Configuration, Change, and Release Management Policies and Procedures Guide

Table of Contents

Section 1 Introducing Configuration, Change, and Release Management

This section provides a brief introduction to policy and procedures for configuration, change, and release management. By reading this introduction, you will gain a sense of how these policies and procedures apply to you.

1.1 Overview of Configuration, Change, and Release Management

The combined configuration, change, and release management approach provides a set of policies, processes and procedures for information systems. The policy is designed to preserve the integrity and stability of the information systems and to manage their life cycles.

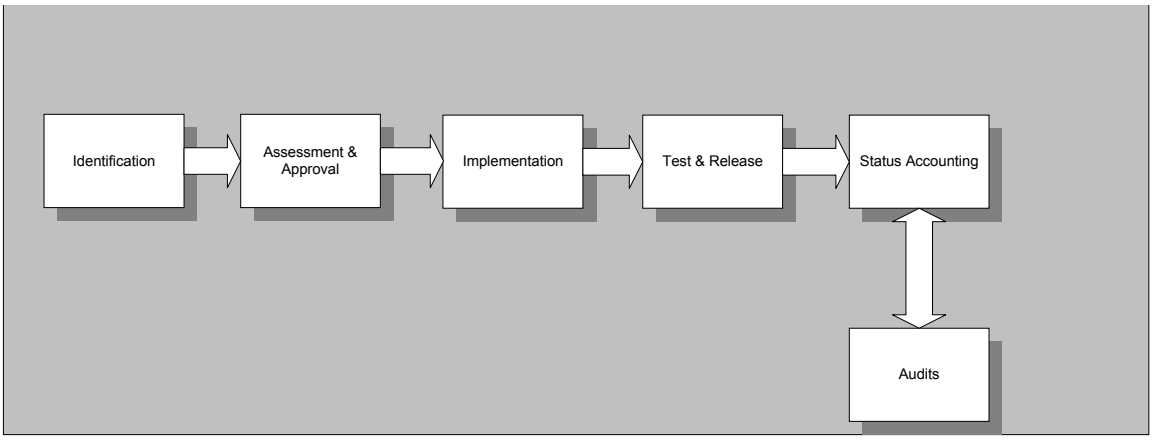
Configuration, change, and release management involves five processes to identify, assess/approve, implement, test/release, and account for all changes to an information system. A commitment to this process will produce fully documented and managed changes to the systems on which the business is built.

The key topics of this section include:

- Objectives - A discussion of the processes, owners, and objectives
- Audience - A description of persons subject to the policies
- Usage - A flow chart depicting the overall process

Figure 1.1 illustrates the configuration, change, and release management process in the environment.

Figure 1.1 Configuration, change, and release management process



1.2 Objectives

Configuration, change, and release management are a set of related processes that will achieve the following objectives:

1. Maintain a detailed record of each system's configuration. This means tracking down to the patch and revision level of individual software and hardware modules, components and subsystems.

2. Control changes to systems. This includes a formal process for change requests, assessment, approval, implementation, test and release.

3. Ensure an audit trail. This includes change requests, approvals, test results, installation/deployment dates, and post installation quality assurance tests to support the system's operational baseline configuration documentation.

4. Facilitate life cycle management and operational consistency. This is verified by auditing installed systems against baseline configuration documentation.

1.3 Audience

The audience for this configuration, change, and release management process includes most personnel. These customers fall in to five categories, each using the configuration, change, release management process for differing purposes.

Table 1.1 lists the process owners and their particular usage of the system.

Table 1.1 Configuration, change, and release management audience.

Role	How it Applies
System users	Request a change or recommend an improvement to the system.
System vendors	Provide a change to a vendor-supported system, such as an upgrade, patch, maintenance release, or field change.
Information Technology personnel	Recommend a change or to propose an improvement such as a shell script, NDS reconfiguration, improved system configuration, increased capacity, etc.
Configuration Control Manager	Manage the process.
Service provider (such as hardware vendor, third party service providers)	Follow policy when making changes to baseline in connection with providing service; i.e., updating drivers on systems, documenting IMAC (installs, moves, adds, changes) activities, etc.

1.4 Usage

The configuration, change, and release management meta-process consists of a series of processes that flow clockwise. Five of the six processes are sequential, starting with requirements or improvements identification and ending with status accounting. The sixth process, auditing, is an ongoing process to verify that the system configuration documentation accurately reflects the system's actual configuration.

The flow chart in figure 1.2 illustrates the steps in the overall configuration, change, and release management process.

Figure 1.2 The configuration, change, and release management process.

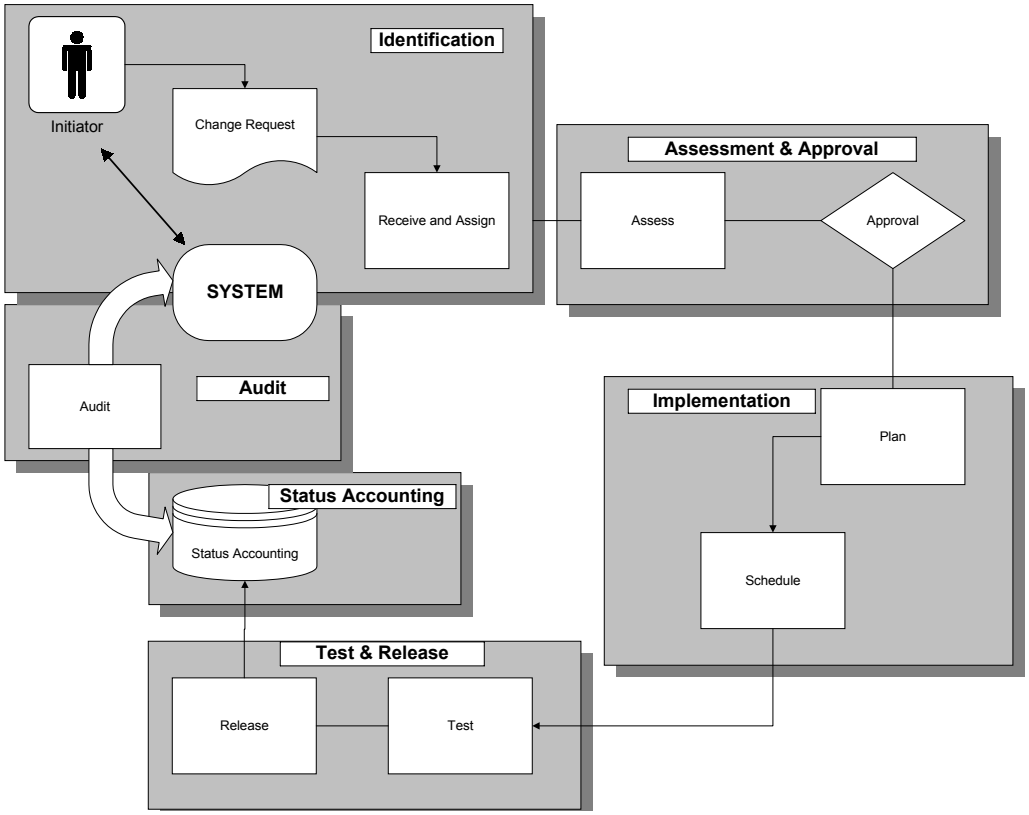


Table 1.2 describes the processes involved in configuration, change, and release management.

Table 1.2 Configuration, change, and release management processes.

Processes	Activity	Description and Form
Identification	User initiates a change request. Help Desk Services logs and validates a change request, notifies the initiator, and passes the request on to the Configuration Control Manager.	Change Request Form (Exhibit A)
Assessment and Approval	The Configuration Control Manager assigns priorities, initiates a review, compiles the findings, and forwards the findings to one of three parties. Depending on the requester, the Configuration Control Board, subject matter experts, or peer review approves or denies the requests directed to them.	Generic Checklist (Exhibit B) Generic Checklist (Exhibit B)

Implementation	If the Change Request is approved, the project is implemented. If the Change Request is denied, it is closed out and the initiators are notified.	Generic Checklist (Exhibit B); related to Sample Quality Assurance and Test Plan (Exhibit C)
Test and Release	Provides criteria for testing the system after a change has been implemented and releasing the system back into operation.	Sample Quality Assurance and Test Plan (Exhibit C)
Status Accounting	This process documents the change that has been implemented.	Generic Checklist (Exhibit B)
Audits	This process audits the configuration documentation against the actual system configuration.	Generic Checklist (Exhibit B)

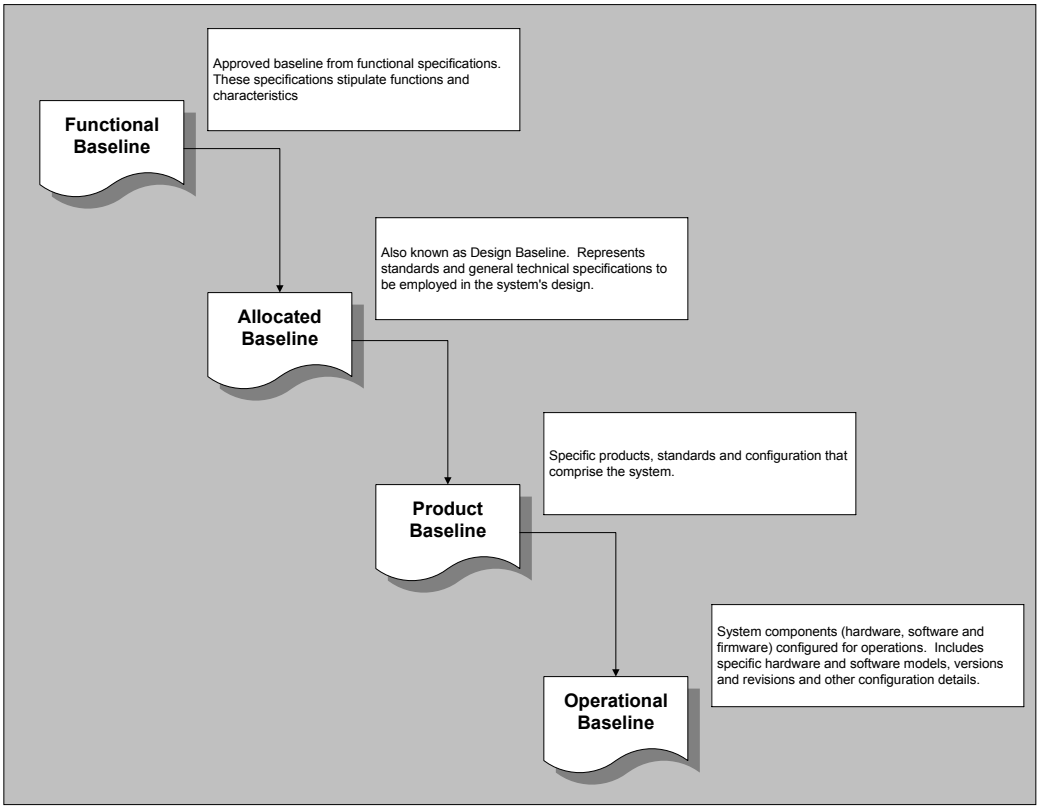
1.4.1 Establishing the Baseline

The foundation of configuration and change management is baseline identification. Release management is not directly affected by baseline identification. A baseline is a reference point that describes the state of a system at a given stage in its life cycle.

A system evolves through four distinct baselines, depending on the system's development stage. The first and second stages, functional and allocated baselines, are developed from functional specifications and designs. The third stage, product baseline, is developed from the final design or allocated baseline. The final stage, the operational baseline, reflects the system after it has been placed into production.

Figure 1.3 shows how the system baseline evolves during the system life cycle:

Figure 1.3 Baseline evolution during system life cycle.



The configuration control and change management processes and procedures set forth in this document apply to all baseline stages. Release management processes and procedures, on the other hand, apply only to the final two stages, the product and operational baselines.

1.4.2 Defining Change Request Classes

A change request is the principal means of initiating any change to a system. Change requests are divided into three classes based on who submits the request to the Help Desk. The classes are:

- End-user change requests - changes initiated by any system user or stakeholder.
- Vendor-initiated requests - changes based on vendor-supplied requirements.

- IT changes - changes for internal technical improvements by IT personnel.

The last two changes are initiated by vendors and internal IT personnel and take the form of upgrades, patches, maintenance releases, and field changes. For these changes, the process is more streamlined than the end-user change requests.

Table 1.3 defines types of changes and who would typically initiate them:

Table 1.3 Profile of change initiators.

Request Types	Functional Improvement	Additional Capacity	Additional Services	Technical Fixes	Enhance-ments	New Requirements
User Change Request	X	X	X		X	X
Vendor Initiated Changes	X			X	X	
IT Changes	X	X	X	X	X	X

Functional improvement. An improvement in the way an information system operates. End user change requests for functional improvements are typically to correct a misalignment between how an information system supports their work and the supporting business processes. Vendor and IT initiated functional improvements usually focus on functional improvements related to technology.

Additional capacity. More of a specific resource, such as shared disk space, a faster communications connection, etc.

Additional services. Range from telecommunications access to required features. Examples include: workflow-enabled applications, networked fax services, and departmental databases.

Technical fixes. Always initiated by vendors and IT to correct deficiencies or inefficiencies in products, systems, subsystems and components. A technical fix from an end user's point of view falls under the purview of problem and incident management. Example technical fixes include improved shell scripts, patches, maintenance releases, etc.

Enhancements. Added features, such as the latest version of an application, improved user interface, additional functions, etc.

New requirements. Any system, subsystem or component needed to fulfill a new requirement. Some examples include a warranty tracking system to meet a

new customer service requirement, high-speed color laser printer in response to a requirement to develop marketing materials in-house, etc.

Section 2 Policies

This section defines the policies which underlies configuration, change, and release management processes. It describes the specific policies and who owns them. The section includes background information about why the policies exist and how they relate to other IT policies.

2.1 Overview of the Configuration, Change, and Release Management Policy

The business follows a configuration, change, and release management policy to manage the life cycle of all information systems supporting business and technical objectives. As such, the processes and procedures set forth in this policy document will govern configuration, change, and release management.

The overall configuration, change, and release management policy consists of processes for managing all changes to the information systems. Understanding these processes will simplify the job of learning how to apply the configuration, change, and release management policy.

This policy consists of three interrelated elements:

- Policy statement
- Process owners
- Related policies and procedures

2.2 Policy Statement

It is policy to manage the life cycle of all information systems supporting its business and technical objectives. As such, the processes and procedures for identification, assessment and approval, implementation, test and release, status accounting, and auditing set forth in this policy document govern configuration, change, and release management. Specifically:

1. Configuration, change, and release management will begin in the functional baseline stage, the first baseline in the information system implementation cycle.

2. Before any change to a system or a baseline, the proposed change will be evaluated, approved, tested, released and documented.

3. No approved change will be implemented without:

    - An approved plan of action with milestones for implementation, showing assigned roles and responsibilities.

    - An approved test plan.

    Changes will not be released into production until they have been tested and signed off by a member of the test and quality management team and the system owner. The system owner is defined as the system administrator, business process unit (BPU), or other major stakeholder affected by the change.

4. All implemented changes will be documented by updating the system configuration documentation and all affected systems documentation.

5. At planned intervals, selected information system changes will be audited for operational consistency against the baseline configuration documentation.

2.3 Process Owners

Table 2.1 lists the process owners and their particular usage of the system.

Table 2.1 Process owners.

Process Owner	Process Owned
Help Desk Services	Identification.
Change Management Managers	Assessment, Status Accounting, Audits.
Configuration Control Board	Approval.
Test and Quality Manager	Test and Release.
As Assigned	Implementation. (Process owners for implementation will be project managers for each individual implementation project.)

2.4 Related Policies and Procedures

Table 2.2 lists other IT policies that relate to or depend on the configuration, change, and release management policies.

Table 2.2 Related policies and procedures.

Policy	Relationship
Hardware Maintenance	Changes made in support of hardware maintenance objectives fall under processes and procedures for making vendor- or IT-initiated changes and/or improvements.
Problems and Incident Management	Changes made in response to problems and/or incidents are governed by configuration, change, and release management policies and procedures.
Help Desk Services	Help Desk Services plays key roles in identification and implementation processes and procedures contained in this policy and procedures manual.
IT Security	Security policy and procedures will govern how certain changes and improvements are assessed and implemented, as well as test and release management procedures for subsystems and components that are under the cognizance of IT Security.
Disaster Recovery	System configuration data required for inclusion in the Disaster Recovery Plan has a direct relationship to the status accounting and auditing processes contained in this manual. Other related areas are: implementation back-out plans and the test and release process.

