> <li>13. Evaluates impact analyses or change requests initiated by technical domain owners outside of Support Services and either functions as the SME or delegates this responsibility to the BSA. If this responsibility is delegated the BSM will retain accountability for quality and results.</li> </ol>
<b>Application Owner</b>	Determination of change impact to business operations.	<ol style="list-style-type: none"> <li>1. Reviews change requests for impact to business operations based on degree of risk associated with the requested change, trade-off between accepting known defects and deferring the change.</li> <li>2. Reviews and approves any extended maintenance windows or waivers to meeting service level objectives that are associated with the requested change.</li> </ol>

*Roles and Responsibilities (continued)*

<b>Change Control Process Owner</b>	<b>Process Owned</b>	<b>Specific Responsibilities</b>
<b>Global Change Control Board Coordinator</b>	Global change control review and approval process.	<ol style="list-style-type: none"><li>1. Review change control packages for accuracy and completeness.</li><li>2. Maintain a log of open, pending and closed change requests.</li><li>3. Coordinate with other technical domains to ensure that any system or subsystem identified in the impact analysis is aware of the proposed change(s) and will participate in the change control board review and approval process.</li><li>4. Convene the Global Change Control Board.</li><li>5. Track change status to closure.</li></ol>
<b>SME/Technical Domain Owner</b>	Impact analysis and communication .	<ol style="list-style-type: none"><li>1. Reviews or develops impact analyses.</li><li>2. Communicates planned changes to owners of applications or systems that are affected by the change.</li><li>3. Serves as a member of the Support Services change control board when assigned by their supervisor or manager, BSM or requested by the Support Services Change Control Coordinator.</li></ol>

## **Procedures**

### **4. Overview**

This section provides specific information and step-by-step instructions on how to perform key procedures that support the change control process. The following procedures are covered:

1. Verifying and validating entry criteria.
2. Initiating a change request.
3. Performing an impact analysis.
4. Developing a planning package.
5. Reviewing the change request package.
6. Opening a change request with the COMPANY Global Change Control Board (GCCB).
7. Performing a post-implementation validation.

#### **4.1 Verifying and Validating Entry Criteria**

Verifying entry criteria entails checking all entry criteria against a list of minimum requirements and ensuring that all checkpoints and artifacts (items) are accounted for and complete. Validating entry criteria involves checking the accuracy of each item or checkpoint and assuring that the criteria conforms to specifications and/or established standards.

The following table sets forth the minimum requirements for entry criteria before a change request can be initiated, and the specifications and/or standards to which the entry criteria must conform.

Verification		Validation
Checkpoint or Item	Comments	Specification/Standard
Release Notes	Required for all software configuration items that are new versions, revisions, add-on modules, etc. For patches see <i>Build Analysis</i> below.	Minimum requirements: <input type="checkbox"/> List of items contained in the release (software, scripts, documentation, media) <input type="checkbox"/> Description of release specifications (basis for the release, to which specifications was the release built) <input type="checkbox"/> List of fixes/features [ <b>optional: fixes &amp; features associated with issue number from the issue tracking system</b> ] <input type="checkbox"/> Installation steps, requirements and impacts <input type="checkbox"/> Test results including open and closed issues in the release <input type="checkbox"/> <b>Optional: additional information/instructions needed to implement and/or support the release.</b>
Build Analysis	Required for all patches. Can be used in conjunction with or in lieu of release notes (see above).	<input type="checkbox"/> Nature of the patch: emergency (Sev 1), Urgent (Sev 2) or Routine (user-requested) <input type="checkbox"/> List of items contained in the patch (software, scripts, documentation, media) <input type="checkbox"/> Description of patch specifications (basis for the release, to which specifications was the release built) <input type="checkbox"/> List of fixes/features [ <b>optional: fixes &amp; features associated with issue number from the issue tracking system</b> ] <input type="checkbox"/> For COMPANY-Developed software: SCM log showing build objects [ <b>Mandatory after SCM has been implemented</b> ] <input type="checkbox"/> Patch unit and integration test results
Installation Documentation	Required for all hardware implementations and upgrades and new applications, utilities and add-ons. Can be used in conjunction with release notes for software supporting hardware implementations (drivers, front-end utilities, diagnostics, etc.)	<input type="checkbox"/> Manufacturer-provided installation documentation <input type="checkbox"/> Any errata sheets <input type="checkbox"/> All media and other materials referenced in the installation documentation <input type="checkbox"/> Check all included media for README files or other forms of release notes, errata or special instructions <input type="checkbox"/> Inventory that accounts for all actual hardware components to be installed/implemented if applicable

