

SFA Enterprise Configuration Management Approach

SFA Modernization Program

US Department of Education

Table Of Contents

1	Introduction	1
1.1.	Purpose	1
1.2.	Scope	1
1.3.	Current State of Configuration Management	2
1.4.	Document Organization	2
1.5.	Document Development	3
2	Configuration Management	4
2.1	Configuration Management Processes	4
2.1.1	Administer CM Environment	7
2.1.1.1	Change Management	7
	Version Control	9
2.1.1.2	Migration Control	9
2.1.1.3	Baseline Control	10
2.1.1.4	Release Management	10
2.1.2	Configuration Management Process Components	12
2.1.2.1	Change Request	14
2.1.2.2	Plan Release	14
2.1.2.3	Manage Configuration Items	14
2.1.2.4	Install Configuration Items	15
2.1.2.5	Implement Configuration Items	16
2.2	Organization and Responsibilities	16
2.2.1	Configuration Management Team	17
2.2.2	Enterprise Architecture Team (EAT)	18
2.2.3	Development Team	18
2.2.4	Configuration Control Board (CCB)	18
2.2.5	Investment Review Board (IRB)	19
2.3	Configuration Management Tools	19
2.3.1	Tool Selection Criteria	20
2.3.1.1	Software Version Control	20
2.3.1.2	Software Distribution	21
2.3.1.3	Change Management	22
3	Next Steps and Implementation Plan	24
3.1	Next Steps	24
3.1.1	Identify the Implementers of Enterprise Configuration Management	24
3.1.2	Define Scope of Implementation and Timeline	24
3.2	Implementation Plan	25
3.2.1	Assemble Organization for Configuration Management	25
3.2.2	Establish an Environment on Pilot Platform	25
3.2.3	Define Configuration Items	25
3.2.4	Develop Proof of Concept	26
3.2.5	Certify Environment	26

3.2.6	Support Environment	26
3.2.7	Train Personnel in CM Processes and Tool	26
3.2.8	Measuring CM	28
4	Appendix A: Candidate Forms	30
5	Appendix B: Candidate Checklist	34
6	Appendix C: Configuration Items List	37
7	Appendix D: Detailed CM Process Flows	44

1 Introduction

1.1. Purpose

In order for the Student Financial Assistance (SFA) Modernization Program at the US Department of Education to accomplish the objectives of the performance based organization (lower operational unit costs, higher customer and employee satisfaction), it will require reengineering of technical processes and technical architecture. This level of activity will require a significant amount of effort to coordinate, monitor progress, track updates and validate the technical enhancements. To accomplish this, the SFA Modernization Program requires a support structure to provide the oversight and coordination of modernization capability release activities (to the SFA executive team and stakeholders). This is required so the right decisions can be made to achieve performance objectives related to planned application and/or system capability releases. For this document, a capability release can be defined as an SFA initiative and/or project, that has the potential of providing SFA business value, while helping SFA achieve the mission outlined in the Modernization Blueprint.

This approach identifies the need for an enterprise configuration management function that will maintain focus on the overall technical and functional objectives of the program. This enterprise configuration management function will also provide the continuous guidance needed to support the delivery of SFA's targeted business capabilities.

Implementing an enterprise configuration management structure will provide senior management with an oversight ability that is lacking in the current lower level and partitioned model of configuration management. In addition, this enterprise perspective will support the Modernization Partner by working with the SFA Chief Information Office (CIO) Enterprise IT Management, IT Services, IT Application teams, and the virtual data center to standardize configuration management practices across channels.

Upon the conclusion of reading and understanding this approach, the reader will understand the necessity of creating an enterprise focused configuration management organization. To jump start this process, the reader has been supplied a series of next steps outlined at the conclusion of this approach. These steps will provide the reader with the requirements for implementing the organizational structure and the associated enterprise configuration management concepts.

1.2. Scope

The configuration management approach applies to all information systems and related system engineering activities that might affect the achievement of the SFA Modernization effort. This would include hardware, software (commercial off the shelf software (COTS) and/or custom), and documentation. In particular, the focus of this document is on the enterprise perspective of configuration management.

1.3. Current State of Configuration Management

SFA currently performs configuration management in two ways: controlling infrastructure (project and data center) and by controlling the individual application/system software components. SFA does not have an enterprise level configuration management process. Nor does it have an integrated (software, hardware, infrastructure) approach throughout the software development life cycle. Generally, an SFA system is configured and managed by a third party contractor organization. These organizations were tasked with establishing and maintaining the configurations for multiple systems. Because each contractor organization has different methodologies and practices, there is no uniform configuration management approach across channel lines.

The two approaches to controlling configuration have implemented review boards to help in the decision making process regarding migration of components and application releases. The boards function similarly but their objectives are very different. The Configuration Control Review Board (CCRB) is run by the Virtual Data Center(VDC). It meets once a week and focuses on infrastructure requirements at both the project and data center levels. This board is not dedicated to SFA concerns. It is comprised of thirty-nine other Virtual Data Center clients plus the Department of Education. Several of the Virtual Data Center clients are competing with each other. The Virtual Data Center has an established privacy policy. As a result, no information (practices, procedures, lessons learned) is being shared between clients. The Software Configuration Control Board (SCCB) is run by the SFA project/application teams. It meets on an infrequent basis. Its primary concern is with application/functional requirements. In addition, there are individual application development team meetings to discuss day-to-day software/hardware configuration control issues. Note the agenda could include procedures, standard responses and other production related activities. Its at these meeting that new infrastructure requirements are revealed. As a result the Virtual Data Center is tasked, by the development application teams, with implementing those requirements.

As mentioned previously, there is no enterprise level organization within SFA that oversees the activities of these two boards. Additionally, there is no representation by senior management level to provide input on release capability dependencies and/or impacts prior to a proposed application release reaching the product testing phase of implementation. All these things point to a need for establishing and/or redefining configuration management with an enterprise (top-down) view.

1.4. Document Organization

The configuration management approach includes the primary responsibilities of configuration management in the system development life cycle process. Additionally, the steps required to fulfill these responsibilities, plus a high level design of the process and the organization structure that supports the configuration management are included.