Section 3 Processes

This section walks through the processes associated with configuration, change, and release management. Each individual process fits within the larger combined process. Step-by-step explanations and tips are provided for each process.

3.1 Overview of Configuration, Change, and Release Management Processes

Five processes constitute the overall configuration, change, and release management meta-process. A change request goes through all five processes as it evolves from initial request to completion.

Each process has defined milestones and specific requirements that must be fulfilled. The approach ensures that each change results in a new, documented baseline configuration.

The overall process is described sequentially under the following topics.

- Before You Begin - Prerequisites and setup actions that must happen before the process can happen
- Process - The individual processes that constitute the change control cycle
- After the Process - Criteria for when the cycle is completed

This section will introduce each process by its function and will provide a diagram and description that shows the major elements of the process. Figure 3.1 and Table 3.1 illustrate a typical process.

Figure 3.1 Process diagram elements.

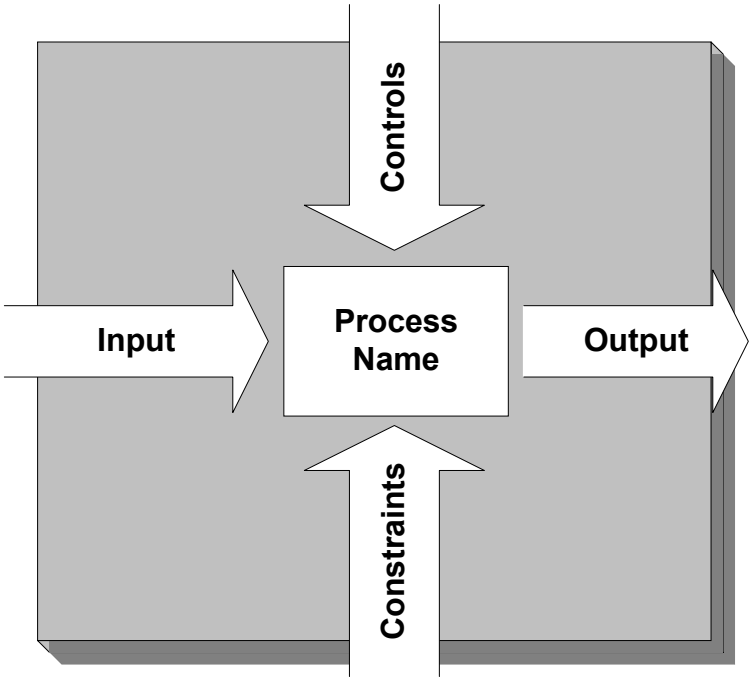


Table 3.1 Process diagram element descriptions

Components	Description
Inputs	What activities or information triggers the process or provides required input
Outputs	What results are produced by the process
Controls	What controls are applied to the process; i.e., service level objectives, safety considerations, etc.
Constraints	What constraints impact the process, such as budget, time, etc.
Activities	What happens inside of the process

Each process description will also have an associated diagram or flow chart that shows the basic operation of the activities that are contained within the process (identified in the above process map as Process Name).

A more detailed description of the activities, to the task level, are the procedures that are employed to produce the process output. The corresponding procedures are provided for each process in Section 4 of this manual.

3.2 Before You Begin

Change control begins when the Help Desk has received a request for a system change from a user, a vendor or from the IT department. The following activities are involved in preparing the change request.

- Filling in the change request form, or optionally calling the Help Desk
- Determining the priority of your request. The priorities levels include urgent, high, moderate, and low.
- Submitting the form to the Help Desk or having them submit it.

Note: For definitions of these levels, refer to Section 4.2.1, "Identifying a Request."

3.3 Configuration, Change, and Release Management Processes

The five-process cycle begins with identifying the request and ends with recording the change in the system configuration documentation. The five processes are defined in sequence below. The sixth process, auditing, is independent of the first five and will be defined in subsection 3.4.

3.3.1 Identifying a Request

The identification process starts with a requirement and ends with a requirement that has been defined. Figure 3.2 and Table 3.2 describe this process with its related controls and constraints.

Figure 3.2 Identification process, controls and constraints.

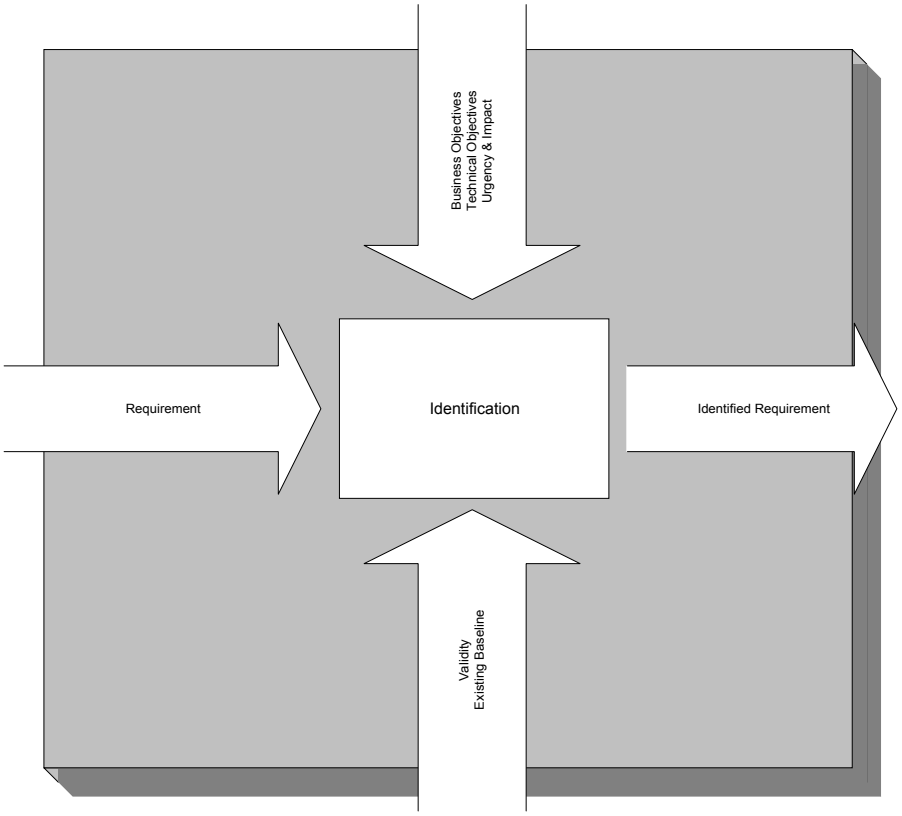


Table 3.2 Identification process, controls and constraints.

Components	Description
Inputs	Requirements that trigger the request for change.
Outputs	Identified requirement.
Controls	Business or technical objectives, urgency and the impact of operating without the change.
Constraints	Validity of the requirement and the existing system baseline.
Activities	A change request is initiated, logged and assigned a control number, the priority is validated, the initiator is provided with a tracking number, and the request is passed to the Configuration Control Manager for action.

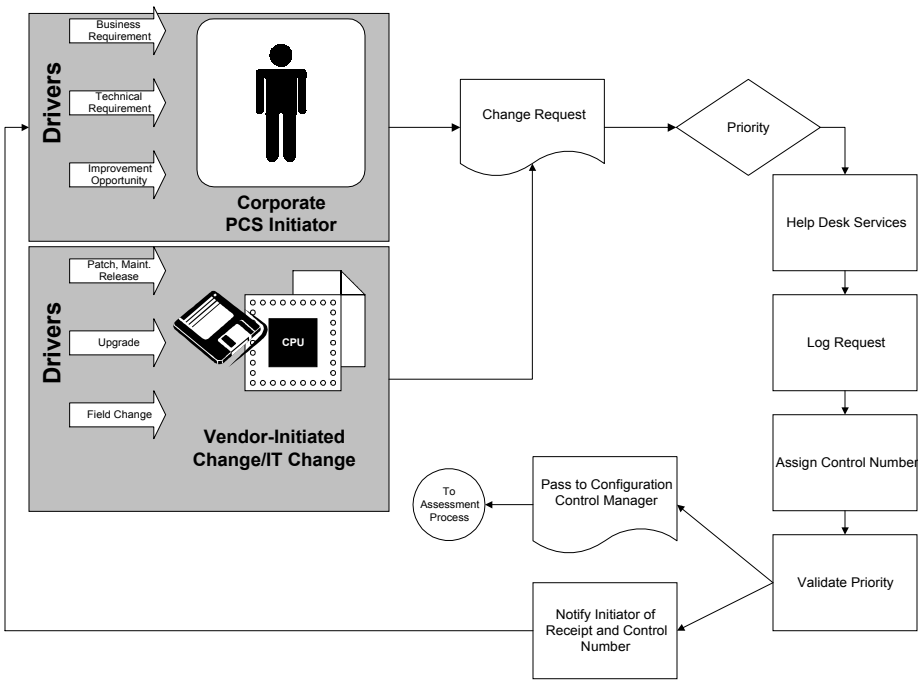
Help Desk Services is designated as the central point for receiving and tracking change requests. Upon receipt of a change request, Help Desk Services will initiate the following screening activities.

- Logging the change request and assigning a control number.

- Validating the request priority (as requested by the user).
- Notifying the initiator and providing a control number for reference.
- Passing the request to the Configuration Control Manager for resolution.

Figure 3.3 illustrates the identification process flow.

Figure 3.3 Identification process flow.



The vendor-initiated change and the IT changes follow different assessment paths. These changes proceed on a different track, passing directly to the configuration control manager or to a peer review.

3.3.2 Assessing and Approving a Change Request

As implied by its name, the combined assessment and approval process has two parts. The assessment process involves analyzing and recommending a specific system change, and the approval process involves evaluating and deciding whether or not to proceed with the change.

3.3.2.1 Assessing the Change Request

The assessment process starts with a defined requirement and ends with findings and recommendations for change. Figure 3.4 and Table 3.3 describes this process with its related controls and constraints.

Figure 3.4 Assessment process, controls and constraints

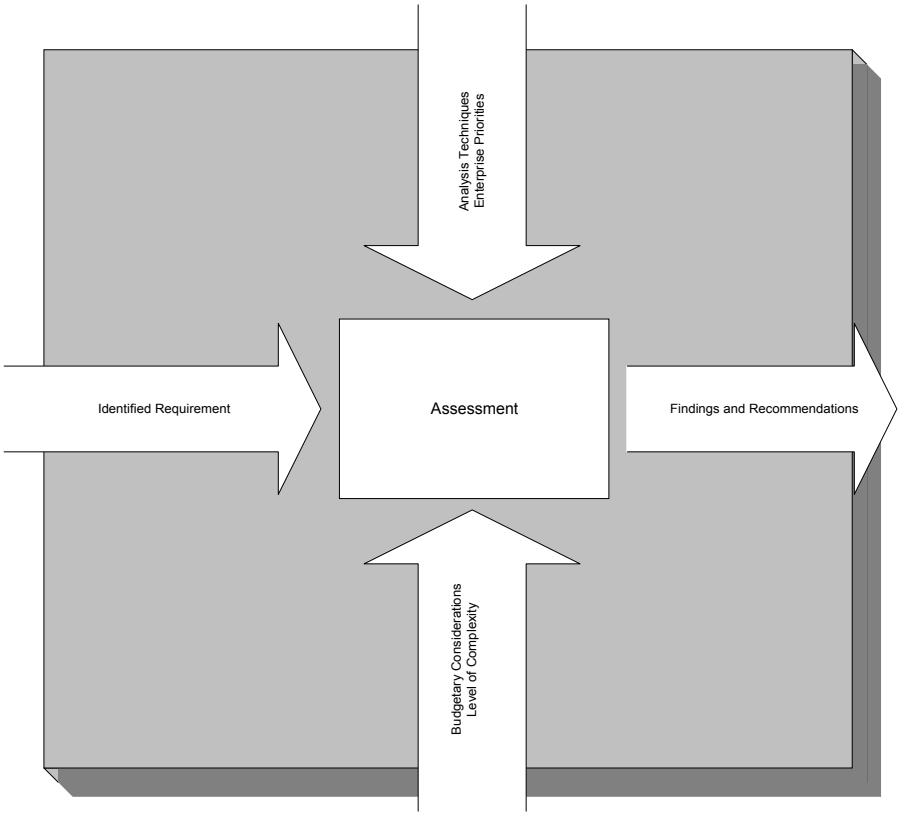


Table 3.3 Assessment process, controls and constraints.

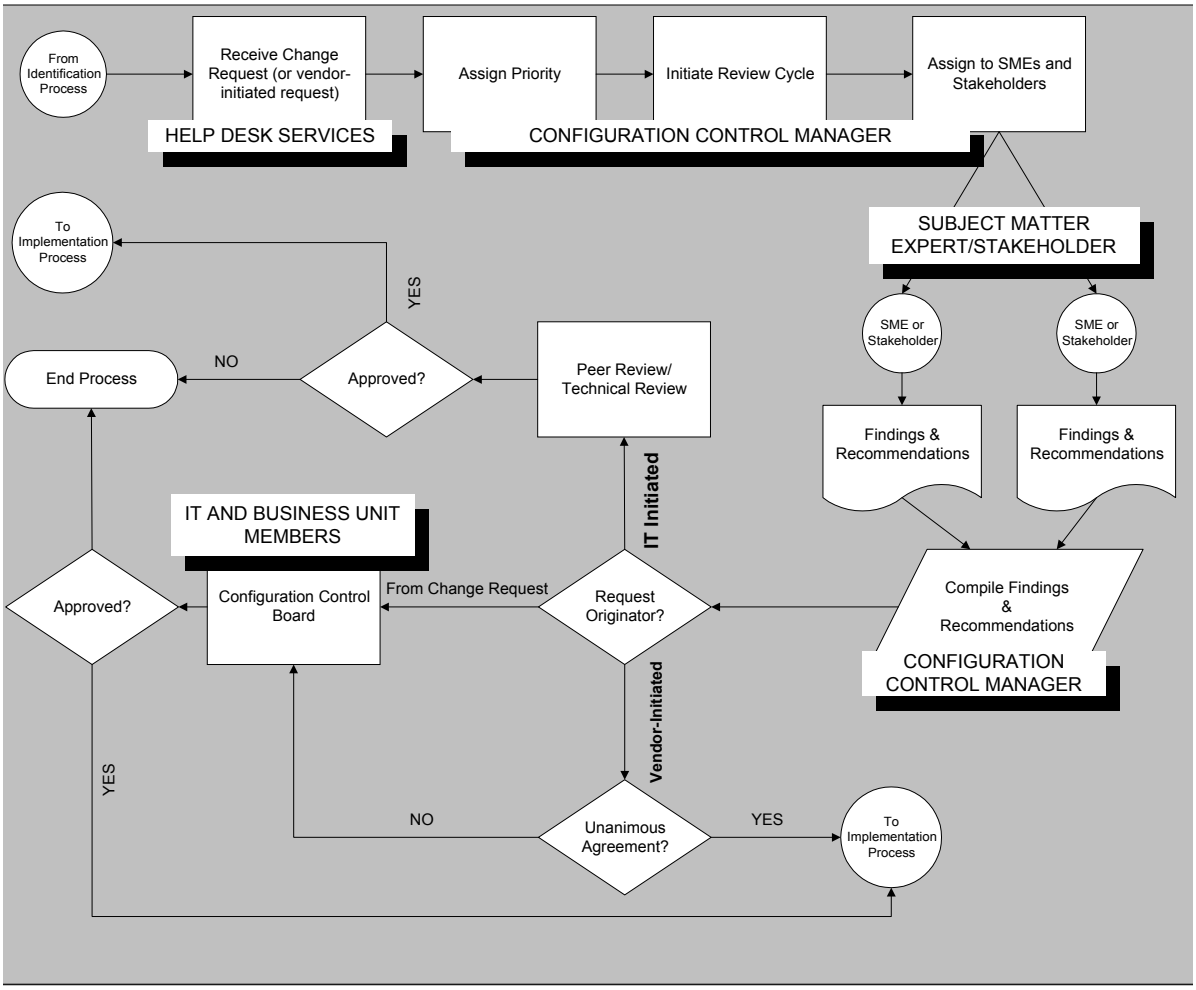
Components	Description
Inputs	Identified requirement.
Outputs	Findings and recommendations.
Controls	Analysis techniques, company priorities.
Constraints	Budget considerations, level of complexity to implement change.
Activities	Assign a master priority, initiate a review, compile findings and recommendations, forward findings and recommendations for approval/action.

The assessment process is initiated by the Configuration Control Manager and involves the following activities.