## Minimum entry criteria requirements (continued)

Verification Checkpoint or Item	Comments	Validation Specification/Standard
Test Results	Required for all hardware and software configuration items (see <i>Glossary of Terms</i> )	<ol style="list-style-type: none"> <li><input type="checkbox"/> Test Plan used to perform UAT (see <i>Glossary of Terms</i>) [<b>Applicable to new products and versions</b>]</li> <li><input type="checkbox"/> Test cases executed in accordance with the UAT test plan showing <input type="checkbox"/> expected results <input type="checkbox"/> observed results <input type="checkbox"/> traceability of features to functional requirements specifications <input type="checkbox"/> test summary listing discrepancies</li> <li><input type="checkbox"/> Test Plan used to perform pre-production testing (see <i>Glossary of Terms</i>) [<b>Applicable to all software and hardware configuration items</b>]</li> <li><input type="checkbox"/> Test cases executed in accordance with the pre-production test plan showing <input type="checkbox"/> expected results <input type="checkbox"/> observed results <input type="checkbox"/> traceability of fixes to technical requirements specifications <input type="checkbox"/> test summary listing discrepancies</li> </ol> <p><b>Optional:</b></p> <ol style="list-style-type: none"> <li><input type="checkbox"/> Operational readiness test plan and associated test cases (applicable to new systems)</li> <li><input type="checkbox"/> Load and capacity test plan and associated test cases when required by operational requirements. Test cases will show <input type="checkbox"/> expected results <input type="checkbox"/> observed results <input type="checkbox"/> traceability of test parameters to operational requirements (response time, stress loading, transactions-per-second, etc.) <input type="checkbox"/> test summary listing discrepancies</li> </ol>
Operational Requirements	Required for all changes that (1) impose, or have the potential to impose, an extended maintenance window, (2) affect, or have the potential to affect, meeting service level objectives, (3) have a high risk of failure	<input type="checkbox"/> Service level agreement with defined service level objectives for the application or system to which the change will be made. <input type="checkbox"/> Risk assessment of the proposed change. See <i>Performing an Impact Analysis</i> below.

**Note:** In the event of an emergency change that is severely business-impacting and time-critical, a waiver of entry requirements can be granted. The waiver must be jointly issued in writing by both the application owner and business system manager.

## 4.2 Initiating a Change Request

The initiator (typically the application BSM or BSA by direction of the BSM) will:

1. Verify and validate all entry criteria.

2. Complete the Identification and Classification section of the Support Services change request form (item ❶). This form is the Implementation Plan of Action and Milestones that is an appendix to this document.
3. Proceed to the Impact Analysis step.

### 4.3 Performing an Impact Analysis

#### 4.3.1 Dependency Analysis

Evaluate dependencies as follows:

IF THE CHANGE:	THEN:	CHANGE CATEGORY
Affects systems or applications external to the system or application to which the change will be implemented.	Contact the technical owner of the system or application to determine if the change will have any impact on the external system. If change is determined to have an impact, than external system/application technical owner will provide list of activities required to support the implementation and post implementation validation plan for external system PIV. <b>Include this information in the implementation plan.</b>	<p><b>Category 2</b> if the implementation will not require active involvement of external system(s) technical owner(s).</p> <p><b>Category 3</b> if the implementation will require active involvement of external system(s) technical owner(s).</p>
Will require an extended maintenance window.	Inform the BSM who will negotiate the scheduled start and completion times with the application owner. Base the implementation schedule on the negotiated schedule. <b>Peer-review the schedule to ensure that the negotiated schedule is realistic.</b>	<b>Category 3.</b>
Is complex and it is uncertain whether or not that the maintenance window is sufficient to perform all steps associated with the implementation.	Inform the BSM who will inform the application owner of the potential risks and determine the worst-case start and completion times. <b>Compute the total time required to roll-back and establish a go/no-go decision point in the implementation plan based to the latest point in the implementation which the change can be rolled-back and still meet the maintenance window.</b>	<b>Category 2.</b>



### 4.3.2 Risk Assessment

After all dependencies are known they are factored into a risk management plan. The risk management plan consists of:

1. Identification of risks.
2. Determination of probability and impact of risks (risk factor).
3. Plan to eliminate or mitigate the risks.

A risk represents a condition that is subject to causing a failure or unexpected result. For example, if a server has a single disk drive, it is *exposed* to the possibility that if the drive fails data could be lost, and the system will not function until a replacement drive is installed. Preventing the risk associated with this particular exposure could be accomplished by using mirrored drives, whereby data is written to both drives simultaneously and a failure to either drive would not result in the loss of data or system availability.

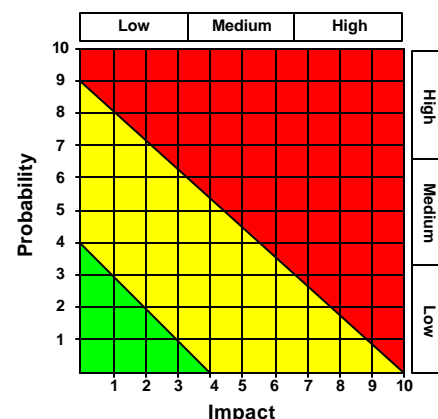
Other sources of risks include (but are not limited to):

1. A new technology, method or process.
2. Resource constraints or lack of effective coordination.
3. Uncertain estimating baselines.

After risks have been identified, the next step is to determine both the probability of it occurring, and the impact it will have on the change implementation.