- Section 1: Describes the overall purpose of configuration management of the scope of this approach
- Section 2: Describes the configuration management key concepts, stages (change request, plan release, migrate configuration items, administer configuration management environment, manage configuration item, implementation, tracking and monitoring configuration items. This section also describes the organization and the responsibilities allocated to each element of the organization for configuration management. The functions that CM tools need to fulfill is covered in this sections, as is the relationship of CM with other processes and organizations.
- Section 3: Describes the next steps required to implement an enterprise-wide Configuration management Process.
- Section 4: This contains the appendices. The first two portions of the appendix are devoted to candidate forms and checklists. They can be used as a start on deriving fully customized documents for SFA. The other two portions are referred to within this document. They contain a discussion on configuration items and a series of process flow diagrams of the configuration management approach presented in this document.

1.5. Document Development

The following organizations and individuals were source of information in writing this approach:

- SFA Enterprise IT Management: Denise Hill
- SFA Enterprise IT Services: David A. Elliott
- SFA E-Commerce Application Development: Mike Rockis
- SFA CIO Chief Scientist: Constance Davis
- SFA CIO Business Manager: Harry Feely
- CSC: Wayne Burgess
- CSC: Jerry Ryznar
- NCS: Chris Ledman
- EF Kearney Limited: Seth Baldwin

The following sources were also used:

- Andersen Consulting Configuration Management Best Practices
- Andersen Consulting Business Integration Methodology
- *Implementing Configuration Management: Hardware, Software, and Firmware*, Fletcher J. Buckley, IEEE Computer Society Press, 1992

2 Configuration Management

Configuration Management (CM) enables the controlled and repeatable management of Information Technology (IT) architecture components as they evolve in both development and production environments. CM implements a process by which the various enterprise IT management organizations, project teams, and business stakeholders communicate, implement, and document changes in the systems environment. When properly implemented, CM provides efficient and prompt handling of all change requests. The purpose of this configuration management approach is to establish a sound configuration management process that maintains the integrity of SFA systems and provide traceability for changes incorporated into the environments.

The configuration management process integrates the technical and administrative actions of identifying the functional, performance and physical characteristics of a configuration item (CI) and changes to those characteristics. CM provides information on project status and project control for each configuration items. The objective of configuration management during these phases is to establish and manage baselines, to produce the lowest overall project life cycle cost, optimal operating efficiency, and readiness of a configuration item. Configuration Management is the means through which the integrity of the design, development, and cost trade-off decision are made between technical performance, operability, and supportability. These aspects of configuration management are recorded, communicated, and controlled by program and functional managers. Configuration Management reporting allows the project members to know exactly what is in each release.

2.1 Configuration Management Processes

An enterprise perspective configuration management program requires a number of important steps be taken to realize its benefits. This section of the approach summarizes those steps. The backbone for any good approach are a set of guiding statements or principles. For the purpose of this document these statements are referred to as key concepts. These concepts are described briefly in the section to follow. The second step involves discussing the configuration management technical processes. Thirdly, there needs to be some discussion on the organizational requirements necessary for the success of the enterprise. And finally, after having understood the concepts, technical processes and required organizational structure, tools to do the job are the only things missing from the equation. The last section will provide a discussion on what types/tools are required to operate the enterprise configuration management organization.

2.1.1 Administer CM Environment

To administer the environment properly, maintaining the status of the system configuration and its configurable items must become a critical component of the day-to-day process. How do we ensure the status of configuration item is accurate? Through a tracking and monitoring program. Tracking and monitoring involves the recording, monitoring, and reporting of all configuration item changes to established baselines. Status accounting provides information about each change to those with a need to know. The purpose of configuration status and reporting is to answer the following questions:

- What happened?
- Who did it?
- Why did it happen?
- When did it happen?
- What else will be affected?

The flow of information for configuration status reporting is tied to the identified tasks or configuration items for each project. Each time a configuration audit is conducted, the results are reported as part of the status reporting. Each time a configuration item is changed, it must be tracked to assure that updates to the documentation is disseminated. The implementation of approved changes is tracked, and if problems occur, the appropriate configuration item (requirements, design, and code) can be identified. Status reporting must be able to accurately report the configurations of multiple configuration items during development and in the installed baseline. Configuration item status reports are generated on a regular basis to keep management and software engineers apprised of project status. These reports represent the methodology the project uses to obtain the information necessary to perform the status accounting and reporting function.

Status accounting is the recording activity. It follows up on the results of the configuration identification, control, reporting, and auditing activities. Status accounting keeps track of the current configuration identification documents, the current configuration of the delivered software, the status of changes being reviewed, and finally the implementation of approved changes. Status accounting refers to the record keeping function of CM and exists to provide all the technical information about the software configuration.

The key concepts of administering an enterprise CM environment are described below.

2.1.1.1 Change Management

The concept of change management focuses on managing modifications to the configuration items.

The primary concern is that all modifications made must be controlled and predictably managed. It further states that project costs, schedules and technical requirements should be established at specific milestones and placed under formal configuration management governance. Changes to these baselines would require a formal action. Thus, these proposed changes could only be initiated through a change request form.

Version Control

There is a concept in configuration management that identifies and manages an individual component of a software system as it changes over time. The individual component is referred to as a configuration item (CI). The concept or process used to manage these configuration items is called version control. It is important because it helps to ensure that an accurate component audit trail is maintained. By examining the contents of the audit trail you should find the modification time and date, the author's name, and the reason for which the modification was carried out.

As systems development continues, it is important to keep a history of the changes made. Following a set version control methodology at all stages of a system's life cycle would enable the developers to always have access to programs as they existed in the past.

This is required whenever more than one version of a configuration item must be supported at any point in time. Version control requires that steps are taken to ensure that development tasks are performed on the correct version.

System development may require multiple versions of a configuration items co-exist within the same overall SFA version development effort. For example, software modules may be updated in the construction environment while the previous version continues to be used in the product test environment. This co-existence is temporary - only one version of the configuration item makes it into the final assembly of that SFA version.

Each version is comprised of a specific assembly of SFA configuration items. Initially, each incremental release delivers a new version of the SFA solution which carries the same number as the incremental release e.g. Increment 2 delivers SFA V2.0. This version may be enhanced through maintenance creating subsequent versions, each of which is identified by a unique number within the overall major version e.g. V2.1, V2.2, V2.3.