- Assigning a priority based on enterprise-wide requirements.
- Initiating a review cycle comprised of appropriate subject matter experts and stakeholders.
- Compiling findings and recommendations from the review cycle.
- Forwarding findings and recommendations to the proper assessment entity: The assessment entities include:
    - For change requests, the Configuration Control Board.
    - For vendor-initiated changes, the appropriate subject matter experts and stakeholders who must agree unanimously.
    - For IT changes, peer review and/or independent verification and validation.

Figure 3.5 illustrates the assessment process for the three types of requests.

Figure 3.5 Assessment process flow.



3.3.2.1.1 Criteria for Change Request Assessment

The principal objectives of the assessment process are to determine the feasibility and cost-effectiveness of the requested change by examining the following factors:

- Business requirements
- Technical requirements

- Impact of the requested change on:
  - Company strategic architecture
  - Life cycle management considerations
  - Interrelated and/or interdependent systems
  - Security
  - Data integrity
  - Meeting OLAs and SLAs
- Technical, cost or schedule risks
- Cost/benefit and predicted return on investment (ROI) that will be attained if the change is initiated

3.3.2.1.2 Processing Assessments

For each change request class, the assessment resolution is treated differently.

Resolving Change Requests. If the initial assessment meets all conditions for recommendation or further consideration, the change request is forwarded to the Configuration Control Board for approval.

Resolving Vendor-Initiated Change Requests. Vendor-initiated change will be implemented only if there is unanimous agreement among the review team of appropriate subject matter experts and stakeholders.

If review team members cannot reach a consensus either for or against the change, the matter will be referred to the Configuration Control Board for a final determination.

Resolving IT Changes. IT changes will be implemented only if they have been validated by peer review and/or independent verification and validation procedures.

Change Request Cancellation. Changes that neither qualify for forwarding to the Configuration Control Board nor pass peer review and/or independent verification and validation will be canceled in accordance with change request close-out procedures.

3.3.2.2 Approving a Change Request

The approval process starts with a recommendations for change and ends with a decision to implement or deny the change request. Figure 3.6 and Table 3.4 depict this process with its related controls and constraints.

Figure 3.6 Approval process, controls and constraints.

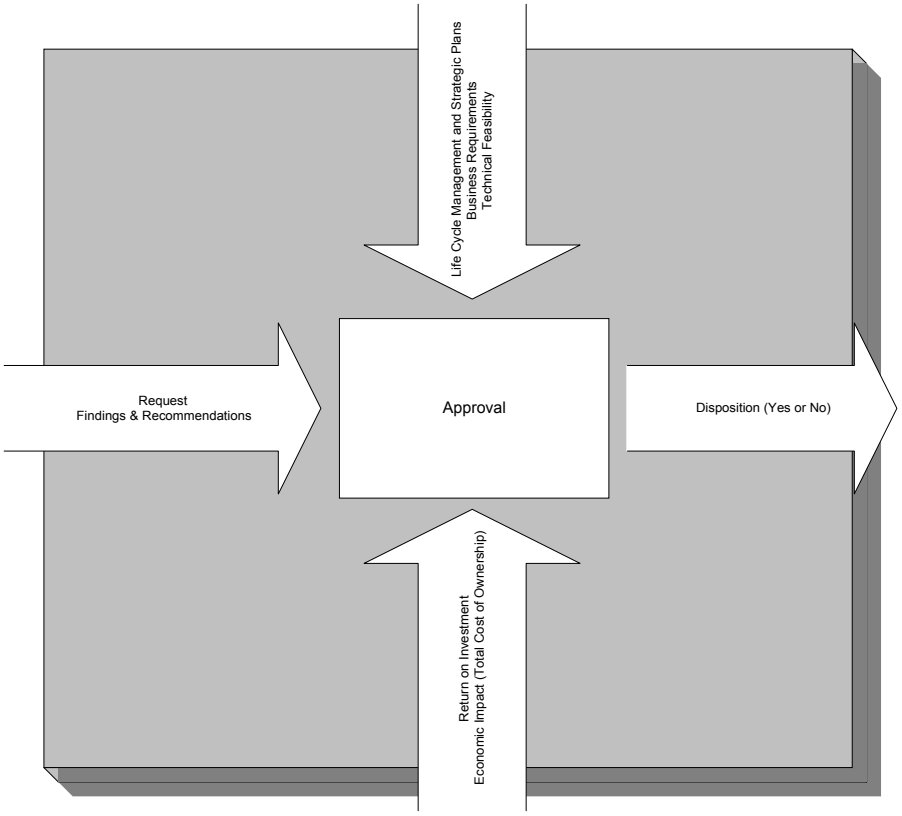


Table 3.4 Approval process, controls and constraints.

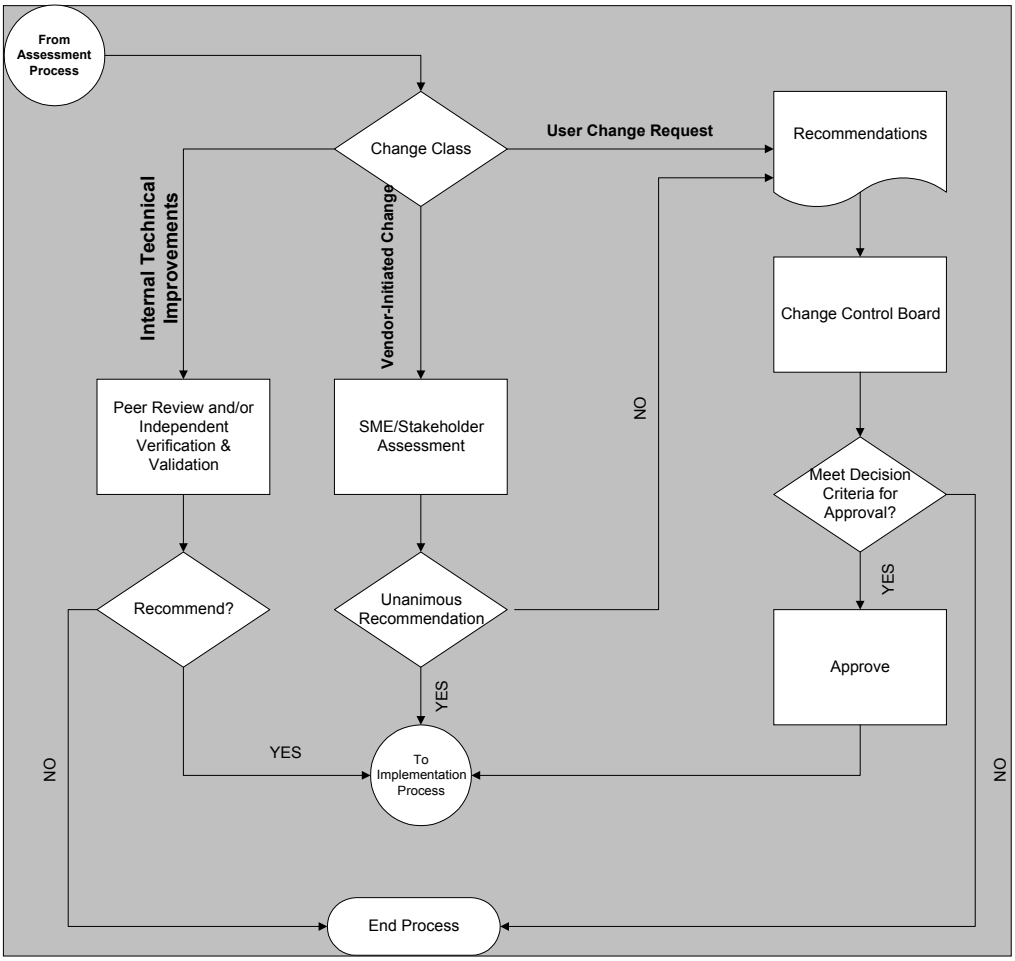
Components	Description
Inputs	Findings and recommendations
Outputs	Disposition (approved or disapproved)
Controls	Life cycle management plan; strategic plan, business and technical objectives
Constraints	Return on Investment (ROI); economic impact
Activities	Decision making techniques

The Configuration Control Board uses specific criteria to approve change requests. The scope of this policy and procedures manual does not allow the definition of these criteria.

In general, the decision criteria for approval will reflect criteria such as business requirements, the experience and judgment of the Configuration Control Board members and budget considerations.

Each class of request follows a different approval path as illustrated in Figure 3.7.

Figure 3.7 The approval process.



3.3.3 Implementing a Change

The implementation process starts with the authority to proceed and ends with the completed change. Figure 3.8 and Table 3.5 depict this process with its related controls and constraints.

Figure 3.8 Implementation process, controls and constraints.

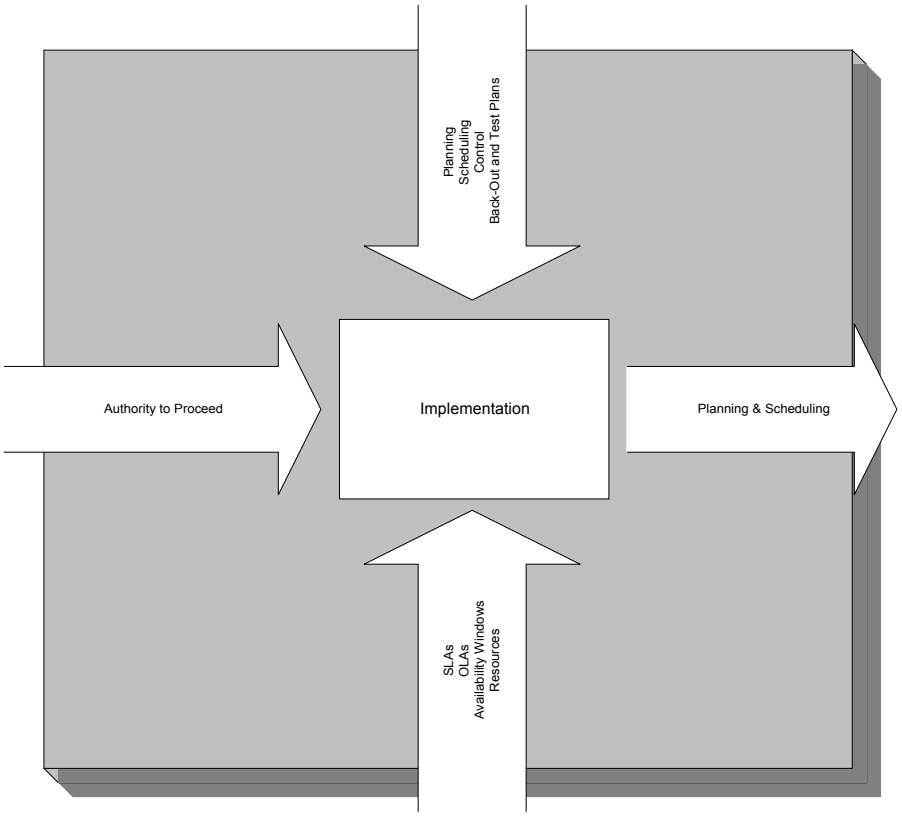


Table 3.5 Implementation process, controls and constraints.

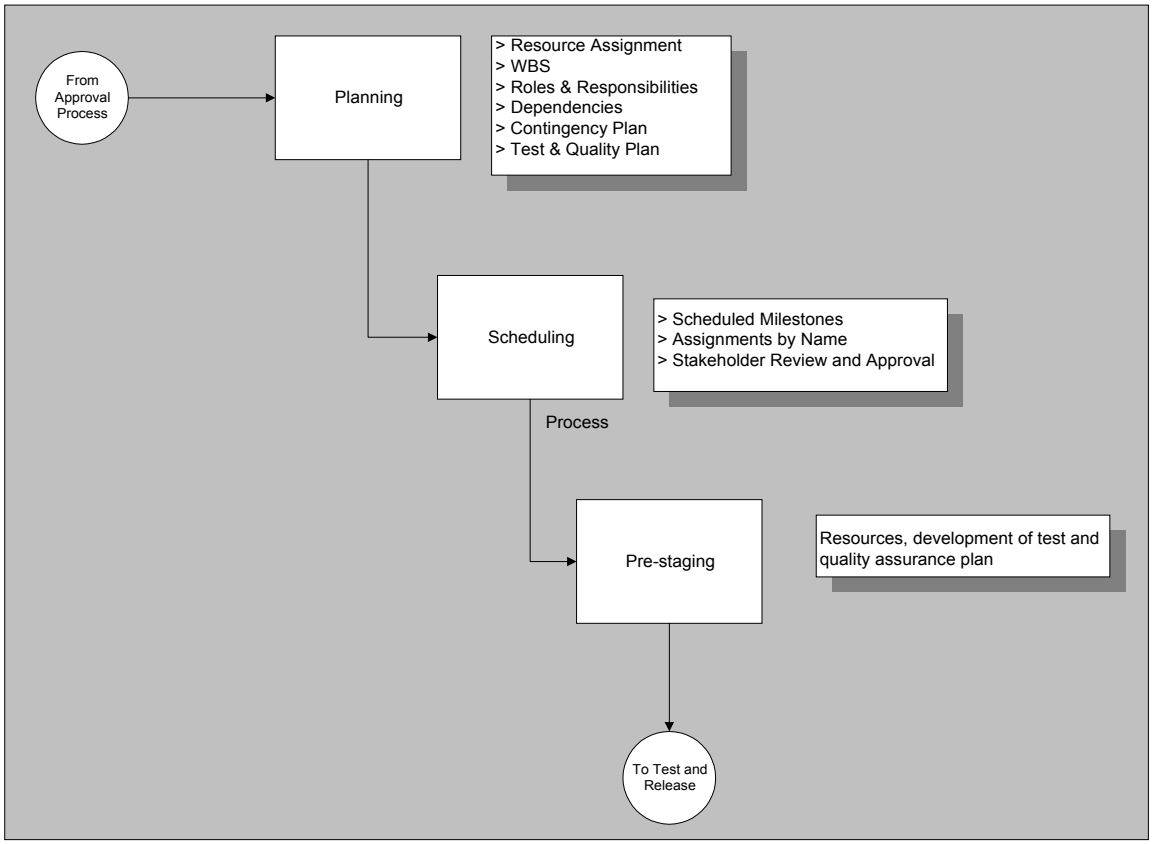
Components	Description
Inputs	Authority to proceed
Outputs	Planning, scheduling and pre-staging
Controls	Planning, scheduling and control; back-out and test plans
Constraints	SLAs, OLAs, availability windows, resources
Activities	Implementation planning and pre-staging (plans, schedules, test and back-out planning, identified resources)

The implementation process requires three critical elements.

1. Planning
2. Scheduling
3. Pre-staging

Figure 3.9 illustrates the flow of these four elements in the implementation process.

Figure 3.9 The implementation process.



3.3.3.1 Planning

All implementation requirements and resources will be planned in advance of the actual change to be effected. Planning encompasses the identification and commitment of:

- Personnel resources

- Parts and materials
- Availability of alternate sources of services when applicable or possible
- Plan of action (for example, work breakdown structure, roles and responsibilities charts, contingency and back-out plans, etc.)
- Quality assurance test and release plans and criteria

The implementation plan will also address:

- Statement of work and defined scope
- Upstream and downstream dependencies
- Technical, cost and schedule risks

3.3.3.2 Scheduling

At a minimum the implementation schedule will contain:

- Scheduled implementation window, including worst-case stop times and identifiable rollback point
- Milestones for task completion, quality assurance checkpoints and release into production
- Personnel assigned to tasks by name

The implementation schedule will be reviewed and approved by all stakeholders prior to proceeding with the implementation activities. Stakeholders include all technical and business unit personnel who are involved with, or affected by, the implementation.

3.3.3.3 Pre-Staging

This activity provides support of the actual implementation. Steps include confirming resource availability, staging materials, preparing the system for the change and executing all plans that are scheduled to occur before test and release.

3.3.4 Testing and Releasing a Change

The test and release process starts with a completed implementation and ends with the system restored for operational availability with documented system changes. Figure 3.10 and Table 3.6 depict this process with its related controls and constraints.

Figure 3.10 Test and release process, controls and constraints.