Determining probability need not be an exercise in mathematics - in many cases the SME's past experience will indicate whether the probability will be high, medium or low. In the matrix on the right, a low probability falls within the range of 0 to 30%; medium, 31-65%; and high, from 66-100% possibility of occurring.

A high probability of occurrence does not necessarily mean that the risk is significant. The true significance of a risk, called the *risk factor*, is derived by multiplying the probability by the impact of the risk on the project. For example, if the risk of losing a disk drive in a drive array every 6 months is 80% probable (high), but the impact is 2 (low), then the risk factor is 1.6 (the product of multiplying .8 by 2), which is also low. In this case the risk does not warrant much attention and would probably not rank among the project risks that require measures to avoid or mitigate. Conversely, if the same disk



drive was used without the fault-tolerance of an array and the impact was deemed to be 9 (high), then the risk factor will be a probability of .8 multiplied by 9, which equals 7.2 (high). This risk would need to be factored into the implementation plan.

Risk level is a required field in the Global Change Control change request, as is a risk assessment and impact level determination. The following table provides guidelines for computing risk and impact levels that are consistent with the Global Change Control request form:

Condition	Risk Level
Involves external systems	20-50%
Complex	20-50%
Service level(s) are at risk	20-30%
Change involves DDL (data definition language) scripts to create, delete or modify database objects	10-20%
There is no direct experience with this type of change (including the underlying technology)	40-50%
Roll-back is difficult or approximately the same number of steps as the implementation	40-50%
Roll-back requires complete application recovery (from tape, DLT, etc.)	30-50%
Roll-back requires complete recovery of database	30-50%
New, unproven technology is employed	20-50%
Source of change (internal development or ISV) does not employ basic quality assurance methods (configuration management, unit and integration testing, etc.)	30-50%
Routine maintenance/changes with no predicted user impact (background changes that do not directly affect performance or availability)	01-05%
Routine maintenance/changes that involves copying files (possible overwrite in the wrong direction)	05-20%

#### 4.3.3 Additional Factors

Also note that certain change classifications and categories have inherent risks that need to be considered when assessing the overall risk and impact levels of a change.

Change Classification or Category	Factors to Consider
<b>Urgent</b>	Possible lack of detailed planning and preparation due to short interval implementation schedule.
<b>Emergency</b>	Compressed implementation timeframe and testing completeness/adequacy
<b>Category 2</b>	Inaccurate estimates for rolling back a change - go/no-go checkpoint insufficient to fully roll-back change and still remain within the maintenance window.
<b>Category 3</b>	Excessive implementation schedule compression to minimize time maintenance window is exceeded.

In addition to the above factors, the cost of downtime needs to be taken into account when determining the overall impact of a risk. A business-critical application with 500 users can cost thousands of dollars *per minute* of unscheduled downtime, which would warrant assigning a high impact. Conversely, a system with 500 users who only occasionally use the system or switch to manual methods with a minimum loss of productivity or ability to support business operations will have a much lower cost of associated downtime, which merits assigning a lower impact rating. The true impact of a risk will be determined by the application owner and BSM.

***No change with an associated risk impact greater than .5 (50%) will be implemented until a formal review of the implementation plan has been conducted. The Support Services Manager will appoint the review team and will have final approval authority for any change that exceeds this threshold.***

#### 4.4 Developing a Planning Package

The planning package consists of the following sections:

1. Implementation Plan of Action and Milestones.
2. Roll-Back Plan.
3. Escalation Plan.
4. Communications Plan.
5. Results.
6. Post-Implementation Review.

##### 4.4.1 Implementation Plan of Action and Milestones

This section of the planning package is divided into four phases: Planning, Tasks, Validation and Release to Production/Exit. Each phase is identically formatted as follows:

Task	Scheduled Start Date & Time	Scheduled Complete Date & Time	Responsible	Expected Result	Quality Gate or Communication Checkpoint	Comments

Most of the above columns are self-explanatory. The column titled *Quality Gate or Communication Checkpoint* is used to denote tasks that either have an associated quality gate or a communication checkpoint (see Communications Plan). The illustration on the right depicts example quality gate and communication checkpoints.

Quality Gate or Communication Checkpoint	Comments
Notify on-line users that application will go off-line	Communications Checkpoint #2 - page out and e-mail
Compare Pre-production and production databases	Quality Gate # 4 - estimated time to complete: 45 minutes

The *planning phase* encompasses all activities associated with preparing for the implementation, including such tasks as performing back-ups, disabling logins, and pre-implementation notifications.

*Task phase* is the sequence of tasks to perform the actual implementation after the all preparatory steps in the *planning phase* have been completed.

Tasks associated with the *validation phase* include such activities as initiating a post-implementation validation and performing any other tests or quality assurance functions that need to be conducted in order to ensure that the system or application is ready for release to the production environment. In most cases all of the tasks in the *validation phase* will be either a quality gate or communication checkpoint.

The *release to production/exit phase* tasks include phase II post-implementation validation (see *Performing a Post-Implementation Validation* below), any additional quality gates that must be successfully passed before release to production, and communication checkpoints, such as notification that the implementation is complete and users can login.