Another thing to mention is the cost factor of maintaining multiple versions. This can be significant throughout the system development (and deployment) life cycle maintenance period. It should not be underestimated, when it could involve massive amounts of data. In particular, with government regulatory requirements for archiving reports, audit files, databases, the cost of maintaining could be an overwhelming factor in the type of version control you choose to implement. Most detailed discussions on this topic should be held during the implementation phase of configuration management.

2.1.1.2 Migration Control

To facilitate development, a standard CM environment (platform) should be chosen and followed throughout the life-cycle of the project. However, multiple environments should be established to reduce the possibility of contention between different tests competing for the same resources and data. To minimize the re-creation of system components in subsequent environments, components are

migrated from one environment to another. To facilitate this, a migration release strategy should be developed and followed by each project.

The development environment should be capable of supporting one or more autonomous development teams during the application development phase. They would essentially have overlapping development, testing and production release phases. This is called parallel development, as depicted in Figure 2.1.1.3 below.

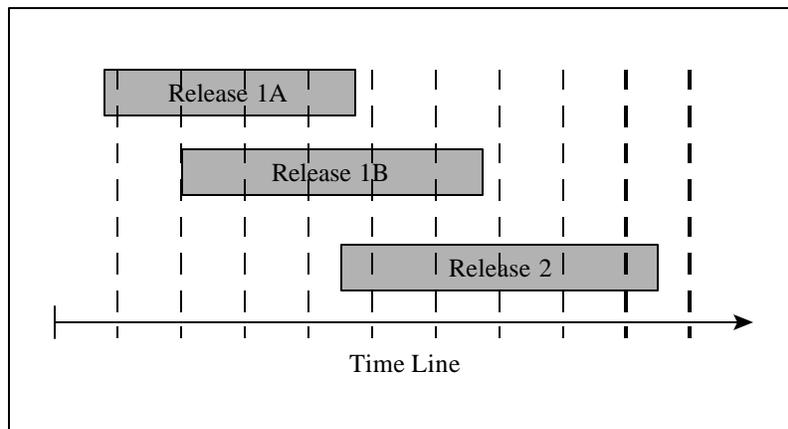


Figure 2.1.1.3 - Parallel Development

2.1.1.3 Baseline Control

Configuration baselines are defined at points in time when all configuration items are “frozen” (no additional modifications can be made). This is done to promote project-wide change control and to provide a stable environment for the accurate development of subsequent deliverables. The project scope and solution components form the basis for establishing the configuration baselines. A copy of the initial baseline will be made and saved for reference during the life of the project. In addition, the projects have multiple baselines that should be saved and referenced at each major milestone review.

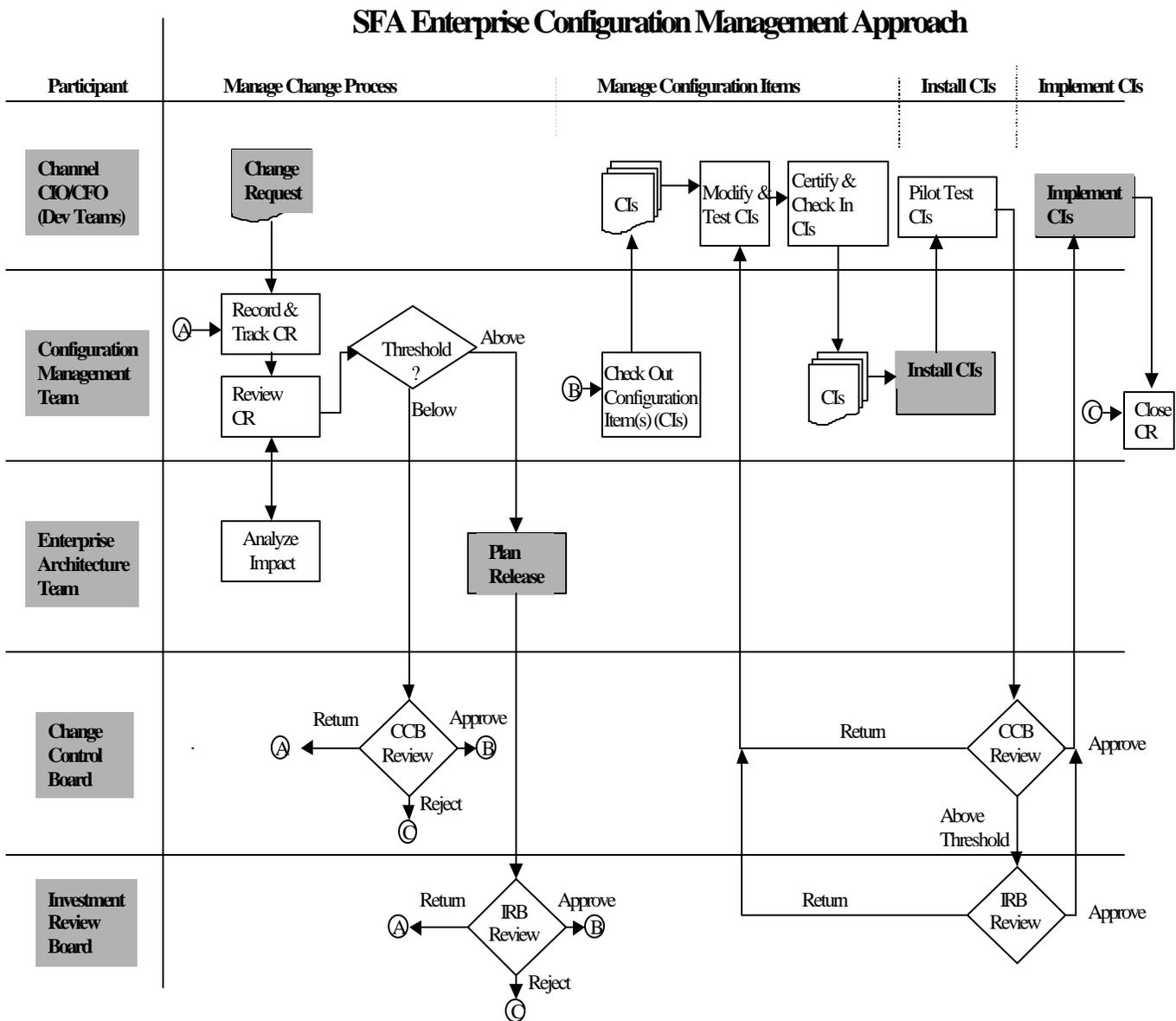
2.1.1.4 Release Management

The release management concept ensures the coherence of system objects and their relationships when the system is deployed. This should be carried out by a deployment team member, supported by the configuration management coordinator within the application development team. It includes:

- Initial planning of features that must be included in the release
- Ensuring compatibility between the constituent components of the release.
- Distributing system software, custom application software, application software packages and data to the appropriate nodes in the network
- Verifying that all components that constitute the release have been delivered to the correct place.