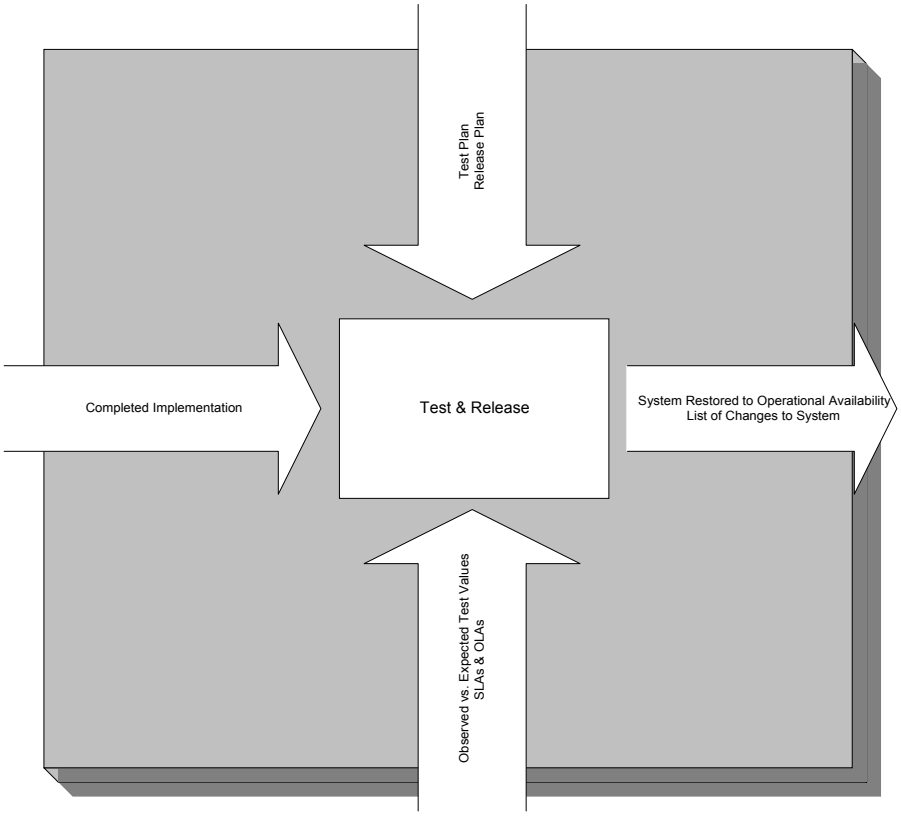


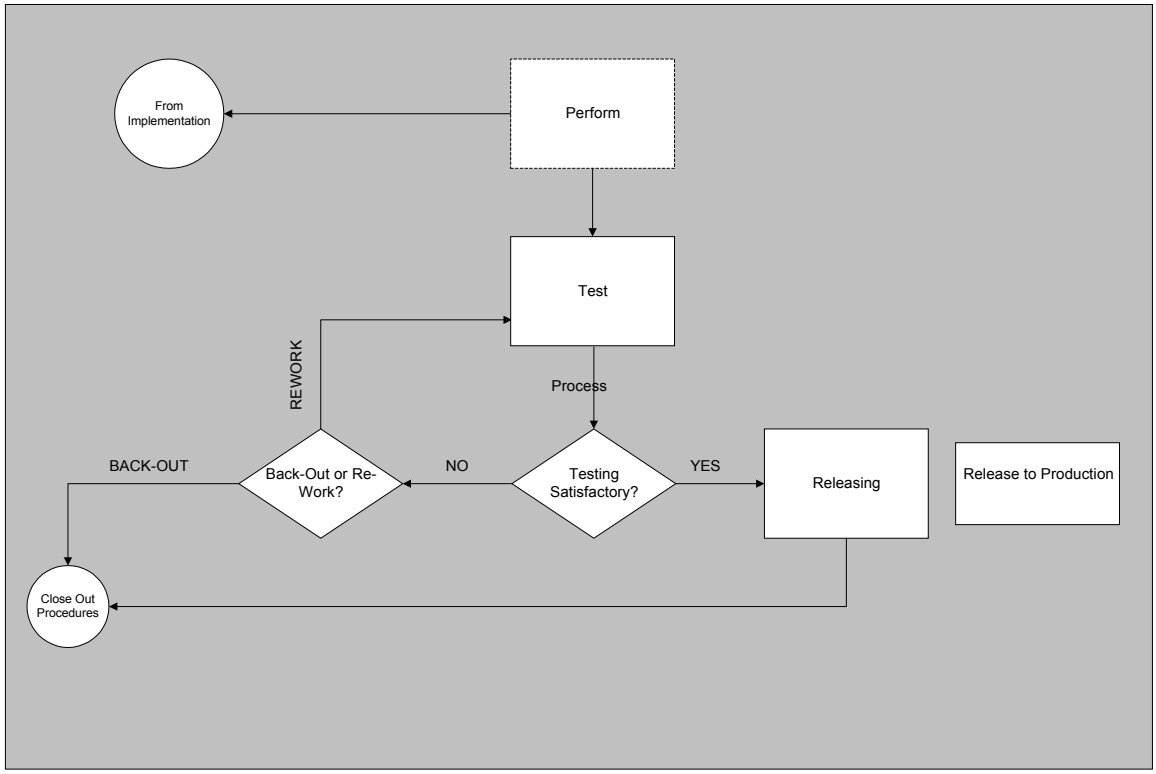
Table 3.6 Test and release process, controls and constraints.

Components	Description
Inputs	Completed pre-staging, planning and scheduling.
Outputs	System returned to operational availability.
Controls	Test plan.
Constraints	Observed vs. expected test values and pass/fail checkpoints.
Activities	Test in accordance with test plan to ensure correct system operation per specifications, release system to operational status after all test and

	release conditions are met..

Figure 3.11 illustrates the test and release process.

Figure 3.11 Test and release process flow.



3.3.4.1 Testing

The quality assurance and test plan is the most critical element of the implementation plan. This plan is a checklist of test inspections, observations and validations to ensure that the affected system operates within expected parameters.

The quality assurance and test plan also includes production checklists and addresses any factor or issue related to ensuring that the system is fully operational and supportable. This includes ensuring that all required materials (media, documentation, etc.) in connection with the change are updated or provided.

A test plan needs to be thorough and comprehensive, and designed to exercise the system as a whole as well as upstream and downstream dependencies.

3.3.4.2 Release

The basis for releasing a change to production will be a satisfactorily executed test plan.

Test plans will be executed and signed off by a member of the IT staff who was not involved in the implementation process or by the system owner. Under no circumstances will the test plan be signed-off by the individual who performed the implementation task(s).

3.3.5 Status Accounting

The status accounting process starts with a listing of changes to the system and ends with an accurate configuration baseline. Figure 3.12 and Table 3.7 depict this process with its related controls and constraints.

Figure 3.12 Status accounting process, controls and constraints

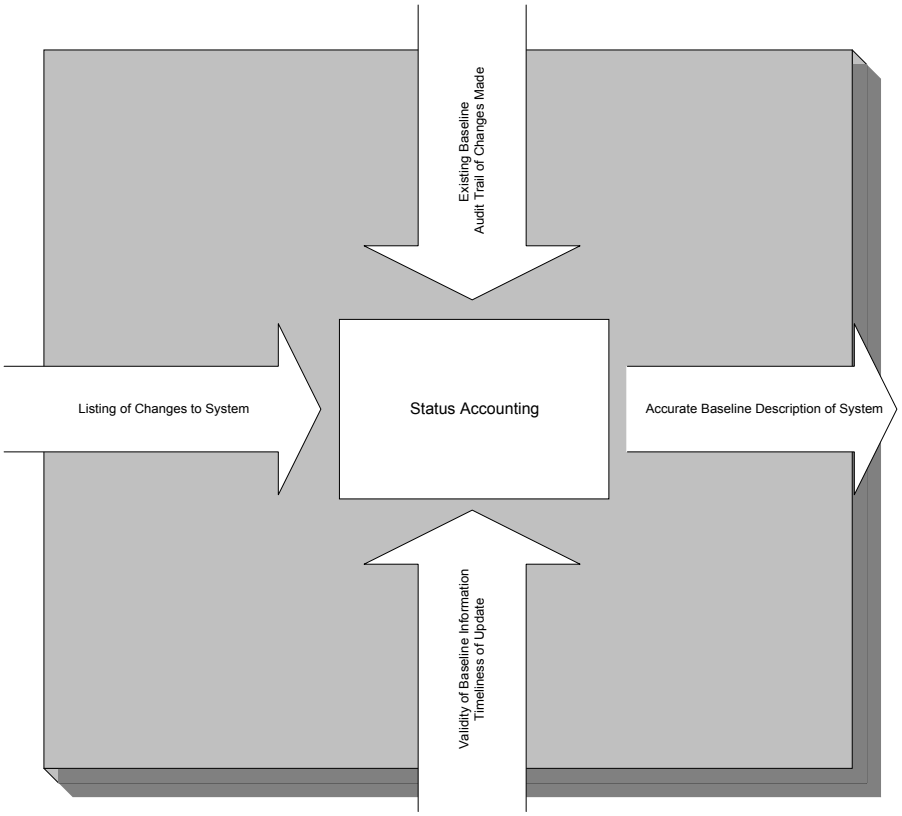
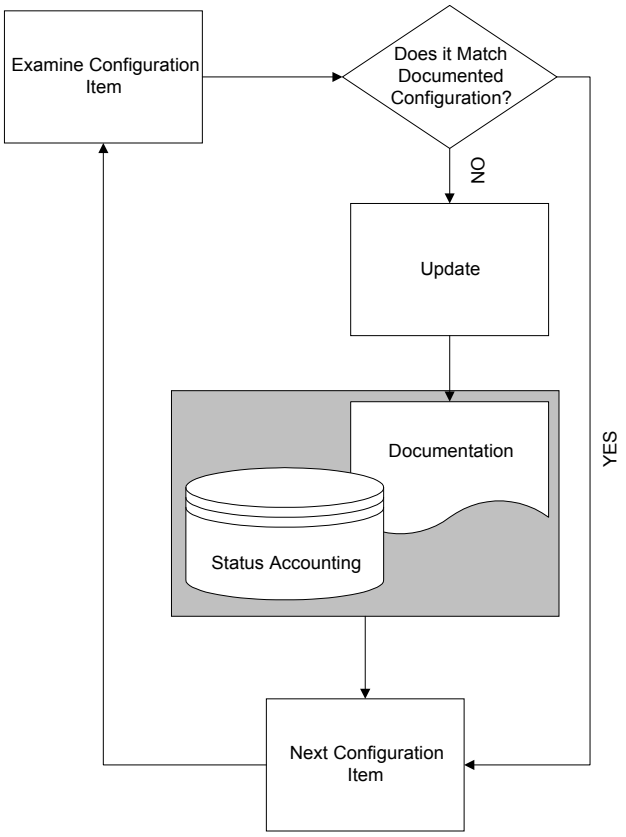


Table 3.7 Status accounting process, controls and constraints.

Components	Description
Inputs	List of changes made to system during implementation.
Outputs	Accurate baseline description of system.
Controls	Existing baseline.
Constraints	validity of baseline; timeliness and accuracy of update.
Activities	Update all configuration data.

Status accounting ensures that all changes to the system are documented as they occur. Figure 3.13 illustrates the flow of the status accounting process.

Figure 3.13 The status accounting process.



The status accounting process depends on baseline configurations, beginning with the functional baseline and ending with the operational baseline.

Functional Baseline. The functional baseline or specification, derived from business requirements, are tracked, modified and consolidated until an allocated baseline is established. The functional baseline specifications are, at some point, frozen and the development of the allocated baseline begins.

Allocated Baseline. The allocated baseline, comprised of required standards and a set of technical specifications, are used to either issue a request for proposal or to build and develop the system.

Product Baseline. The product baseline, the final system configuration before the system is operational, results from the allocated baseline. At the product baseline stage, usually after the product has been delivered or built, the following configuration items must be documented as built.

1. Hardware platform configuration items
    - Manufacturer and model
    - Manufacturer serial numbers and company asset numbers
    - RAM (configured quantity in MB, density in modules and capacity, speed in Ns, and available slots for additional RAM modules)
    - Firmware revision levels and date installed
    - Mass storage devices attached to system (magnetic and optical); including: manufacturer/model information, applicable serial numbers and/or asset numbers (for external devices), physical characteristics (capacity, SCSI ID assigned, operating parameters, and so on.)
    - I/O devices attached to system (serial, parallel, SCSI, network) and all identifying configuration information (adapter ID, MAC address, and so on.)
    - Display subsystems attached to system and all identifying configuration information (settings, performance characteristics, manufacturer serial numbers and asset numbers)
2. Operating system software
    - Version and revision level
    - Applied patches
    - Current build number
    - Configuration of installed network extensions (version/revision, IP address(es) assigned, and so on.)
    - Version/revision level of system utilities and drivers (device drivers, compilers, performance monitors, transaction monitors, and so on.)
3. Application Suite
    - Version and revision level

- Applied patches

4. System Documentation

Operational Baseline. After the system has been placed in operation, status accounting consists of maintenance functions. They include:

- Maintaining the configuration database and associated documentation in a manner that reflects the true state of the system.
- Maintaining the procedures for documenting system changes.
- Ensuring that all the above mentioned related system components are synchronized (i.e., hardware, software and documentation).

3.4 Completing the Cycle

Once the status accounting has been completed, the request is closed out. These completion activities are described below.

3.4.1 Closing out a Change Request

Configuration, change, and release management will be completed when the criteria listed in Table 3.8 are met for each of the process owners.

Table 3.8 Criteria for change request completion

Process Owners	Action	Criteria for Completion
Help Desk Services	Close out change request, notify all stakeholders	One of the following conditions: • Notification from the implementation project manager that (1) the implementation is complete (2) the system has operated for 48 continuous hours without errors and per test and release procedures (3) all status accounting procedures have been completed • Change request has been denied • Change implementation was aborted and no further attempts to implement the change will be made
Configuration Control Manager	Update configuration documentation	Notification from the implementation project manager that (1) the implementation is complete (2) the system has operated for 48 continuous hours without errors per test and release procedures

3.4.2 Auditing the Configuration Database

Auditing is the verification and validation process in configuration, change and release management. It entails verification of the actual system configuration against the documented system configuration. This is an ongoing task after the system has been placed into production.

The auditing process is a separate activity that is independent of the change cycle. The auditing process starts with an existing baseline and ends with a verified baseline that reflects the actual configuration. Figure 3.14 and Table 3.9 depict this process with its related controls and constraints.

Figure 3.14 Auditing process, controls and constraints

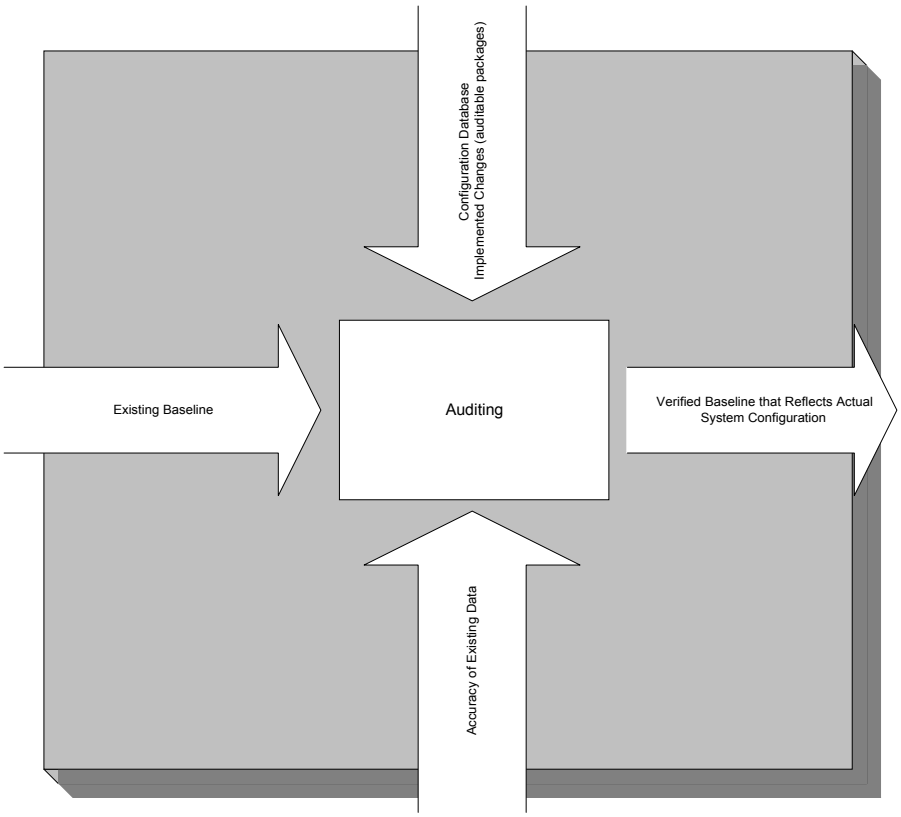


Table 3.9 Auditing process, controls and constraints.