**Quality Gate** - a checkpoint for verification and/or validation. Implementation cannot proceed until the checkpoint has been successfully passed. If a quality gate fails an escalation will be triggered. If a quality gate cannot be successfully passed in a timely manner (to be determined by decision makers named in the Escalation Plan), or if the quality gate is a go-no go decision point that fails, back-out and roll-back procedures will be executed if so directed by decision makers to whom the issue has been escalated.

#### 4.4.2 Roll-Back Plan

The roll-back plan section describes how the implementation will be backed-out if there is a problem, such as failing to meet quality gate criteria to move forward in the implementation, or unsuccessful post-implementation validation.

The roll-back plan section is depicted in the following illustration.

## Roll-Back Plan

Task	Estimated Time to Complete	Responsible	Comments
Post Roll-back Validation			
Total Time To Roll-Back			
Actual (HH/MM [AM /PM] Time Roll-back Needs to Occur to Meet Maintenance Window/SLO Requirements)			

Key elements of the roll-back plan are:

1. Tasks - sequence of steps to back-out the implementation.
2. Estimated time to complete - how long will each task take.
3. Responsible - person identified to perform the task.
4. Comments - examples: responsible person's pager number, notes regarding probability of success for completing the task and/or additional contingencies.

5. Total time to roll-back - the sum of all task estimated times to complete.
6. Actual time roll-back needs to occur

The roll-back plan can also be multi-part to cover more than one condition. For example, a roll-back plan might address:

- problems encountered during the implementation, such as inability to successfully pass a quality gate.
- problems that were not detected during the post-implementation validation, but so severely degrades the application or system in production that a roll-back is required.

A worked example will show how a roll-back plan is developed. In this example the roll-back plan will address problems encountered during the implementation, such as inability to successfully pass a quality gate.

In this case the BSM (or BSA) developing the implementation plan develops a task list and how long each task will take. The hypothetical tasks in this example are backing out code and database changes, restarting the application, re-establishing interfaces and a communications checkpoint. The first step in the roll-back plan [in this example] will be to determine what tasks are required to back-out the code changes:

1. Bring down the application.
2. Delete implemented code from production locations.
3. Copy pre-implementation code from archive to production locations.
4. Perform checksum on the restored code and compare to checksum performed prior to implementation.

Next, the tasks to back-out the database changes are determined:

1. Disable triggers and constraints for database objects affected by the release implementation
2. Drop database objects modified during implementation.
3. Copy pre-image of objects into production schema.
4. Create indexes, constraints and triggers on restored database objects.
5. Perform comparison between restored objects and list generated prior to implementation.

Finally, the tasks to re-establish interfaces are listed:

1. Launch communication daemons.
2. Perform a test transaction.

The two remaining tasks will be to perform a post roll-back validation and enable logins and notify stakeholders.

After all of the tasks necessary to roll-back have been determined the estimated time to complete each task is made. One effective way to accomplish this is to lay the roll-back out on a timeline worksheet, such as the following example:

		Bring down application	Backout code	Checksum		Bring application up & Re - establish Interfaces	Enable Logins
SA		.75	.5	.5		1.0	.2
DBA	.2	5.8 ± Restore tables from copy				2.0	
	Disable triggers & constraints / Drop M4 objects					Create indexes, enable constraints & triggers & compare restored objects to pre- implementation listed objects	

The above worksheet yields the following estimates:

Back-out code changes	1.75 hr
Back-out database changes	8.0 hr
Restart application	0.75 hr
Re-establish interfaces	0.25 hr
Perform post-roll-back validation	1.0 hr
Enable logins/notify stakeholders of application availability	0.2 hr

This totals 11.95 hours to roll-back the implementation. The resulting roll-back plan will look like the one illustrated below:

**Roll-Back Plan**

Task	Estimated Time to Complete	Responsible	Comments
Bring down the application	0.75 hr	System admin John Doe	
Delete implemented code from production instances	0.25 hr	System admin John Doe	
Copy pre-implementation code from archive to production instances	0.25 hr	System admin John Doe	
Perform checksum on the restored code and compare to checksum performed prior to implementation	0.5 hr	System admin John Doe	Quality Gate
Disable triggers and constraints for database objects affected by the release implementation	0.1 hr	DBA Susan Quares	
Drop database objects modified during implementation	0.1 hr	DBA Susan Quares	
Copy pre-pro image of objects into production schema	5.8 hr	DBA Susan Quares	
Create indexes, constraints and triggers on restored database objects	1.0 hr	DBA Susan Quares	
Perform comparison between restored objects and list generated prior to implementation	1.0 hr	DBA Susan Quares	Quality Gate
Launch communication daemons	0.2 hr	System admin John Doe	
Perform a test transaction	0.05hr	System admin John Doe	Quality Gate
Post Roll-back Validation	1.0 hr	System admin John Doe performs application validation, DBA Susan Quares perform database validation, ASA Jack Nantle performs validation per post-implementation validation plan	Quality Gate
Enable logins/notify stakeholders	0.2	System admin John Downes enable logins - implementation Jane Smith supervisor notify stakeholders	Communications checkpoint
<b>Total Time To Roll Back</b>	<b>11.95 hr</b>		