2.1.2 Configuration Management Process Components

The configuration management process exists throughout the full system development life cycle. It can be broken down into six stages. They are: requesting change, planning for release, managing configuration items, controlling the migration of configuration items, installing the configuration items, implementing the modified configuration items in a controlled manner. The following diagram describes the required interactions and responsible organizations involved in this process.



2.1.2.1 Change Request

All strategic and/or enterprise level change requests are made in accordance with the business requirements of the Investment Review Board. Change requests can be initiated by SFA business area (i.e., the channels or the CIO, Modernization Partner, legacy vendors) with the assistance and participation of the enterprise architecture team. The configuration management team is responsible for recording and responding to the change requests.

Change requests that are below threshold criteria, require review, support and a decision by the CCB. Upon analysis and agreement by this group the configuration management team will respond to the change requests submitted. This process is outlined in Appendix D. Change requests that are above the threshold criteria will be analyzed via the plan release process. If a “go” decision is made the change is assigned to an existing development team to begin the process of managing the configuration.

For a more detailed diagram of the change request process flow, see Appendix D.

2.1.2.2 Plan Release

The plan release process begins with the enterprise architecture team defining the release objectives and if necessary, developing the business case/cost benefit analysis associated with this proposed release. The enterprise architecture team attempts to assign the change requests to a specified release. The accumulated data is packaged for delivery to the Investment Review Board/Decision Support Group (IRB/DSG) for further definition of the proposed release justification and contents. The proposed release (change request(s), supportive documentation) is sent to the IRB/DSG for deliberation. Upon approval of the change request(s). The last step in this process is to transition the documentation from the CM team to a new IPT or development team to begin the process of managing the configuration.

For a more detailed diagram of the change request process flow, see Appendix D.

2.1.2.3 Manage Configuration Items

This section of the configuration management process combines the concepts of managing the configuration items and migrating them throughout the system development life cycle. They have been bundled in this streamlined approach to minimize the number of organizations involved to successfully navigate the manage configuration process.

This process depicts the day-to-day activity of the individual application development teams. During the development life cycle, the development team members will work with their CM coordinator/configuration management team in accessing the individual configuration items to conduct their development efforts. The process is as follows: the configuration management team has control of the configuration items (baseline). In order to work on them, the development team issues a

request via the configuration management tool, to check out a specific version of the configuration item. The tool will keep track of who has the configuration item checked out and will not allow anyone else to check out the same version. The development team will work with these configuration item(s)/modify them. At the end of the development effort, the configuration management team will be notified that an update to the release configuration is necessary. The configuration management team confirms the request with the development team. As part of this process, an audit trail is created. It will contain the modification time and date, the author's name, and the reason for which the modifications were carried out. The configuration management team will then check in the updated configuration items. The CM team creates a new baseline and updates the release configuration documentation.

The migration step in this process demonstrates how configuration items are maintained by the configuration management team throughout the system development life cycle of the development teams. In each phase of development, CM maintains strict baseline control of the application and infrastructure environments. Upon the conclusion of each phase, part of the exit criteria involves creating an interim baseline, generating configuration audits, verification & review, and archiving the previous one (for disaster recovery purposes). In addition, this process initiates tracking and monitoring of configuration item (status accounting). Note, if the configuration items attempting to be migrated encounter errors, problem reports/trouble tickets will be issued. If they are rated as severe and/or critical to meet the functional requirements, the configuration items will not be migrated. Thus another exit criteria that must be met is no high, severe or critical problem reports can be outstanding. At the conclusion of the product testing phase, configuration management team creates a final baseline version of the release. All previous interim baselines can be deleted at this point.

For a more detailed diagram of the change request process flow, see Appendix D.

2.1.2.4 Install Configuration Items

The configuration management team ultimately has the responsibility of validating and certifying not only the software being developed (and maintained), but they are responsible for certifying the environment in which the software was developed and tested meets the standards established by the Department of Education and the enterprise architecture team. This process specifies how the configuration management team only installs or coordinates the installation, of configuration items verified by the development teams and/or the virtual data center. The change requests that initiate the inclusion of new/updated configuration items into the CM controlled environment are critical to this process. Without them and verification from those requesting them, no updates will be made to the CM environment. After validation, CM will install the configuration items, but before these items are actually included in a baseline configuration, CM initiates a testing and certification activity. If this activity is in support of production testing, it will be performed by the quality assurance organization(with independent verification & validation contractor support). Only after passing all the tests and being certified, will the configuration items become included in the baseline.

For a more detailed diagram of the change request process flow, see Appendix D.