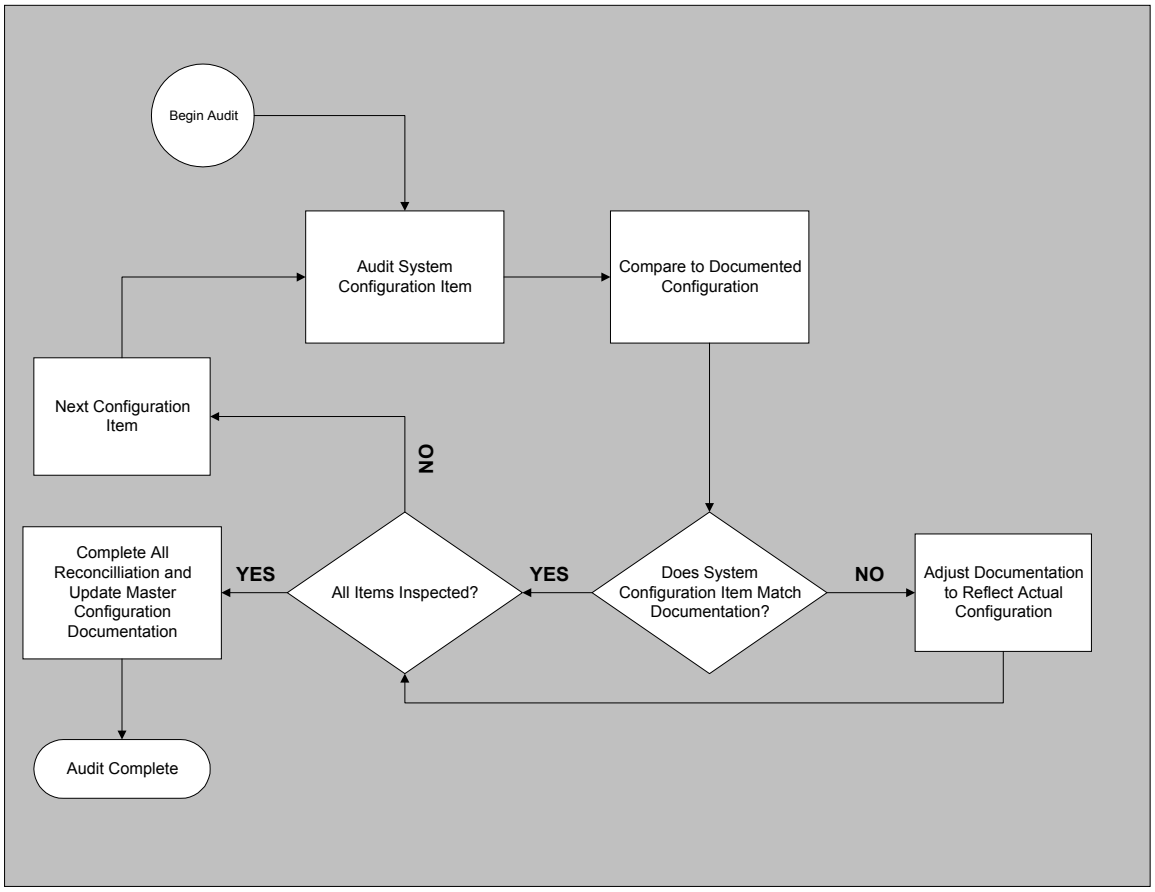
Components	Description
Inputs	Existing baseline
Outputs	Verified baseline that reflects the system's actual configuration
Controls	Configuration documentation, audit copies of implemented change requests
Constraints	Accuracy of existing data and documentation
Activities	Verify system configuration against configuration documentation, adjust baseline documentation to reflect any discrepancies

The sole purpose of the audit process is to ensure that configuration documentation accurately describes the system's actual configuration. As such the process is straightforward and conducted in a manner similar to a financial audit: the system is inspected (audited) and compared to the documentation. When discrepancies are found the configuration documentation is adjusted to

reflect the audited configuration. At the end of the process the system and its associated documentation will be reconciled in a fashion similar to reconciling a ledger against receipts and canceled checks.

Figure 3.15 illustrates the auditing process.

Figure 3.15 Auditing process flow.



Note: For a list of functional areas to be audited, the audit scope, and recommended audit intervals, see the auditing procedures in the Section 4, Procedures.

Section 4 Procedures

This section provides procedures for configuration, change, and release management activities. The procedures explain in detail what you need to do to use the process. Following these procedures will help you coordinate with others involved in the process and ensure that system changes are well-managed and documented..

4.1 Overview of Configuration, Change, and Release Management Procedures

The change request form is the vehicle for communicating about the proposed change. You initiate a change by filling out the form, which details the desired outcome of the change to those who will help define, approve and implement it.

The procedures follow the processes described in Sections 3 with closer attention to the specific actions you must take at each step along the way. The procedures will help you synchronize with others in the organization who are also contributing and responsible for moving the request through the change process.

To fully cover the broad requirements of a change process, the procedures have been organized under the following topics:

- Using Configuration, Change, and Release Management Procedures - Procedures for carrying out the change control tasks in change management systems
- Auditing Configuration, Change, and Release Management - Procedures for ensuring the accuracy of the baseline configuration documentation and handling inconsistencies
- Improving Configuration, Change, and Release Management - Procedures for diagnosing and resolving issues in order to improve the overall process

Table 4.1 outlines the specific procedures available to you in this section.

Table 4.1 Configuration, change, and release management tasks

Topic	Tasks
Using Configuration, Change, and Release Management Procedures	Identifying a Request Assessing a Change Request Approving a Change Request Implementing a Change Testing and Releasing a Change Status Accounting
Auditing Configuration, Change, and Release Management	Auditing a Change Process Handling Inconsistencies Found in an Audit
Improving the Configuration, Change, and Release Management Process	Improving the Process

4.2 Using Configuration, Change, and Release Management Procedures

Change requests follow a particular sequence from initiation to completion. The steps may vary depending on whether you are a system user, vendor, or IT employee. Although the procedures cover the complete process, you may only be involved in certain aspects of the end-to-end process.

4.2.1 Identifying a Request

System changes are grouped into three classifications:

1. Change requests - usually initiated by a system user
2. Vendor-initiated changes - patches, maintenance releases
3. IT changes - internal technical improvements and fixes, usually initiated by IT staff

Each of these three change request initiators, fill out the same change request form. The differences occur in the way the change request is processed after it has been submitted by the initiator.

To fill out a change request form.

1. Fill in the change request form, or optionally call the Help Desk and fill it in with a Help Desk associate.
2. Submit the form to the Help Desk or if you called the Help Desk, they will submit the change request form for you.
3. Choose the priority of your request from the options list on the form..
4. Obtain the name of the Help Desk associate and the control number of your change request for future reference.

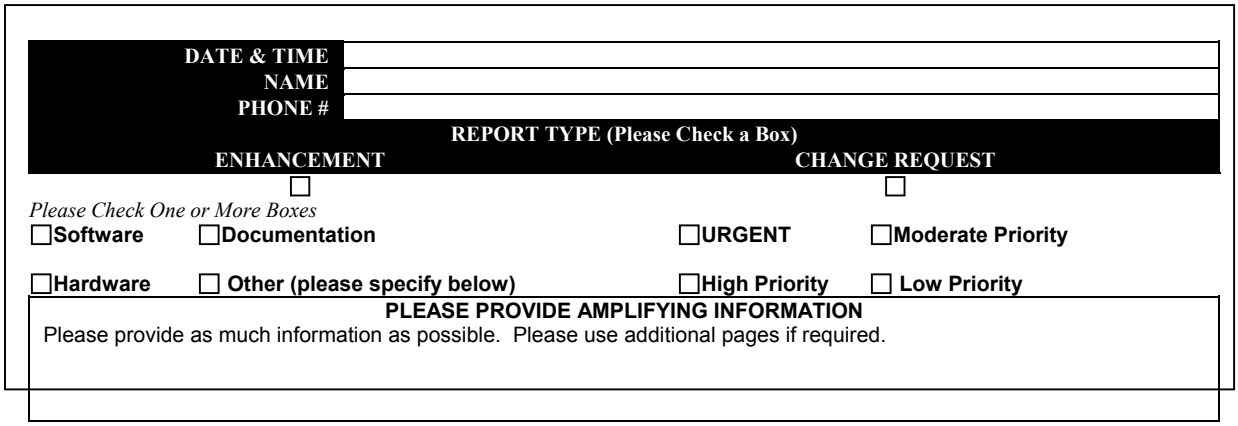
4.2.1.1 Change Requests

Change requests will be made on the change request form depicted in Section 5. A detailed description of this form follows.

The identification section of the change request form has fields for date and time, initiator identification and the nature of the request. This section is designed to simplify the report initiation process, especially for non-technical end users.

Figure 4.1 illustrates the identification portion of the change request form.

Figures 4.1 Identification portion of the change request form.



The nature of the request will fall under one of the following two categories:

- Enhancement: Suggestions for ways to improve the system for usability, reliability, user friendliness, etc.
- Change Request: Formal requests for changes to improve the system

All initiators determine the request priority using the following options:

Table 4.2 Change request priorities.

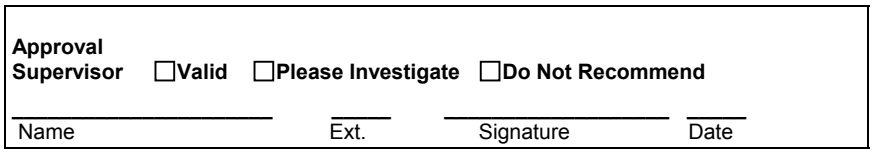
Priorities	Definitions
Urgent	The request must be acted upon immediately. This priority is intended to address required changes to align processing to business processes. An example is an encoded business rule that has been rendered invalid by a new corporate policy or reengineered business function
High	The request must be acted upon as soon as possible. An example of a High priority request is a requirement for more user file space on a system that is at 80% mass storage capacity. Any delay in implementing a solution could result in severe system problems and shutdown of business functions.
Moderate	Important near term requirement, such as an anticipated growth in a business unit user population, need for a higher speed peripheral such as printer or modem to meet business processing demand in a timely manner, etc.
Low	“Nice-to-have” feature or enhancement, such as color printing capability, departmental facsimile server for convenience, etc.

4.2.1.2 Review and Disposition

Change requests will be directly forwarded to Help Desk Services for logging and further action. All enhancements and change requests must be approved by the initiator's supervisor before submission to Help Desk Services. The review and disposition fields in the form, provided in Section 5.1, Exhibits, Change Request Form, are shown in Figure 4.2.

Figure 4.2 illustrates the supervisor approval portion of the change request form.

Figures 4.2 Supervisor approval portion of the change request form.



This portion of the report form is for the initiator's supervisor to review and to make recommendations for either approval or disapproval. This will ensure that all recommendations for changes or improvements are first validated at the business unit level.

Upon receiving a change request, Help Desk Services has the responsibility of logging and validating the completeness of the request. The Help Desk procedure for this task is as follows:

To log, review, and hand off the change request:

1. Check that the request has been completed properly.
2. Log change request and assign a control number.
3. Validate the request priority (as requested by the user).
4. Notify initiator and provide control number for reference.
5. Pass request to the Configuration Control Manager for resolution.

The outcomes of the FINDINGS AND DISPOSITION will either be RESOLVED or REFERRED FOR ACTION. This provides Help Desk Services with an opportunity to provide a solution in response to the change request if the request can be so handled. In all cases Help Desk Services will track the report

until the report is closed out. Help Desk Services will notify the report's initiator of all actions taken after the report is closed out.

Figure 4.3 illustrates the approval portion of the change request form

Figure 4.3 Disposition portion of the change request form.

Control Numbers will be assigned by the Help Desk Services

CHANGE REQUEST ACTION **CONTROL NO. __-_____**

Date/Time Received (Help Desk)				
Date/Time Logged (Help Desk)				
Date/Time Received (Configuration Control Manager				
Integrated Priority Assigned				
Assigned to Review Team		**Review Team Member(s)**		
Name				
Date/Time Assigned				
Date/Time Response Required				
Date/Time Response Received				
Recommendation	☐ **YES** ☐ **NO**	☐ **YES** ☐ **NO**	☐ **YES** ☐ **NO**	☐ **YES** ☐ **NO**
Date/Time Originator Notified Of Receipt Of Report				

FINDINGS AND DISPOSITION (Attach Report)

☐**Resolved** ☐**Referred to _____ for action**

Notice that the fields in this portion of the change request capture names and dates. This allows "cradle-to-grave" tracking, accountability, and enables the identification process to be audited.

If the change request cannot be immediately resolved by Help Desk Services it is forwarded to the Configuration Control Manager for assessment.

4.2.2 Assessing and Approving a Change Request

The combined assessment and approval procedures describe how to arrange for review of the request and how the change request approval happens.

4.2.2.1 Assessing a Change Request

Change assessment procedures will be determined by the change classification. There are separate procedures for change requests, vendor-initiated changes and IT changes. Each of these procedures are described below.

The assessment process is initiated by the Configuration Control Manager and involves the following procedure.

To assess a change request

1. Assign a priority based on enterprise-wide requirements
2. Initiate a review cycle comprised of appropriate subject matter experts and stakeholders
3. Compile findings and recommendations from the review cycle
4. Forward findings and recommendations to the proper assessment entity: The assessment includes:
    - For change requests, review by Configuration Control Board
    - For vendor-initiated changes, unanimous agreement among the appropriate subject matter experts and stakeholders
    - For IT changes, peer review and/or independent verification and validation

4.2.2.1.1 End-User Change Requests

In the assessment process, change requests initiated by system users have two primary activities: the initial actions and the review procedures. This provides for review by a Configuration Control Board to determine whether to accept or deny the change.

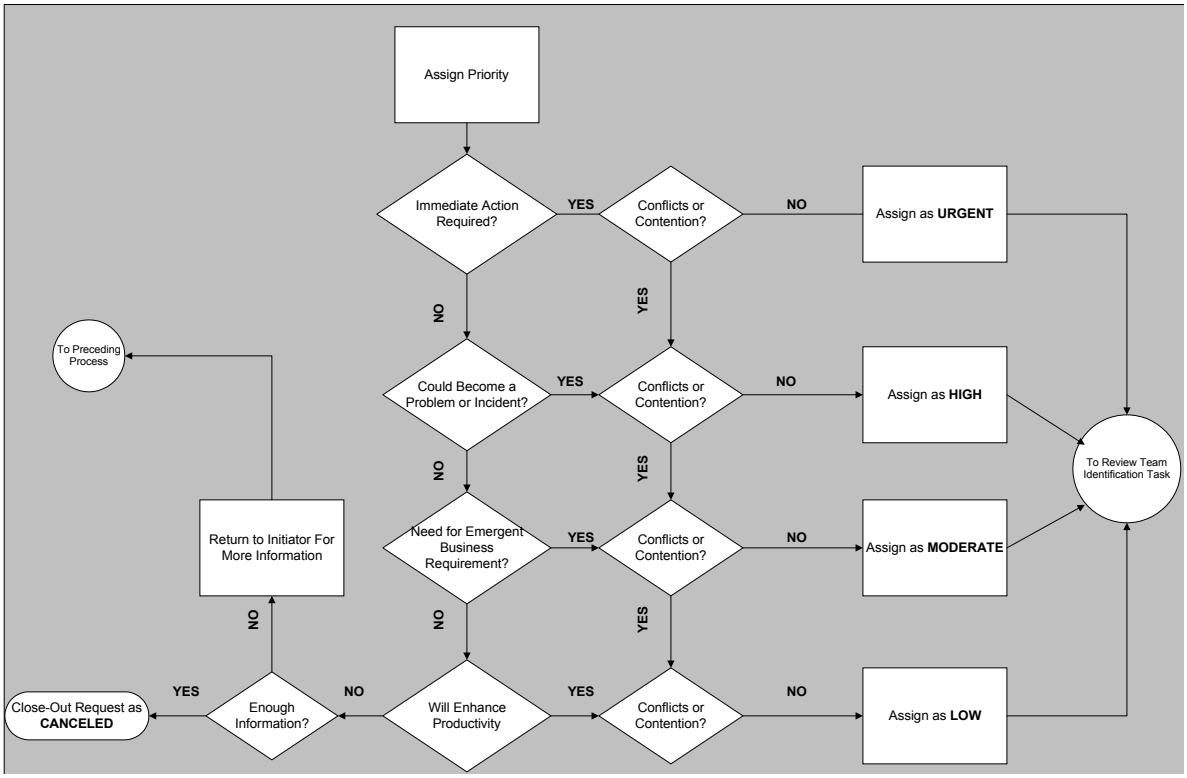
Initial Actions

Upon receiving a change request from Help Desk Services the Configuration Control Manager will take the following initial actions:

- Determine the request priority within the context of enterprise-wide requirements and pending change requests and projects.
- Identify candidates to comprise the review team (subject matter experts and stakeholders)
- Initiate the assessment

The decision tree at this point in the change request assessment procedure is shown in the flowchart below.