The final step in developing the roll-back plan is to compute the time the roll-back needs to occur to meet maintenance window or service level objective requirements, which is the final block on the roll-back plan template. This time is the "drop dead" point in the implementation, which is the latest time a roll-back can be initiated and still make the application or system available to users

Total Time To Roll-Back	11.95 hr
Actual (HH/MM [AM   PM]) Time Roll-back Needs to Occur to Meet Maintenance Window/SLO Requirements	

within the maintenance window or applicable service level objectives for availability. In the example used, assume that the application is only used Monday through Friday from 7:00 AM to 5:00 PM, and all users are in the same time zone. In this example the maintenance window is Monday through Thursday is 5:30 PM until 6:30 AM, (13 hours), and 5:30 PM Friday through 11:00 PM Sunday (53.5 hours). If the implementation was scheduled to start at 5:30 PM on a Monday through Thursday the actual time that the roll-back needs to occur to meet maintenance window or service level objectives would be 6:33 PM. This is because the total time needed to roll-back is 11.95 hours, and the maintenance window is 13 hours in length. That means that any roll-back needs to commence 1.05 hours (1 hour/3 minutes) after the implementation has started in order to stay within the maintenance window and/or meet service level objectives for availability. For routine maintenance the most obvious time to schedule this example implementation would be during the 53.5 hour weekend maintenance window; however, an emergency change may not allow flexible scheduling. In such cases an extended maintenance window or waiver for meeting service level objectives needs to be negotiated between the BSM and application owner.

The roll-back plan is one of the most important elements of the change request package because it:

1. Ensures that the tasks and times associated with a roll-back have been carefully examined, documented and incorporated into the implementation plan.
2. Provides the implementation team with a methodical, structured approach to rolling back a change, versus a reactive, unprepared attempt to restore an application or system to its previous state by an implementation team that is under pressure with no plan or checklist.
3. Evidences due diligence on the part of the implementation planner with respect to all facets of the implementation.



#### 4.4.3 Escalation Plan

The escalation plan identifies the roles and responsibilities of the problem management team who will be notified in the event of a situation that threatens the successful completion of an implementation.

There are three levels of escalation:

1. Requirement to deviate from an implementation plan, or an unexpected result.
2. Roll-back.
3. The need to involve higher level management, such as a catastrophic failure like not being able to complete a roll-back, severe impact on business operations or final approval authority for go/no-go decisions.

The template for the escalation plan is illustrated below:

Condition	Name/Group	Fluor Phone #	Pager #	Home Phone/Cell #	Comments
Deviation From Implementation Plan or Unexpected Result					
Roll-Back					
Management Escalation					

#### 4.4.4 Communications Plan

The communications plan portion of the change control package identifies all communications checkpoints that will occur during the implementation, to whom the communications are sent, and who is responsible for sending them.

There are seven identified announcement types, with standard announcement text in the communications plan template, and space to include additional announcements tailored to the implementation plan.

Announcement Type	Addressees	Event or Date/Time to Make Announcement	Responsible for Sending	Announcement Text
<b>Stakeholder Notification of Change</b>	<ol style="list-style-type: none"> <li>1. Application Services Manager</li> <li>2. Help Desk Services</li> <li>3. MicroAge</li> <li>4. Support Services Application Support Coordinator</li> <li>5. BSM</li> <li>6. Application owner</li> <li>7. End users</li> <li>8. ASA</li> <li>9. Other identified stakeholders</li> </ol>	<b>Planned Maintenance/Normal Maintenance Window.</b> Announcements will occur as follows: <ol style="list-style-type: none"> <li>1. Upon approval by Change Control Board</li> <li>2. 1 working day prior to implementation</li> <li>3. 1 hour prior to implementation</li> </ol> <b>Fast Track</b> <ol style="list-style-type: none"> <li>1. As soon as possible after review &amp; approval</li> <li>2. 1 hour prior to implementation</li> </ol> <b>Emergency</b> <ol style="list-style-type: none"> <li>1. As soon as possible - target = at least an hour before implementation commences</li> </ol>	<b>Planned Maintenance/Normal Maintenance Window</b> <ol style="list-style-type: none"> <li>1. Change Control Coordinator</li> <li>2. Change Control Coordinator</li> <li>3. Implementation Supervisor</li> </ol> <b>Fast Track</b> <ol style="list-style-type: none"> <li>1. Change Control Coordinator</li> <li>2. Implementation Supervisor</li> </ol> <b>Emergency</b> <ol style="list-style-type: none"> <li>1. Implementation Supervisor</li> </ol>	What is the change? Why make the change? What date/time is the change planned to be implemented? What will be the impact of not making the change at the requested time? What will be the impact after the change? What actions do you (recipient of the message) need to take? Contact name and phone number for additional information.
<b>Stakeholder Notification of Checkpoints</b>	<ol style="list-style-type: none"> <li>1. Application Services Manager</li> <li>2. Help Desk Services</li> <li>3. MicroAge</li> <li>4. Support Services Application Support Coordinator</li> <li>5. BSM</li> <li>6. Application owner</li> <li>7. ASA</li> <li>8. Other identified stakeholders</li> </ol>	In accordance with implementation plan	Implementation Supervisor	Expected and observed results at checkpoint

#### 4.5 Reviewing the Change Request Package

The change request package will be reviewed and approved by the application owner and BSM. The application owner will ensure that scheduled start and completion times are acceptable and that all noted risks (including excessive roll-back time requirements) are acceptable.