2.1.2.5 Implement Configuration Items

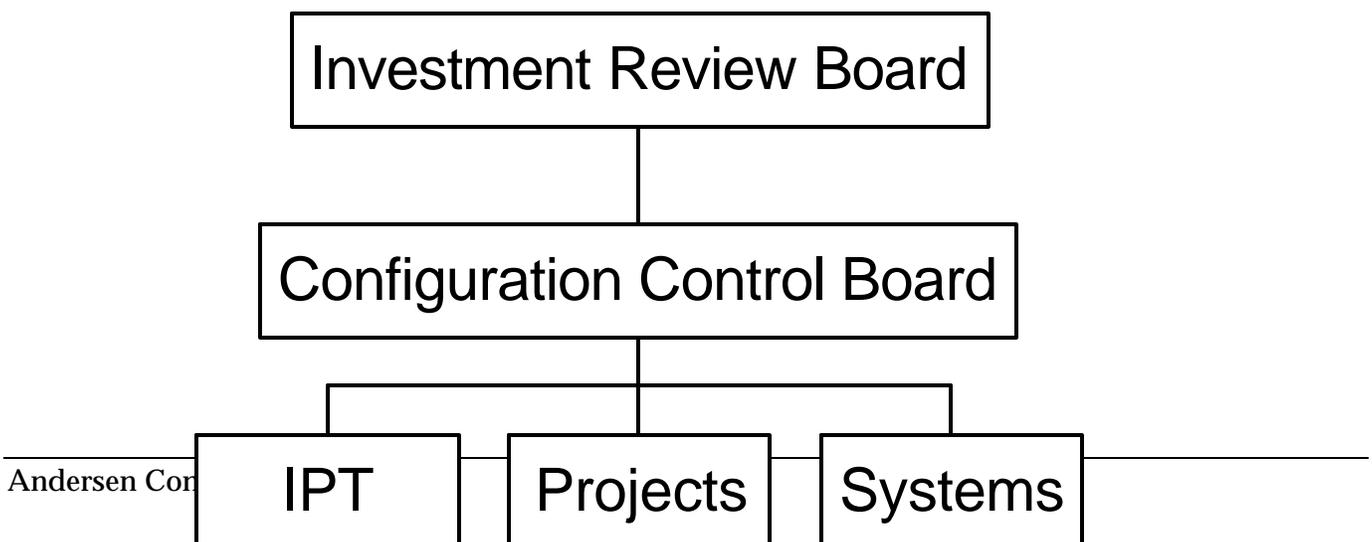
An implementation workplan and schedule are defined, then submitted to the enterprise architecture team for verification of requirements and standards adherence. A configuration control board review is convened to discuss the validity. Upon approval the virtual data center is tasked with performing the installation and testing. The configuration management team and quality management organization are called on to validate the implementation. They will utilize an independent verification and validating (IV&V) testing approach to validate the release. The efforts of the IV&V contractor will be coordinated with CM and the quality assurance (QA) organization. Together, independent testing will be performed by the IV&V contractor. The results will then be verified by the QA group.

For a more detailed diagram of the change request process flow, see Appendix D.

2.2 Organization and Responsibilities

A functional representation of the enterprise configuration management organization is depicted below. Note, the investment review board level is responsible for defining the capability release plan and IT portfolio. It has review authority over the IT release budget as well. The configuration control board level has oversight responsibility of the requirements of architecture (modernization blueprint, enterprise architecture, virtual data center, multiple information systems).

Enterprise Configuration Management Functional Representation



2.2.1 Configuration Management Team

The configuration management team will be responsible for maintaining the configuration management plan(s) and implementing the program and procedures. This function will also maintain the system databases that document change status, prepare and distribute configuration status accounting and tracking reports to appropriate project and client personnel. It will also coordinate configuration management activities between the development teams and the virtual data center. The configuration management functions are organized into three areas: architecture, documentation and software. The architecture area maintains configuration items related to the development, execution, operational and test architecture environments. The documentation area manages configuration items related to the business requirements, data/process flows, technical design documents and training materials. The software area manages application software, interfaces, the software development files, and the requirements traceability matrix.

Configuration management will provide guidance and assistance on all SFA CM issues. These responsibilities will include the following major activities:

- Controlling the product baseline(s) and all development, integration and testing configurations.
- Ensuring that change and report tracking is kept current for change request(s) and maintaining a history of such changes.
- Moving source code into a CM controlled area and performing system builds. Maintaining the system baselines. Producing sizing guidelines
- Preparing project master release tapes from the CM library and delivering them to the Development Team Leader for distribution to the client.
- Controlling all hardware baselines that support the SFA requirements.
- Capturing and controlling all project baselines.

The configuration management organization is comprised of a number of CM coordinators. The CM coordinator heads the individual project configuration management Team. It is made up of project managers and representatives from the different development teams. The CM coordinator determines the impact of a proposed change in the SFA environment. The coordinator reviews the change request at weekly status meetings and logs the assessment. His/her role is to ensure that changes are aligned with SFA strategic goals and business requirements. He/she does this by determining the change impacts in terms of tangible benefits and conformance to existing or enhanced architectural guidelines.

The activities required by the virtual data center will be the responsibility of the configuration management team. The activities the virtual data center will be conducting include the configuration, maintenance, upgrade and tracking of all infrastructure (hardware, system software,

communications, database, network, CASE Tools, utilities) requirements received from the SFA application development teams. The coordinating point for the requirements, tasking and implementation will be the Configuration Control Board. In particular, decisions that affect the enterprise, such as changing the logical data models and/or infrastructure enhancements, which may have significant impact, can be monitored and reviewed from a coordinated release viewpoint.

When the SFA application Development Team(s) complete their development lifecycle, the configuration items (updates and new) are migrated to the virtual data center where they go under Virtual Data Center change control for the production testing phase. During this testing phase, configuration item change/update will be coordinated with the Development Team via the Configuration Control Board process. In addition, the virtual data center will conduct performance and capacity planning tasks.

2.2.2 Enterprise Architecture Team (EAT)

The Enterprise Architecture Team, in the enterprise configuration management process, is responsible for advising all the previously mentioned teams in the following areas: impact assessment of prospective modifications to configuration items; enterprise engineering standards adherence as it pertains to SFA achieving its strategic technological initiatives; and providing input into the Decision Support Group and Investment Review Board process of identifying enterprise-level change requirements. In addition, this team will recommend changes/enhancements to the enterprise architecture standards as situations arise.

2.2.3 Development Team

The day to day implementation of configuration management policy and procedures are the responsibility of all SFA application development teams. The Development Team Lead and the respective project CM Coordinator are responsible for identifying and packaging each of the configuration items which make up a unique version and handing this off to the configuration management Team. This includes the initial version of a new release and any subsequent versions of that release (e.g. fixes and enhancements). The Development Team is also responsible for developing test plans, scripts and data. The Development Team Lead, with the help of the CM Coordinator, must ensure that the integrity of the solution is maintained once a change has been implemented in a particular development/integration baseline.

2.2.4 Configuration Control Board (CCB)

The Configuration Control Board (CCB) reviews the change requests against Investment Review Board (IRB) requirements and Enterprise Architecture standards and methodologies. The Configuration Control Board ensures that changes are prioritized, coordinated, controlled and integrated within the program quality framework and result in real, tangible benefits. Authorized

changes are included in a rollout plan. Deferred or rejected requests are reviewed. The assigned configuration management coordinator, upon analysis, and rework by the development team, can resubmit the change request.

This board is chaired by a Deputy CIO delegate and is comprised of the General Manager representatives, CIO representatives (Enterprise IT Management, Enterprise Services, Enterprise Applications), COO representatives, CFO representative and the Enterprise Architecture Team representatives(CM, Methods & Standards).