Figure 4.4 Assessment procedure



Priority Determination and Assignment (Example)

If a change meets all of the conditions for a particular priority, but it conflicts with other pending requests of the same priority profile, the Configuration Change Manager will assign it to the next lower priority.

An example of how and why a request would be downgraded during the initial phase of the assessment process is as follows.

Situation. A change request is received for additional disk space on a file server. The reason this request was marked as HIGH by

the initiator is that existing available disk space is approximately 15% and some large print jobs cause server ABENDS, disrupting productivity for a 30-person business unit at least 3 times a week. Work-around measures to date have been to remove all non-essential files on a daily basis to hold available disk space to 15% availability, and to reconfigure applications that use the server to not make back-up images of files (i.e., MS Word .BAK files). Because the situation indicates that the system is in imminent danger of failure, and because work-arounds have required disabling disaster recovery features, all conditions for a valid priority of HIGH exist.

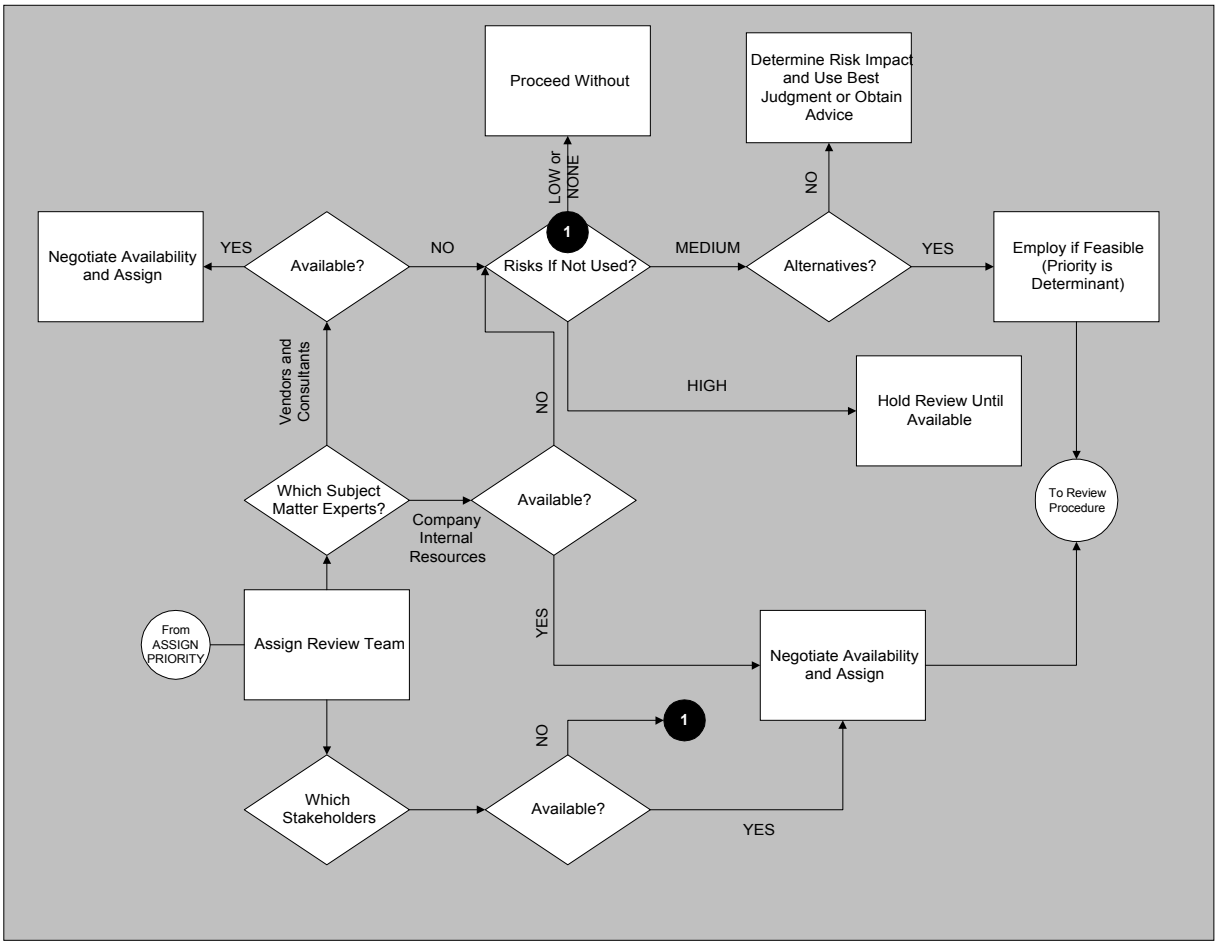
Conflicts and Contentions. The Configuration Control Manager examines open change requests and pending projects and finds that all available resources are committed to implementing an URGENT change request that impacts the entire company and has close executive and management attention. As a fall-back the Configuration Control Manager attempts to contract with outside systems integrators to effect the change, but none can respond immediately. Resources will be available to implement the change within two days.

Resolution. The change request is downgraded to HIGH.

Review Team Selection and Assignment

The next step in the change request assessment procedure is to select and assign review team members. This task is diagrammed in the following flowchart:

Figure 4.5 Review team selection procedure.



As can be seen, this task requires careful coordination of available resources. In the event that resource conflicts arise (not an uncommon occurrence) there are available avenues for effective conflict resolution based on the degree of risk from not using a particular resource as a review team member. Risk assessment and management methods are addressed below under the topic "Review Procedures, Examine Impacts and Dependencies."

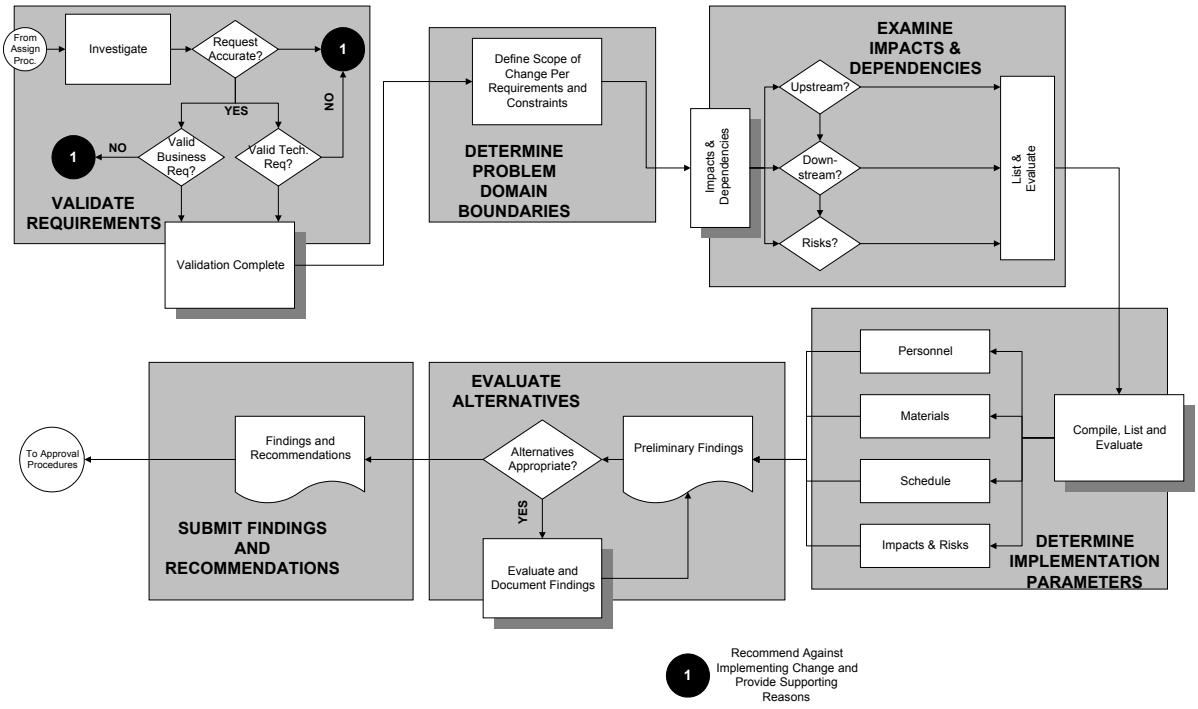
Review Procedures

The tasks that comprise the review procedure set is based on straightforward analysis techniques. Key elements are:

- Validate requirements and [any] assumptions
- Determine the problem domain boundaries
- Examine impacts and dependencies
- Determine implementation parameters
- Evaluate alternatives (if applicable or appropriate)
- Submit findings and recommendations

Task flow and interrelationships are:

Figure 4.6 Change request review procedure.



Validate Requirements

Requirements validation is a matter of ensuring that the change request is accurate (it reflects a real condition or need) and that it addresses valid business and/or technical requirements.

If the request is deemed to be inaccurate the course of action is to clarify, with the initiator, what conditions led to submitting the request. Because many of the requests will originate from the end user community perceptions, not facts, may be the primary factor in the change request initiation process. At this point if the request is indeed evaluated as inaccurate or without justification the reviewer's next action is to recommend against the change request and provide supporting reasons, ending the review process.

In cases where valid business ***and*** technical requirements are both found not to exist the reviewer will recommend against the change request and provide supporting reasons. As in the case above, this action ends the review process.

The change request review will continue to the next task, determining problem domain boundaries, if the validation procedures find the change request to be accurate ***and*** there is ***either*** a valid business ***or*** technical requirement.

Determine Problem Domain Boundaries

Problem domain boundaries define the scope of the change under review. For example, an application configuration change request will probably not go beyond the scope of configuration guidelines and procedures for the particular application. However, if the change request was for a departmental web server, the scope of the request expands to encompass business policies, security issues, network traffic, telecommunications interfaces and ongoing costs to operate the system after it has been implemented.

Determination of scope requires that the following factors and issues be taken into account:

- Business and technical requirements
- Constraints

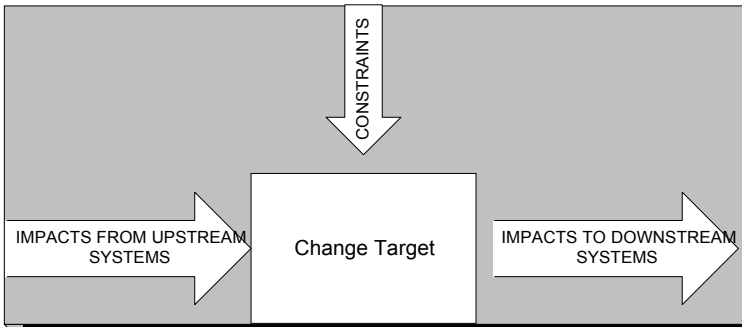
Common constraints when defining the scope are:

- Business and corporate policy
- Security issues
- Budget
- Support resources
- Life cycle management

Scope is tightly coupled with impacts and dependencies, which is the next task in the review process.

Examine Impacts and Dependencies

Figure 4.7 Impacts and dependencies.



Once the problem domain has been defined the next set of tasks in the review process involves the examination of impacts and dependencies to the system that will be changed.

Upstream impacts are factors that feed into the system that must be taken into consideration when developing findings and recommendations. An example of an upstream impact is the SIRS

file transmission from P2K to Unitech ACRplus. The impact factor to making a change to the Unitech ACRplus system that would require it to be off-line is that the change implementation either has to be made during a maintenance window when the P2K system is not scheduled to transmit an SIRS file, or to make arrangements with CBIS to transmit the file at a different time.

The same example has potential downstream impacts: if the SIRS file transmission is rescheduled, processes that require information in the SIRS file will also have to be rescheduled. This can also impact business operations. The extent of potential impact, from upstream and to downstream, needs to be taken into account during the review process.

Constraints fall into the following categories:

- Risk
- Internal (business operations, technical considerations, etc.)
- External (regulatory, interconnected supplier and customer systems, etc.)

The following areas should be examined when considering the risks and other constraints inherent in implementing a particular change to the system:

- Sources of risk (technical, cost and schedule)
- Potential risk events
- Cost estimates
- Activity duration estimates

An overview of techniques for accomplishing risk and constraint assessments is provided in later in this section.

Determine Implementation Parameters

Examination of the following factors and issues will determine applicable implementation parameters:

- Risks and constraints
- Dependencies (up and downstream, impacts on the existing baseline)
- Resource availability (personnel, material)
- SLAs and OLAs

By examining these factors and issues during the review phase a better defined scope will emerge and will set the boundaries for the pre-implementation planning phase.

Evaluate Alternatives

In the case of major risks or impacts, or where the change implementation may (or will) adversely impact attainment of OLAs and/or SLAs, alternatives to the change need to be explored. An analysis of opportunities and threats that will result from not implementing the change should be compared to the opportunities and threats that will transpire if the change is implemented. The alternative evaluation will be included in the reviewer's findings and recommendations in cases where it is appropriate to perform the evaluation.

Submit Findings and Recommendations

The reviewer's report of findings and recommendations will be in the following form:

Table 4.2 Reviewer's report of findings and recommendations

SECTION I. Request validation. What business needs are addressed? Is the requirement redundant?, etc.)
SECTION II. Proposed solution (details of the solution that will meet requirements in the

most efficient and cost-effective manner, and supporting reasons)
SECTION III. Alternatives (What alternative ways to meet requirements were considered? Why were they rejected?)
SECTION IV. Resources (personnel, systems, etc.) that are required to implement the proposed solution.
SECTION V. Dependencies and Risks. How the proposed solution fits within the existing the environment, impact on life cycle and configuration management
SECTION VI. Estimated Level of Effort.
SECTION VII. Resource and life cycle management issues, including cost/benefit analysis.

Sections that do not apply in a particular review, such as Section III for minor changes that do not warrant an investigation of alternatives, will be marked "N/A".

4.2.2.2.2 Vendor-Initiated Changes

This class of change will be handled in the same manner as change requests with the following exceptions:

1. Identification will originate from a vendor and will usually be in the form of a technical notice, field note, upgrade or maintenance release

2. Review will be [usually] conducted by IT subject matter experts (however, it will be appropriate for end user stakeholders to be included in the review process for applications or other changes that impact on business operations or in how the system is presented to the end user)

3. Recommendations against the change must be unanimous among review team members. Unlike a change request, which is closed out in such cases, a recommendation against a vendor-initiated change will be referred to the Configuration Control Board for final disposition.

Vendor-initiated changes often have dependencies that need to be carefully examined during the review process. Examples of common dependencies include:

- Requirement to upgrade other system components (i.e., RAM, firmware revision, etc.) Beware of caveats included in vendor descriptions of the change.

- Interoperability impact on other systems (i.e., will the system still communicate on the network with other vendor equipment after an

upgrade to operating system network extensions? Will [any] third party products currently used with the vendor's products continue to operate?

- Does the change impact on the cost of current service contracts/agreements or level of service?

4.2.2.2.3 IT Changes

The procedures for internal technical changes or fixes differ from change requests in the following ways:

1. The identification process for this class of change may be driven by:
    - Strategic IT plans (i.e., a planned implementation of a particular technology or upgrade to a new version of an existing technology)
    - Non-availability (through product constraints or obsolescence) of system components, spares, etc. that require using alternative equipment or software in order to maintain and operate the system
    - Opportunities for improvement that are normally not visible to end users, but are highly visible to IT (i.e., a module or upgrade to the enterprise management toolset, elimination of known and documented idiosyncrasies in an operating system, resizing network packet lengths for improved transmission efficiency, etc.)
    - Correction of *minor* problems or effecting *minor* enhancements[11]
2. Review is usually conducted by peers within IT, and/or independent verification and validation (i.e., consultants) in the case of major technical changes

[1 1]A *minor problem* would be the behavior of a system, subsystem or component that can be corrected with a minor adjustment from what is documented in the baseline. An example is setting a different base I/O address on an adapter to eliminate a sporadic problem that is known to exist with the current setting. Addressing a problem that *requires* immediate attention in order to safeguard system integrity and/or preserve OLAs and SLAs is not considered to be a change. See the company Policy and Procedures Manual for *Problem and Incident Management* for information and guidance in these cases.

Minor enhancements are changes to the baseline that are relatively trivial in nature, such as changing a few lines in a shell script, but will result in an improvement.