The BSM will perform a technical review of the implementation plan, and will negotiate any extensions to the maintenance window or waiver for meeting service level objectives with the application owner.

After the change request and implementation plan comprising the change request package has been reviewed and approved by the application owner and BSM it is sent to the Support Services Change Control Coordinator for final review. The Support Services Change Control Coordinator will ensure that all entry criteria has been met, the change request package conforms to policy and procedures, and that all information necessary to evaluate and approve the change is included.

An approved change request package will be reviewed by the Support Services Change Control board convened to review all change requests in conjunction with applications supported by Support Services. This is a due diligence checkpoint that is performed before the change request package is submitted to the Global Change Control Board (GCCB) for final review and approval.

#### 4.6 Opening Change Request with COMPANY GCCB

After the change request package has been reviewed and approved by the Support Services Change Control Board the Support Services Change Control Coordinator will open a change request in accordance with GCCB policy and procedures. The GCCB change request number will be entered into the Global Change Control # block (see illustration).

Upon review and approval by the GCCB, the implementation proceeds in accordance

## Support Services Change Control Request & Implementation Plan

**Change Request**

**Identification and Classification**

Planned Maintenance/General Maintenance Work: ☐ Fix/Track - Incident or problem requires investigation and/or opportunity exists to prevent future incidents Global Change Control Board meeting 24 hours in advance

Emergency Change - Restoration of Service: ☐ Immediate Need

Date:  Work Order #:  Service Request #:

Application Name:

Change Category:

☐ 1 - Information Services and Systems modification or enhancement (e.g., new software, new equipment, new data storage medium, etc.)

☐ 1.1 - Service Level or Risk or Affairs External Systems/Applications: Task or a major or new task or a new event

☐ 2 - Service Level Affairs: ☐ 2.1 - Service Level Contribution with Owner(s) of Internal Systems(s): Example: ...

Global Change Control #:

with the schedule set forth in the implementation plan.

## 4.7 Implementation Results

The *Results* block is a key component of the COMPANY Support Services change request package. It is in this block that the implementation supervisor records a synopsis of the implementation, which serves as the basis for any required process improvement initiatives and as a knowledge base for future implementations. The following illustration shows the information that is recorded:

Results	
Planned start date/time:	Actual start date/time:
Planned completion date/time:	Actual completion date/start time:
Maintenance window <input type="checkbox"/> was not <input type="checkbox"/> was exceeded. Reason(s) if exceeded:	
List of observed results that deviated from expected results (including roll-back) and reason(s):	

It is important to note reasons for any variances between planned and actual start and completion date/times, and whether or not the maintenance window was exceeded.

The definition of *maintenance window* in the context of the *Results* block is the actual maintenance window specified in the application/system service level agreement. Therefore, even if an extended maintenance window was negotiated with the application owner in advance of the implementation, the normal maintenance window is the basis for determining whether or not the maintenance window was exceeded. The objective is to establish a baseline for any future implementations that are similar to the one being documented, which will allow the implementation planner to plan for an extended window. Another reason is to determine if there are opportunities for reducing the time it takes to complete similar implementations.

Also included in the *Results* block is a list of any observed results that deviated from expected results during the implementation. For example, if a task requires a database update such as the following example:

```
set echo on
spool msisdn_status_r2y.log
update directory_number
set dn_status = 'y'
where
    dn_status = 'r'
    and
    substr(dn_num, 7, 4) < '9000'
    and
    substr(dn_num, 1, 3) not in ('301', '304')
commit;
/
```

In the above example the SQL\*Plus script updated every phone number in a database table named "directory\_number" by changing all dn\_status attributes that were set to "y" to a value of "r" if the phone number's last four digits were less than 9000 and did not have an area code of 301 or 304. The expected result from running a validation query (typically an implementation plan quality gate) would be that all values that were set to "r" were, in fact, set to "r". Executing a validation query such as

```
select (count) from directory_number
where
    dn_status = 'r'
    and
    substr(dn_num, 7, 4) < '9000'
    and
    substr(dn_num, 1, 3) not in ('301', '304');
```

which counted all records matching the script parameters that were still set to "y" should return 0. Any other result is unexpected would be documented in the list of observed results that deviated from expected results.

Unexpected results from any task or quality gate are used to determine root cause of the problem, fine tune future implementation plans and detect opportunities for process improvement.