The Configuration Control Board should establish a standing weekly meeting. Attendance should be mandatory for all Development Team representatives actively engaged in Business Capability Release efforts being brought before the CCB. This will promote a coordinated and integrated approach to issue resolution. The issues/concerns brought before the board should be screened and filtered by the CM Coordinator/Development Team Lead, to those having significant impact (functional/technical) on the SFA strategic initiatives.

This group is also chartered to resolve issues involving strategic benefit, cost considerations, and quality management issues. They oversee significant enterprise architecture changes. It will be empowered to make business capability release decisions (approval/disapproval of change request) involving strategic benefit, excessive cost and quality. It will meet as necessary to resolve issues that are less technical in nature but more strategic.

2.2.5 Investment Review Board (IRB)

If significant funding issues exist and/or a business case analysis is required, the CCB will forward the proposed change requests (and analysis) to the IRB/Management Council for their review. If budget implications are above the threshold criteria, the investment review board enters the process at two decision points. The first is during the analysis of the change request. After planning the release, the process calls for the IRB to deliberate and provide a go/nogo decision for the development work to begin. The second place in the process the IRB may be involved (if it is above threshold criteria) is after installing/pilot testing the configuration management items. An IRB meeting will be convened and the go/nogo implementation decision will be made.

Note, the Decision Support Group(DSG)/Chief Financial Office(CFO), depending upon the financial implications of a proposed enterprise change and/or new initiative, will be involved in the decision making process.

2.3 Configuration Management Tools

This section describes the criteria to be used in selecting the enterprise configuration management tools suite to be used throughout the system development life cycle.

2.3.1 Tool Selection Criteria

In order to ensure that resources are efficiently used and distributed environments are effectively managed, configuration management tools need to be identified and appropriately selected. To this end, there are a number of essential areas that must be considered carefully in comparing and choosing configuration management tools. To manage software, there are three considerations to keep in mind: software version control, software distribution, and change management.

2.3.1.1 Software Version Control

The version control and migration control tools should collectively provide the following features:

- Locking to prevent concurrent or unauthorized updates.
- A grouping mechanism for system components (the group defines the configuration and may include repository and non-repository objects).
- Tools need to support sharing of system objects across major subsystems.
- A promotion mechanism, which moves a configuration from one environment to another.
- The option of automatically locking at the conclusion of successful promotion.
- Mechanisms for reverting to an earlier release.
- Audit trails per user and per component (change log).
- The ability to annotate the change log with free text.
- Flexible reporting capabilities, including the ability to list all the changes (deltas) of a component.

The multiple vendor multi-platform nature of modern network architectures will require increased effort when it comes to managing and controlling the many components that exist across a given enterprise. Here are some key goals to look for during your tool evaluation:

Ensuring the integrity of the versions of each software component.

A predictable level of working functionality should exist for each version of each component of your software. Maintenance should be applied in a way that ensures specified versions of a software component do not perform differently, once maintenance has concluded.

Avoiding conflicts and dependencies among software components.

Changes in one version of a software component should not adversely affect other software components. Also, such changes should be thoroughly tested to ensure that all software that utilizes that component continues to function properly.

Ensuring available vendor support.

Coordinating and maintaining versions such that vendor support is consistently available. This includes avoidance of situations where vendors are unfamiliar with the effects of one version of software within a particular environment ("we've never run it on that platform before"). Current vendor directions and policies need to be monitored to ensure that currently used versions will be supported or will be upgradeable.

Multiple-Version Support

Due to the manner in which software is distributed or changed, it may become necessary to support multiple versions of software on a broad scale. Planning for such events should occur before the first maintenance cycle.

Version Tracking

Tools may be available to verify what versions have been installed or upgraded, and where these versions exist. Without such monitoring, it is possible that version changes and upgrades may produce unpredictable negative results that are difficult to trace.

Multiple Platforms

Some software may exist on more than one platform. Version tracking should provide for a means of consistent control across all supported platforms. This can be a complex task, since many vendors use the same version number for software that runs on multiple platforms, but provides features on one platform that do not exist on another.

Version Backup

For recovery reasons, it should be possible to backup to previous versions after upgrades have occurred. It may be possible (and highly desirable) perform such backups without re-booting an application and without significant downtime.

2.3.1.2 Software Distribution

In large distributed environments, installing software using traditional "sneaker-net" methods has become too cumbersome and slow to maintain standard configurations throughout the company. A means of remotely installing software and upgrades becomes a major factor in software configuration management. These types of tools can be utilized to automate the process of software distribution and system maintenance. Some key characteristics of software distribution packages to look for are:

Ability to distribute software to the workstations on the network

As distributed environments become more heterogeneous, it is important that management of the environment does not become equally complex. There needs to be a means to communicate with the current client platforms over existing network protocols to maintain a single interface to distribute software. With this facility SFA can avoid additional costs incurred by the need for new hardware.

Maintaining consistency among workstation configuration

Keeping all workstations current with the same versions of software, as well as keeping standard platform configurations for users to work on minimizes training costs and support costs. Software distribution products are beginning to integrate asset management and inventory into their products to make this maintenance accessible and centralized.

Minimizing workstation downtime

This implies that software may be distributed during off-peak hours or installed when it is convenient for the user. Most if not all products are now providing both “push” and “pull” capabilities so that scheduled distributions of the software as well as client initiated downloading are possible.

Minimizing network bandwidth in the process of distributing software

As network usage grows within the LAN and WAN environments, distribution must not slow network traffic to a standstill. Centralized control of multiple file servers along with compression/decompression features become important.

Assess the client’s configuration to ensure that installation and execution will be successful

Asset management becomes more important as the size of the network grows. Along with asset inventory, running checks for prerequisite software, checking for sufficient disk space, and verification of any co-requisite software or hardware has become another important feature of software distribution products.

Provide transaction logging and Error tracking/recover.

Maintaining records, verifying successful completion and/or unsuccessful distribution contribute to the overall management of distributed environments. Reporting results to the administrators and the users via logs or other notification methods such as e-mail will reduce customer support needs and verification time.

2.3.1.3 Change Management

Every aspect in an information technology environment changes. Changes are constantly being made in computer networks, in computer software, and in system administration. These changes are continually increasing in size and complexity and have a profound effect on a company’s performance. For these reasons there is a clear need to manage these changes quickly and efficiently.