3. Approval from an authority higher than the Configuration Control Board may be required in the case of major changes; approval for minor changes may be granted by internal IT management

Warning: This class of change, especially in the case of a minor enhancement, often does not go through the configuration, change, and release management process, which can lead to discrepancies in the configuration documentation, system problems that are difficult to troubleshoot and rectify because the system is not accurately documented, and life cycle management problems. Reasons for this threat include:

- What appears to be a trivial task with little or no consequence (except an improvement or quick fix) can have unforeseen ramifications from dependencies that the individual who makes the change may not know about
- IT personnel, by their nature, are "fixers" who enjoy solving problems, improving technology and exploring
- It is easier to "just do it and move on" than to go through an administrative process that may be viewed as unnecessary bureaucracy
- Configuration, change, and release management are not a part of the IT culture

Because configuration, change, and release management are major elements in both life cycle management and attainment of OLA and SLA goals, enforcement of policy and procedures for internal technical changes and fixes is imperative.

4.2.2.2 Approving a Change Request

The approval process is designed to ensure that there is an enterprise-wide view of configuration and change management, and that the IT assets are managed in accordance with life cycle and strategic planning. As such all changes that impact IT systems and services will be subjected to a formal approval process.

The procedures for approval are not as specific as those for the other processes covered by this policy because business requirements and priorities, budget considerations, strategic plans and other factors and issues will dictate how approval (or disapproval) is justified.

The procedures for approval are:

1. All requests considered by the Configuration Management Board will be assigned one of the following statuses:

    - Approved
    - Denied

The Configuration Management Board members represent technical and business functions within the company.

Approval grants authority to proceed with the planning phase of implementation. Authority to implement will not be granted until pre-implementation planning (including test plans) have been completed and approved, a budget allocated, and scheduling and coordination have been satisfactorily completed.

Change requests that are denied will be closed out, the initiator notified of action taken and why, and all associated documentation disposed of in accordance with company policy for records retention.

4.2.3 Implementing a Change

Change implementation procedures are grouped into three stages:

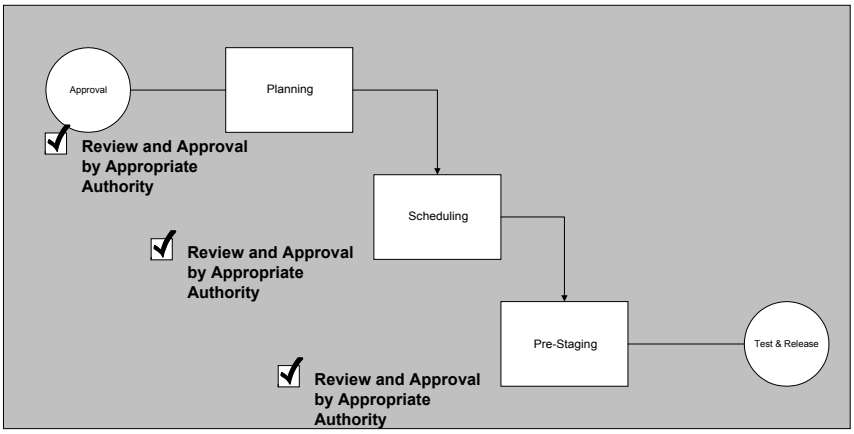
1. Pre-implementation planning
2. Scheduling
3. Pre-staging

Note: All changes, including software and firmware updates, revisions, and manufacturer field changes are subject to pre-implementation planning, implementation and post-implementation procedures.

4.2.3.1 Planning

Upon approval of a change request (or vendor-initiated change or technical change or fix) the following actions will occur in support of pre-implementation planning:

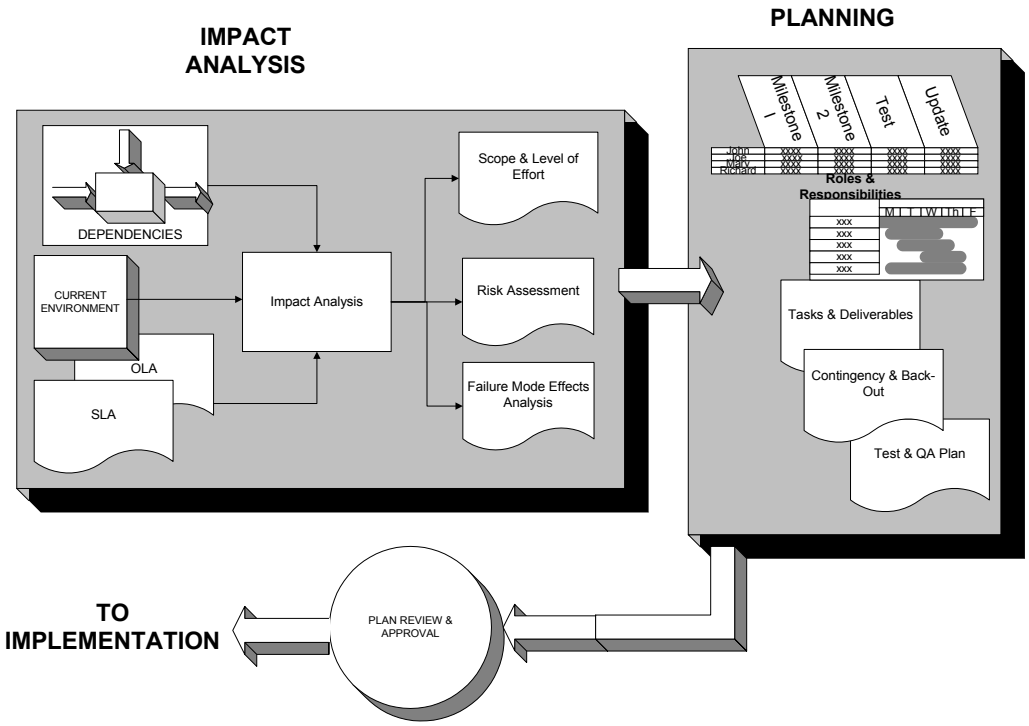
Figure 4.8 Planning procedure



1. Impact analysis to determine the effect of the change on:
    - Current environment (i.e., will the change require an operating system revision upgrade? Is the change compatible with all elements of the applications and hardware suites? Etc.)
    - Related systems (What are the upstream and downstream dependencies? How will the change affect the performance of related systems? Etc.)
    - System resources (Will the change require hardware, firmware or software upgrades?)
    - Operational level agreements (OLAs) and service level agreements (SLAs)
2. Implementation plan of action and milestones, including:
    - Identification of roles and responsibilities
    - Tasks and deliverables related to the implementation
    - Scheduling in accordance with OLAs and SLAs
    - Test and quality assurance plan
    - Contingency and back-out plans

3. Execution of test plan to ensure that the change is stable and suitable for implementation into the production environment

Figure 4.9 Impact analysis and planning procedure



4.2.3.1.1 Tools and Techniques for Risk and Constraint Quantification

Risk and constraint quantification are important elements of project management and should be incorporated into the implementation plan. The following is a list of common considerations:

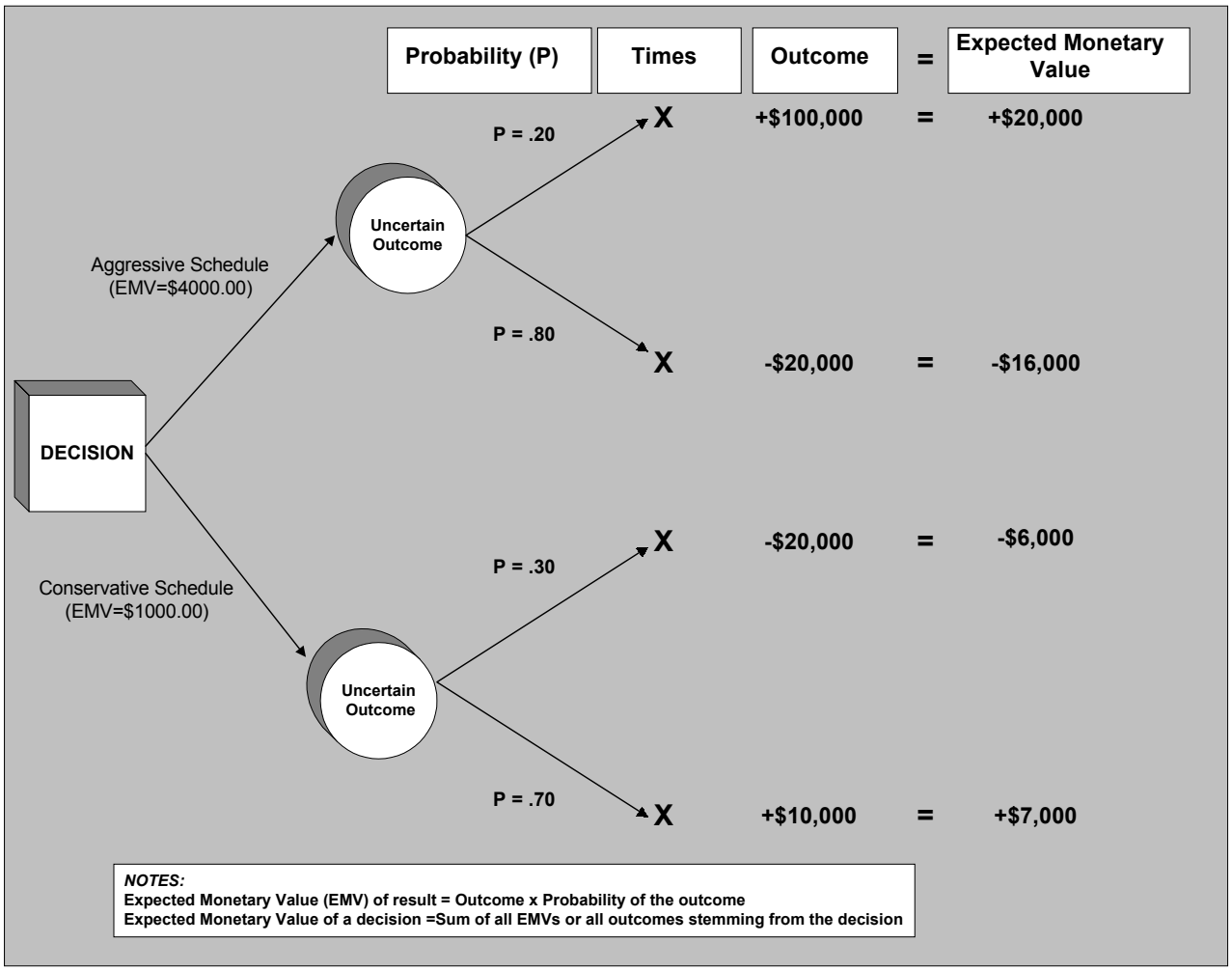
Expected Monetary Value

Expected monetary value, as a tool for risk quantification, is the product of two numbers: *Risk event probability* (an estimate of the probability that a given risk event will occur) multiplied by *Risk event value* (an estimate of the gain or loss that will be incurred if the risk event does occur)

Decision Trees

A decision tree is a diagram that depicts key interactions among decisions and associated chance events as they are understood by the decision maker. The branches of the tree represent either decisions (shown as boxes) or chance events (shown as circles).The following illustration is an example of a decision tree.

Figure 4.10 Decision tree



Expert Judgment

Expert judgment can often be applied in lieu of or in addition to the techniques described above. For example, risk events could be described as having a high, medium, or low probability of occurrence and a severe, moderate, or limited impact.

4.2.3.1.2 Outputs from Risk or Constraint Quantification

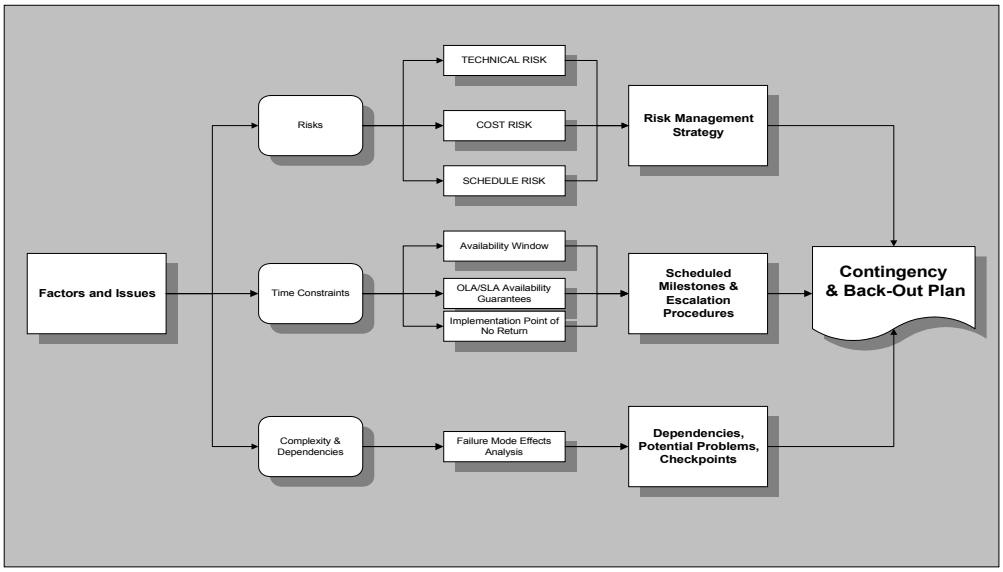
Quantification of risks and constraints will result in a set of opportunities and threats to be considered when evaluating the change request:

- Opportunities to pursue/threats to respond to. The major output from risk quantification is a list of opportunities that should be pursued and threats that require attention.
- Opportunities to ignore/threats to accept. The risk quantification process should also document (a) those sources of risk and risk events that the project management team has consciously decided to accept or ignore and (b) who made the decision to do so.

4.2.3.1.3 Back-Out and Contingency Strategy

One of the most important elements of the implementation plan is the back-out and contingency strategy. If this strategy is not developed and included in the implementation plan a failed change could result in unplanned system downtime with ramifications that extend to customer service, lost revenue, etc.

Figure 4.11 Back out and contingency factor procedure



Procedures for developing a back-out and contingency strategy are:

1. Examine all risks and plan for how the occurrence of each risk will be managed during the implementation process

2. In the case of complex changes that have multiple dependencies a failure mode effects analysis (FMEA) should be conducted to ensure that all possible problems, dependencies and failure modes are identified and incorporated into the implementation plans

3. Identify critical milestones and escalation points, and the point during the implementation that is considered to be the point of no return (the point beyond which the system cannot be restored to full operational capability in support of business operations if a problem occurs)

Failure Mode Effects Analysis (FMEA)

Because the information technology infrastructure is both complex and dependent upon systems outside of the enterprise (i.e., CBIS) FMEA is a valuable tool to employ during the development of a back-out and contingency strategy.

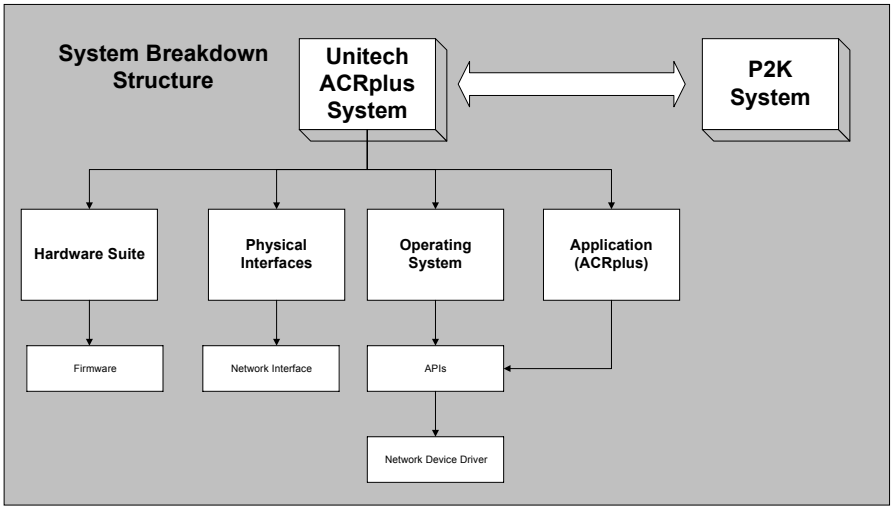
To performing a failure mode effects analysis:

1. Develop a breakdown structure for the system being analyzed to the subsystem and component level
2. Interconnect the components with Boolean logic symbols to show conditions needed to cause a failure
3. Find weak points in the diagram, such as single points of failure, points of failure that can impact multiple related systems, subsystems or components
4. Develop:
    - A strategy to mitigate potential problems, and/or
    - Critical milestones on the implementation timeline that denote escalation points and the point of no return

Failure Analysis Example. To illustrate these procedures, a simplified example is provided.