## 4.8 Performing a Post-Implementation Validation

The final validation checkpoint before releasing a change into the production environment is to perform a post-implementation validation. This implementation quality gate has two phases:

1. Verifying that all major functions, systems and subsystems of an application or system have no apparent defects after the change; i.e., the application or system performs as expected.
2. Ensuring that the system is stable in the production environment.

The post-implementation validation is a *test process*, which consists of the following elements:

### 4.8.1 Process Summary

Purpose: identifies the process owner (the application ASA or system owner), a stated purpose (to define post-implementation test coverage of major application functions, and to exercise any interfaces and subsystems), and a trigger (planned or

PROCESS SUMMARY	
Process Owner	<Fluor Signature Services ASA>
Purpose	Define post-implementation validation (PIV) that ensures test coverage of major application functions, exercises interfaces and subsystems
Trigger	Planned or Emergency maintenance action/patch promotion

emergency maintenance actions).

### 4.8.2 Controls

Purpose: governs the process, such as policies and procedures, maintenance window, implementation planning documents, etc.

CONTROLS
(1) FSS Support Services Change Control Policy and Procedures PIV definition; (2) <Application Name> Maintenance Windows; (3) Permission to perform Emergency Maintenance; (4) FSS Support Services and GCCB Change Control Implementation Planning Documents; (5) FSS Support Services and GCCB Change Control Policies, Procedures and Processes

### 4.8.3 Entry criteria

Purpose: specifies what conditions must be met before the post-implementation validation can begin.

ENTRY CRITERIA
<input type="checkbox"/> FSS Change Control Implementation Plan (including back-out plan)
<input type="checkbox"/> Approved GCCB Change Control Ticket ID
<input type="checkbox"/> Identification of Systems, subsystems and components affected by maintenance
<input type="checkbox"/> Completed maintenance action
<input type="checkbox"/> Tailored test plan to exercise systems, subsystems and components affected by maintenance/patch

#### 4.8.4 Tasks

Purpose: lists the actual validation steps to be performed during the PIV.

Phase I consists of regression test steps and a series of validation steps that are tailored to test the specific changes made to the application or system.

Phase II is post-implementation monitoring to ensure that the application or system is stable in the production environment. The illustration is a hypothetical example of PIV Phase I tasks.

The Phase II tasks shown in the example are recurring for any PIV.

TASKS
PIV Phase I:
•Make credit card payment in the amount of \$0.01 against an account
•Perform Credit Evaluation on an account
•Make test call and view unbilled usage
•View bill image, account notes, access services and configured services, perform account balance estimate
•Exercise systems, subsystems and components that were affected by maintenance
•Perform SMS query for period commencing with system restart
• Stakeholders notified and system turned over to users
⑤ PIV Phase II
•Remain on-site for 1 hour after system is turned over to users to ensure no problems occur during normal operation

#### 4.8.5 Validation

Purpose: provides a checklist that ensures that all tasks return expected results, and that no problems occur during the first hour after the system has been turned over to production users.

VALIDATION
<input type="checkbox"/> All tests return expected results
<input type="checkbox"/> No problems occur during first hour after system is turned over to users

#### 4.8.6 Exit criteria

Purpose: lists conditions that must be met before the application or system is formally released into the production environment.

EXIT CRITERIA
<input type="checkbox"/> No Sev 1 or 2 issues result from maintenance
<input type="checkbox"/> System operational and functional status has been ascertained
<input type="checkbox"/> Test results have been updated in Change Control tickets
<input type="checkbox"/> Implementation Supervisor notified of completion of Phase I and II PIV.

#### 4.8.7 Constraints

Purpose: lists conditions that impede the ability to perform the post-implementation validation, such as lead time to tailor a PIV to a specific patch, release or change, gaps in documentation, etc.

CONSTRAINTS
(1) Time to develop complete and accurate test plan tailored to maintenance/patch release for emergency maintenance; (2) timeliness and completeness of FSS Change Control implementation plan; (3) gaps in <Application> technical documentation regarding details on file systems, system and subsystem hosting, and dependency analysis.



#### 4.8.8 Critical Success Factors/Key Performance Indicators

Purpose: critical success factors are the factors necessary to successfully meeting all process objectives (in the case of post-implementation validation the factors are cycle time, defect threshold and accuracy)

Key performance indicators are metrics against which critical success factors are measured. The following depicts the critical success factors and their associated key performance indicators for PIV: cycle time is the system made available to end users not later than 30 minutes after PIV commences; i.e., the PIV will be completed in 30 minutes or less. The key performance indicator for defect threshold is no severity 1 or 2 issues, and for accuracy is that all defects are caught in PIV.

CRITICAL SUCCESS FACTORS		KEY PERFORMANCE INDICATORS	
<input checked="" type="checkbox"/> Cycle Time	<input checked="" type="checkbox"/> Accuracy	<u>Phase I Test Completion (Cycle Time) - Reduce lag between maintenance completion and operational support of business processes</u>	<u>&lt;= 30 after notification that system is on-line</u>
<input checked="" type="checkbox"/> Defect Threshold	<input type="checkbox"/> N/A	<u>Post Test Errors (Defect Threshold and Accuracy)</u>	<u>No Sev 1 or 2 issues related to maintenance after PIV complete</u>
<input type="checkbox"/> Units per Time Period			

A copy of the Post-Implementation Validation worksheet is provided as an appendix to this document.