Change Control Management is a process that facilitates planning, tracking, and implementing changes as they are introduced into an distributed environment. These tools can be utilized to document, control and facilitate the change control process. Some key features to look for in Change Control Management type tools are:

Tracking changes

A change tracking system should provide a method for change requests, as well as allow for updating these records as the project develops.

Workflow Management

The change must first be verified or approved. Complex changes may need to be broken up in smaller more manageable tasks. These smaller tasks needs to be prioritized, synchronized, and assigned to personnel.

Facilitate communication

Change Control Management should allow multiple individuals to communicate with each other while working on smaller tasks. One of the most common forms of communication is via e-mail. In order to communicate progress and status of a change request, querying and reporting should be available.

3 Next Steps and Implementation Plan

3.1 Next Steps

3.1.1 Identify the Implementers of Enterprise Configuration Management

Upon the decision to move forward, the first step required to fulfill the need of an enterprise focused configuration management organization is to build the support organization that is going to drive it. However, to start, someone or some team must be defined to lead the charge. They should be tasked with the responsibility for obtaining and identifying the resources and implementing the organizational structure for the SFA. Tasked with this responsibility, one of the first decisions that needs to be made is whether the SFA would be better suited if the configuration management Team were composed of SFA staffers or whether it should be outsourced.

Our recommendation is the Modernization Partner Enterprise Architecture Team (Delivery QA unit), be tasked to work with the CIO Enterprise IT Management organization to implement the necessary structures, policies, procedures and guidelines. This is recommended because the potential time lag in identifying and mobilizing the appropriately skilled SFA personnel, could be offset by the readily available resources of Modernization Partner. Utilizing Modernization Partner personnel at least for the first phase of implementation would afford the SFA ample time to conduct a personnel search (and training). It should be noted, SFA would manage the CM effort if the Modernization Partner or any other contractor is selected to work with the CIO IT Management organization. Upon the conclusion of the first phase and once the organization has become operational, at the SFA discretion, the configuration management team could revert to a SFA operation.

3.1.2 Define Scope of Implementation and Timeline

The second step is to define the scope of the implementation and determine how much time should be allocated to implement it. This endeavor can not be accomplished over night. The scope in this case could mean multiple things. For instance, the size of the configuration management team must be determined beforehand.

A phased implementation approach will be required to implement something of this magnitude. A detailed project plan must be developed containing well defined tasks/objectives with measurable milestones. One or two individuals working with the CIO Enterprise IT Management team could produce the detailed project plan in the next period.

3.2 Implementation Plan

3.2.1 Assemble Organization for Configuration Management

The enterprise configuration management segment of the enterprise architecture team should be established. This team will implement the configuration management approach by integrating the configuration management coordinators into the application development teams. They would then become responsible for providing CM with updates on activities (at the project level) that have enterprise level impacts. The first task of the enterprise CM team is to create an enterprise configuration control board. The enterprise configuration control board members include: representatives of the Office of the CIO, COO representatives, General Managers representatives and Enterprise Architecture Team representatives.

The current Software Configuration Control Board which meets at the application development level would be restructured to operate as the new enterprise level configuration control board. Their current membership and representation on the existing software board provides the beginning mixture of the competencies required for the enterprise configuration control board to be effective.

The Deputy CIO Delegate facilitates the board meetings. Channel leads, users, project managers, and technical leads are the normal attendees for the board meetings. However, the CM coordinator should vary the attendance list from meeting to meeting to ensure that just the necessary people to provide information or actually make the decision are there.

3.2.2 Establish an Environment on Pilot Platform

It is important to establish an environment where the enterprise CM model will be tested. The CIO, along with SFA general manager(s) help will identify candidate system(s) to pilot the CM model on. To limit the risk initially, there should only be one system involved. A tool that meets the criteria for SFA should be selected. The platform specified to pilot the CM model should be configured and installed with this tool suite. It is important to review the proposed CM model before doing any pilot. The CM technical specifications document should be created.

3.2.3 Define Configuration Items

The Configuration Management process starts by the identification of the requirements for the items that are being developed. A change control process is required to ensure that the complete impact of any change is known and reported. Change control involves maintaining documentation of changes to configuration item through the system life cycle. Approved requirements, specification, and other project documentation must be tracked to assure accuracy. Action items resulting from discrepancies identified at configuration reviews must be tracked to ensure successful closeout. The implementation

of approved changes must be tracked to ensure that problems that affect the operation and support of organizations using the new configuration do not arise.

During the life of the project, data will be gathered and maintained to ensure that the contents and components of the individual configuration items and their current location (including status in the development cycle) is know. The configuration management team will maintain a continuous and accurate status of individual configuration items that comprise the project.

See Appendix C for more detail on defining and categorizing the configuration items.

3.2.4 Develop Proof of Concept

A “proof of concept” for the defined CM environment should be developed. If the pilot is successful, this solution would be a candidate for the CM enterprise model that could be deployed to other SFA systems. Representatives from all aspects of the development lifecycle must have input in this phase; development, test, CM support, and deployment.

3.2.5 Certify Environment

The objective of this step is to construct and certify technical environments in support of development and test activities. Here, components are created, linked and certified to ensure the technical environment is established. In addition, service level agreements (SLAs) are established and support procedures are defined.

3.2.6 Support Environment

It ensures that the technical environment functions as requested throughout the development or testing effort. In this phase, issues are addressed, change requests are processed, software code rollups are facilitated, software fixes are migrated and data is backed up, restored and refreshed. Specifically, the configuration management team performs hardware, architecture and software support, initial investigation of issues/discrepancies, identification of change requests and resolution of discrepancies.

3.2.7 Train Personnel in CM Processes and Tool

Configuration management training has several aspect. The first is the recognition that to operate effectively people need training. Untrained people in a configuration management organization can be catastrophic to SFA system development activities.

Train Personnel in overall process

The first step is to ensure that everybody in the SFA organization that is affected by CM understands the overall process. A configuration management training that teaches the basics of configuration management should be established. Every person involved in the configuration management process should attend this training

Organizational Training

In an environment with multiple systems like SFA, organizational training is crucial. Individuals from different project teams need to work together to accomplish desired ends. The organization as a whole, when first established, will be inefficient, and an allowance should be made for this factor. One successful approach to organizational training is to establish the configuration management organization with a nucleus of people before putting the new CM process in place. This allows the team to organize the teams, establish the detailed written procedures by which each individual team will operate, test the procedures out, and modify them to the point at which a person a workable solution.