Change Scope	Apply an OS patch
System	Unitech ACRplus
Components	Operating system, hardware suite (including network and communications interfaces), applications suite
Dependencies	Must communicate with P2K • Required to be ready to receive SIRS file transmissions from P2K between 06:00 and 18:00 weekdays OS patch depends on: • System firmware • Network interface device driver revision level Application makes OS service calls through an API

Figure 4.12 System breakdown structure

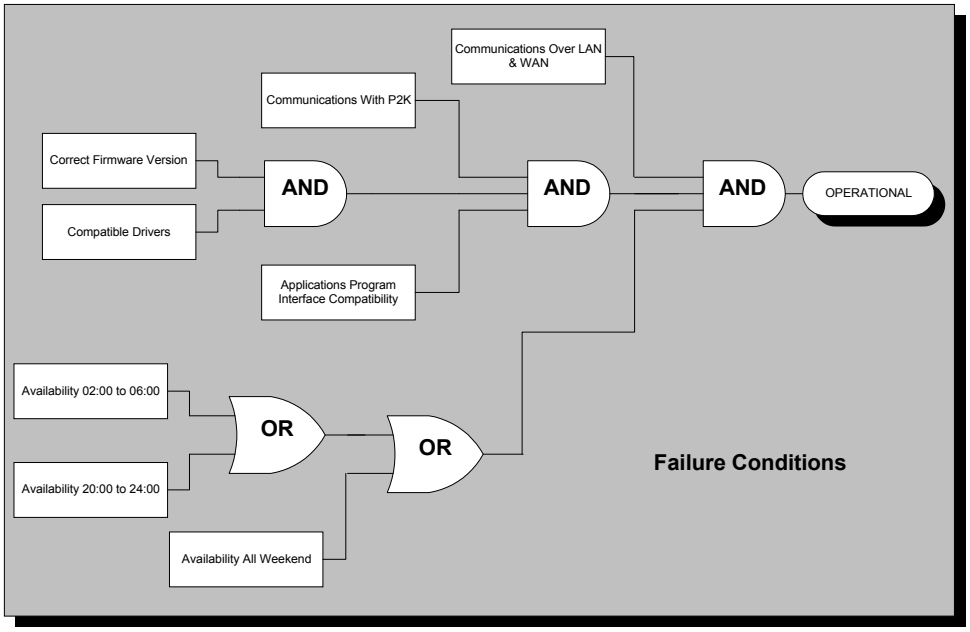


1. Develop the system breakdown structure that shows all of the systems, subsystems and components that fall within the scope of the change. This will provide a clear picture of dependencies and relationships, with a focus on what is most likely to fail during the implementation.

2. Arrange the systems, subsystems and components in a logical diagram to depict the conditions are required for a fully operational system after the implementation has been affected.

   Note that in this example not only have hardware and software factors been considered, but availability windows have been included in the analysis as well.

Figure 4.13 Failure conditions



3. Examine weaknesses and develop methods to rectify or minimize the weaknesses. In this example, the planner feels that more than four hours to safely perform the implementation, conduct a thorough test and release the system back into production. The weakness is the two four-hour availability windows. These are not sufficient time frames within which the implementation can be accomplished, and are considered to be potential failure points. In this case the most prudent approach is to schedule the implementation to occur during a weekend.

   Also note that there are many single points of failure in the above example. Anytime a condition is based on AND logic failure points should be assumed and carefully examined. In the example, there are six mandatory conditions that must be met or a failure will occur. Each of these conditions needs to be individually addressed and planned for. For example, the firmware version requirement can be eliminated by verifying the current firmware version level and ensuring that it is either at the correct version level or replaced in connection with the change.

4.2.3.2 Scheduling

The final step in the process is to develop a timeline with critical milestones and escalation points. This information will be used to schedule and control the implementation activity.

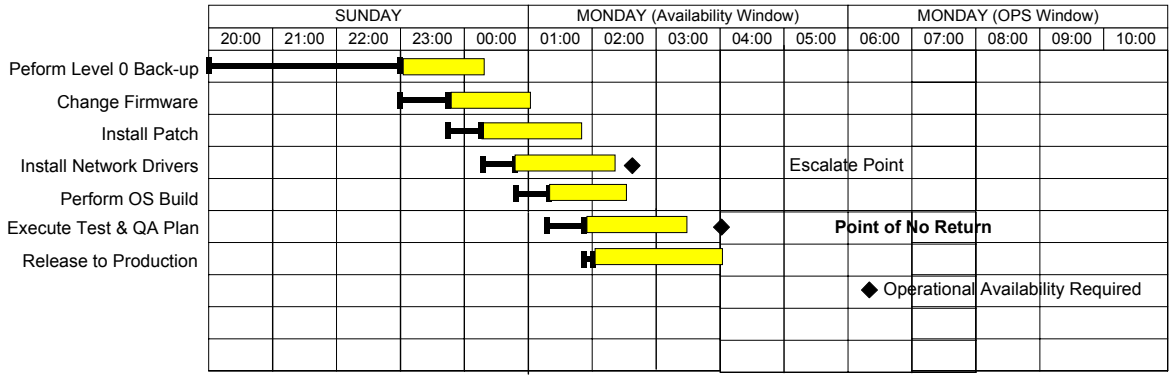
Assume that the implementation has been determined to take 6 hours under ideal conditions, and that the worst-case estimate is 8 hours. Assume also that the implementation team has been scheduled to commence work at 20:00 on Sunday. Under ideal conditions the work will be completed by 02:00 Monday, and under estimated worst-case conditions the job should be completed by 04:00, leaving a 2-hour safety margin for daily operations.

Assume also that the following are activities required to implement the change:

1. Perform a Level 0 back-up
2. Change firmware
3. Install patch
4. Install new network drivers
5. Perform an OS build
6. Execute Test & Quality Assurance Plan
7. Release system to operations

In the event that there are problems with the implementation the team will boot under the previous OS build. The timeline at this point will be as depicted below:

Figure 4.14 Problem time line



This timeline shows start and expected completion times for each task, worst-case times, an escalation milestone, point of no return and a milestone depicting when the system must be available for operations.

The following worksheet provides a means of listing and classifying all risks and impacts, and quantifying all costs and other constraints that are germane to implementing a change to the system:

Figure 4.15 Risk and impacts worksheet

Upstream Dependencies	**Downstream Dependencies**
☐ **Time** _____ ☐ **Event** _____ ☐ **Hardware** _____ ☐ **Software** _____ ☐ **Communications** _____ ☐ **SLA/OLA** _____ ☐ **Other** _____	☐ **Time** _____ ☐ **Event** _____ ☐ **Hardware** _____ ☐ **Software** _____ ☐ **Communications** _____ ☐ **SLA/OLA** _____ ☐ **Other** _____
Comments	
Impact	**Estimated Costs**
Schedule ☐ **Minor** ☐ **Moderate** ☐ **Major** **Cost** ☐ **Minor** ☐ **Moderate** ☐ **Major** **Technical** ☐ **Minor** ☐ **Moderate** ☐ **Major**	**Direct labor costs to effect requested change: _____ hours @ $_____/hr.** **Total: $ _____** **Indirect labor costs to effect requested change: _____ hours @ $_____/hr.** **Total: $ _____** **Other Costs: $_____** **Impact on current milestone: _____ Hours** **Impact on project schedule: _____ Days**
Risk Mitigation Opportunities: **Drop Dead Time In Implementation Schedule:_____** **Additional Impacts:**	**Itemized Additional Costs and Estimated change in technical approach:**

4.2.3.3 Implementation

The following will govern how the implementation phase is accomplished:

- Plans developed during the planning phase
- OLAs and SLAs
- Test and release procedures

Upon completion of all implementation phase activities the following steps will be accomplished:

1. Update system configuration documentation
2. Update all applicable documentation, logs and records
3. Close out the action
4. Notify the initiator and all other concerned parties

Items that are easy to overlook, but have a significant impact on configuration and change management (and system operations) are:

- Updating all affected records *and* documentation
- Adjusting performance metrics to reflect any improvements that were realized from the change
- Disseminating information about the change, including realigning any training to reflect changes made to the system, its operation and its maintenance
- Developing a file of "lessons-learned"

4.2.4 Testing and Releasing a Change

The test and release procedure covers two areas: quality assurance and pre-release activities.

4.2.4.1 Quality Assurance

The principal objective of test and release procedures is quality assurance. A tenet of quality assurance is that individuals who perform quality inspections and sign-off are not the same individuals who accomplish the implementation. This

does not preclude members of the implementation team from functioning as quality assurance inspectors with final sign-off authority. It does preclude inspection and sign-off of a particular task by an individual who actually performed the task.

Company policy and procedures for test and release requires that the following signatures be on any final test and release sign-off before a change is placed into production:

1. Quality assurance inspector
2. System owner or designated representative
3. IT Director or authorized representative

4.2.4.2 Pre-Release Activities

Test plans shall be developed to accomplish the following objectives:

1. Exercise all systems, subsystems and components affected by the change to be implemented
2. Have a clear audit trail
3. Provide certification that the system was properly and adequately tested and found to meet all applicable operational requirements and conform to all company standards
4. Clearly identify the system(s) to which the change was made
5. Provide an operational checklist that ensures the following items, at a minimum, are accounted for:
    - All open severity level problems are properly closed
    - Product loadouts (media, documentation, certifications associated with the change implementation; i.e., materials provided by vendors to initiate a change or upgrade)
    - Applicable documentation that describes the system is identified so that changes can be made to ensure that the baseline is accurately described
    - Test plan summary

4.2.4.3 Release

Upon successful completion of the test and quality assurance plan, including all required signatures to certify that the system is fully functioning in accordance with operational and technical parameters, the system will be scheduled for release into production. At the scheduled release milestone and 48-hours of continuous error-free operation the changed system configuration will be deemed to be the operational baseline and recorded in the system configuration documentation per status accounting procedures.

4.2.5 Status Accounting

Any change to the system under configuration management shall be documented in the:

1. Configuration management documentation
2. Applicable systems documentation

Change is defined as any modification to or alteration of the system baseline. The system baseline includes, but is not limited to, the following systems, subsystems and components:

- Baseline hardware suite
- Peripheral devices
- Firmware
- Software, including patches, scripts and kernel builds
- Data structures
- Security profiles
- Logical schemes and numbering assignments
- Asset identification information
- System location (to reflect M/A/C activity)

All changes will be recorded and made a part of the operational baseline after a period of 48 hours of error-free operation in production.

Table 4.3 shows how systems managed under other baselines will be documented.

Table 4.3 System managed under other baselines

Baseline Type	Description
Functional Baseline	Each agreed upon functional specification will be recorded within 24 hours of inclusion. A revision history will be maintained for traceability purposes and will become a permanent part of the system's life cycle documentation. Functional baseline information is usually in the form of business and/or process studies, findings and preliminary recommendations.
Allocated Baseline	Each technical specification incorporated during the design phase of the system's life cycle will be recorded within 24 hours of acceptance. Information such as capacity planning estimates, assumptions and other notes and analyses used to develop the allocated baseline will become a permanent part of the system's life cycle documentation.
Product Baseline	The product baseline begins as soon as the allocated baseline is frozen. All changes will be recorded within 24 hours during the pre-ORT phase and as they are implemented during the ORT phase. Anticipate a large number of changes in during the ORT phase, as well as for the first two-four weeks after the system has evolved into operations.

Also a part of the status accounting procedure set is change coordination. This process begins when the change request is initiated, but functionally comes under the status accounting process. Change coordination provides status information on new, pending and closed change requests. Table 4.4 provides status categories and descriptions:

Table 4.4 Status accounting categories and description

Status Category	Description
New	The change request has been received and logged; has not been received by Configuration Control Manager for action
Assigned	Received by the Configuration Control manager and has been assigned to a review team
Reviewed	Has been reviewed, pending final decision (approval/denial)
Approved	Request has been approved by the Configuration Control Board (or other designated approving authority for vendor-initiated and technical improvement/fix changes)
Denied	Request has been denied
Confirmed	The implementation team has finished developing all pre-implementation procedures and has confirmed readiness to effect the change
In progress	Implementation of the change is in progress
Completed	Implementation of the change has been successfully completed
Aborted	Implementation was aborted due to technical problems, failed quality assurance test, etc. Back-out plan was executed and system is

Status Category	Description
	restored to pre-implementation condition
Closed	CASE Implementation passed 48 continuous hours of error-free operation and all status accounting procedures have been completed (1) Change request denied, close-out procedures completed (2) Change aborted, decision to make no further attempts made

4.3 Auditing the Change Process

The system baseline shall be audited to ensure the accuracy of the system configuration documentation and [any] other related documentation on an TBD basis. Table 4.5 is a recommended audit schedule:

Table 4.5 Recommended audit schedule

System, Subsystem or Functional Area	Audit Category	Scope	Recommended Interval
Data center CPUs	Hardware Configuration	**Items to be audited**: CPUs and all attached internal and external peripherals. **Level of detail**: manufacturer/model, manuf. serial #, asset number, filed and open slots and ports, system hardware settings and used/available resources, firmware revision level, network interface MAC address(es)	At each baseline milestone and every 12 months after the initial operational baseline audit
Data center peripheral devices	Hardware Configuration Embedded Software Configuration	**Items to be audited**: peripheral devices not attached to a data center CPU (i.e., network attached printers, communications devices, enterprise mass storage systems, etc.) **Level of detail**: manufacturer/model, manuf. serial #, asset number, device specific resource and option information (including configuration settings), firmware revision level, [any] associated MAC addresses, boot images (including patch and revision levels)	At each baseline milestone and every 12 months after the initial operational baseline audit
Data center CPUs	Software Configuration (OS Level)	**Items to be audited**: operating systems, system utilities, network extensions and device drivers **Level of detail**: (for each software component) configuration options information (linked devices, init options, etc.), software version, revision level, patch history, file system layout maps, initialization information (init and mount tables; networks, hosts and IP numbers, etc.), system-wide shell scripts	At each baseline milestone and every quarter after the initial operational baseline audit
Applications Software	Software Configuration	**Items to be audited**: licenses, software components and associated file systems **Level of detail**: license compliance, software version, revision level, patch history, file system layout maps, special configuration options	At each baseline milestone and every quarter after the initial operational baseline audit

Database	Software Configuration	**Items to be audited**: licenses, software components, schemas, business rules and security, and triggers **Level of detail**: license compliance, software version, revision level, patch history, schema, embedded logic, application-specific security profiles	At each baseline milestone and every quarter after the initial operational baseline audit
Documentation	Documentation Configuration	**Items to be audited**: User and system documentation, policy and procedure manuals, configuration logs, run logs, program listings **Level of detail**: Verify that documentation is aligned to current configurations, applications, operating systems, policies and procedures	In conjunction with related hardware and software audit. Policy and procedure audits conducted on an annual basis.
Security	Documentation Configuration Software Configuration	**Items to be audited**: systems and applications security profiles, and associated security mechanisms **Level of detail**: Validate documentation and security mechanism configurations are in accordance with policy, procedure and baseline documentation/configurations	In conjunction with related software audit. Security policy, procedure and compliance audits conducted on a quarterly basis.
Disaster Preparedness	Documentation Configuration	**Items to be audited**: plans, physical and logical safeguards, associated documentation (appointment letters, responsibility matrices, vendor contact information, etc.) **Level of detail**: validate system plans, test and evaluations, configuration of physical and logical safeguards and associated documentation	TBD

4.4 Improving the Configuration, Change, and Release Management Process

As conditions change from time to time, aspects of the configuration, change, and release management process itself will require modification. For example, if a particular process makes it impossible to meet a given service levels, the process may come under scrutiny.

The resolution process for this kind of changes is described below.

As conditions change from time to time, aspects of the configuration, change, and release management process itself will require modification. For example, if a particular process makes it impossible to meet a given service levels, the process may come under scrutiny.

Continuous improvement of the Configuration, Change and Release Management policy and procedures is supported by two methods:

1. Expiration date on this document
2. Change process as described in this policy and procedures manual

This document, like all company policy and procedures manuals, has an expiration date. This enforces a periodic review of the policies and procedures contained herein to ensure that the content will continue to reflect company business and technical practices. Three months prior to the expiration date the document will be reviewed, and any discrepancies between the documented policies and procedures and actual practices and requirements will be reconciled by modifying the applicable sections of this manual. This review process is owned by the Configuration Manager. Reviewers are process owners who are affected by configuration, change and release management policy and procedures.

Because this document subject to the policy and procedures set forth for configuration, change and release management, it can be modified or amended by employing the change request procedures described in this manual. At any time a policy, process or procedure ceases to reflect company business or technical objectives, is deemed to be inefficient or ineffective, or can be otherwise improved upon a change request is warranted.

The design of this document lends itself to process improvement. Each process has been documented using a standard process notation that describes input, output, controls and constraints for each of the processes supporting configuration, change and release management. As such, the processes have been isolated in such a manner that the underlying procedures of any single process can be changed without impacting processes that provide input or receive output.