## 4.9 Post-Implementation Review

This is an optional step in the change control process, and is typically not performed when the implementation was minor (changing a script, adding a range of IP addresses to a firewall access control list, etc.) *and there were no problems during the implementation*. If there *were* any problems during the implementation, such as unexpected results, exceeding a maintenance window, or a roll-back, the post-implementation review is mandatory and will be convened by the Support Services Manager. The post-implementation review portion of the change control package is depicted in the following illustration:

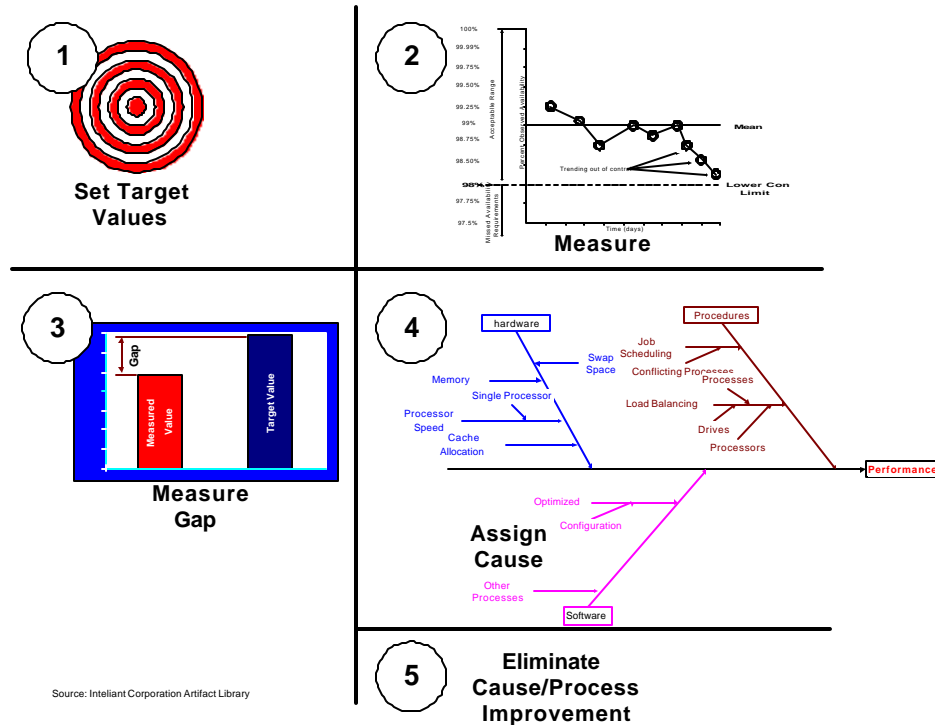
Post-Implementation Review	
A post-implementation review meeting <input type="checkbox"/> will <input type="checkbox"/> will not be held:	
Date:	Time:
Location and/or conference dial-in #	
Mandatory Attendees:	
Purpose: To review the implementation	
Agenda:	
1. Actual time to complete implementation vs. planned time.	
2. Discrepancies between expected and observed results.	
3. Lessons-learned.	
4. Root cause analysis and elimination and/or process improvement plans [if applicable]	
Length:	
The post-implementation review is not expected to exceed _____ minutes.	

The post-implementation review will be followed-up with meeting notes. A copy of these notes will be provided to all attendees, persons who were assigned action items during the meeting, and any other stakeholders that the Support Services Manager determines should receive them. The format for the meeting notes are shown below:

Post-Implementation Review Meeting Notes		
Attendees:		
Synopsis:		
Action items:		
Action	Responsible	Due By

## 4.10 Root Cause Analysis and Process Improvement

The following is a generic guideline for performing root cause analysis and effecting process improvement in connection with implementation events that either failed or were marginal to the point that process improvement is warranted.



Process improvement is based on gap analysis, which is performed using the following steps:

1. Determine target values; i.e., key transaction response times, time-to-resolution, capacity metrics, etc.
2. Measure performance or characteristics - for characteristics/processes that are not in statistical control, appropriate techniques (bar charts for comparison, line charts for trend analysis, or a combination of the two) should be used. For characteristics/processes that have been refined and are under statistical control, consider using control charts.
3. Detect gaps as differences between measured values and target values, or non-random data points on a control chart.
4. Assign cause by performing a root cause analysis using cause and effect diagrams, fishbone diagrams, failure mode and effect analysis or other standard methods.
5. Develop and execute a plan of action for eliminating root cause and/or improving characteristic (quality) or process.

Note that step 4, perform a root cause analysis, can be accomplished independently of process improvement steps. This step is most appropriate for events such as roll-backs or implementations that took longer than anticipated.

## **Attachments**

**Support Services Change Control Request & Implementation Plan**

**Post-Implementation Validation (PIV) Worksheet**