Specific Tool Training

Vendor tool training should take place during the initial month of the establishment of the configuration management task. Only those using the different CM tools should participate in this vendor training.

3.2.8 Measuring CM

The configuration management electronic media libraries are rich sources of data and can easily be mined to obtain significant insight, not only into the configuration management process but into the overall project operation itself.

Configuration Management Metrics

Metrics should be used to provide indicators of where improvements can be made. In terms of configuration management, process metrics are directly available. Metrics might include such items such as: 1) the number of changes processed by the configuration management team and the status of these changes plotted against time, and 2) the number of problem reports and the status of these problem reports against time

Project Metrics

One immediate source of raw data for project metrics is the configuration management library. The configuration management library holds the source files (documents and source code). By examining the documentation source files, the following can be obtained:

- Documentation metrics – These sizes can be easily measured in words using any standard word-counting program. Metrics can be provided for: (1) individual documents, (2) individual types of documents and, (3) documents grouped by configuration item (the total of all the sizes of the individual documents produced for a specific configuration item).
- Source Code metrics – The number of source files of code, which could be further grouped by source file size. This could be further grouped by computer program and by project. The number of revisions can be tracked where a revision is defined as the issuance of a new version of a source code file after the issuance of the initial version.

The different metrics can be established assuming that (1) the organization has some common set of standards for source of code, (2) a standards-checking tool is used when a source file is entered into the library, and (3) as part of that standards-checking effort, the number of source lines of code in a source file , is automatically provided.

4 Appendix A: Candidate Forms

The following template will be used to determine pertinent information for a specific form or document that will guide users in handling data.

 <i>Department of Education</i>									
<FORM NAME>									
1 SECTION I: IDENTIFICATION									
CONTACT INFORMATION:									
1) DATE CREATED:	2) DATA LOCATION:	3) PROJECT:							
4) CREATED BY:	5) E-MAIL:	6) PHONE #:							
7) RECEIVED BY: 1) 2)	8) E-MAIL: 1) 2)	9) PHONE #: 1) 2)							
2 SECTION II: PURPOSE									
10. DESCRIPTION:									
<table border="1"><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr><tr><td> </td></tr></table>									
3 SECTION III: USAGE									
11. INSTRUCTIONS:									
1<Detail usage instructions here>									
2									
3									
4									
5									
6									
4 SECTION IV: REVIEW/APPROVAL.									
12.									

Change Control Request

Forward to the CIO IT Management Configuration Management Lead

Date ___/___/___

4.1.1.1.1 General Information

Change Control Requester: _____ Phone: _____

Change Priority: _____

Business Area(s) Affected: _____

Change Control Information

Change Title: _____

Change Control Number: _____

Component Type: _____

Area(s) of
Responsibility _____

Reason: _____

Issues: _____

Expected Results: _____

When

Scheduled Implementation Date (mm/dd/yyyy): ___/___/___

Impact Assessment Template

Project Information

Project Name:
Project Number:

Change Category

Type of change:

- Business Application Change
- Database Change
- Desktop Change
- Network Change
- Network Operating System
- Other
- Server Change
- Support Application Change
- Workstation Change

If Other, provide a categorization of the change:

Business Impact

Users:
Departments:
Business Function Affected?
Impact to Users
Level of risk to the users?

Yes No

Backout/Recovery Requirements

Recovery Type:
Backout Requirements:

Simple Moderate

Distributed Environment Impact

of Desktops Affected:
of Servers Affected:
Type of Network Changes:

None Minor

Changes to I/T Standards Required:

Yes No

If Yes, list the Standard and the changes required:

Mainframe Environment Impact

of Applications Affected:

Interfaces Affected: Yes No

If Yes, list the Interfaces:

Response Time Change: None Minor

Changes to I/T Standards Required: Yes No

If Yes, list the Standard and the changes required:

Training Requirements

Training Type: None Informal

of days for training:

Component Inventory

<i>Changed Component</i>	<i>Who's Affected</i>	<i>Impact</i>	<i>Estimated Downtime</i>

Overall Impact Assessment

Prepared By:

Date:

5 Appendix B: Candidate Checklist

Change Control Readiness Checklist

The following checklist is to be used when the change request is ready to be deployed to the production environment.

Checklist Description
1. Backout/rollback instructions have been provided to deployment management
2. The appropriate impact analysis's are completed and have been given to deployment management. It will be presented at the Weekly Change Meeting. An impact analysis may include: <ul style="list-style-type: none"> • end users • workstations • file servers • application servers • database servers • dependencies to other applications, databases and systems • server levels
3. A service level agreement (SLA) is complete and operations is ready to meet SLAs
4. End users and support personnel have been trained on the new change
5. Documentation or documentation changes have been included
6. Ensure that the application has been successfully installed and certified in the certification test lab and a package has been created and tested.
7. Confirm that the release packages are ready for deployment as scheduled
8. Ensure that an operations guide is complete for new applications and, if necessary, for application updates. Ensure that transition meetings have been held to transition the application/system to the appropriate support personnel.

Step	Complete
Change Requester	
1. Work with the appropriate IT support teams to develop a deployment plan and complete the impact analysis(s), if necessary.	
2. Submit change control form as soon as the requestor is aware of a production change.	
3. Ensure that a SLA is complete, if necessary	
4. Ensure backout/rollback instructions are complete.	
5. Ensure that an operations guide is complete and the appropriate transition meetings are complete, if necessary.	
6. Attend weekly change review meeting, when change is presented.	

7. Coordinate training with the SFA training coordinator, if necessary.	
8. Ensure appropriate IT teams, end users, application support staff are trained, prior to rollout.	
9. If change requester is the person making the change, continuously communicate with customer service, end users, SFA stakeholders, etc. as the change is implemented.	
Configuration Management Team	
1. Track problems and incidents which require a change to the production environment in order for the incident to be closed.	
2. Close problem ticket once change has been implemented.	
Change Control Coordinator	
1. Review and log requests	
2. Approve low impact requests	
3. Ensure the change control checklist is complete	
4. Ensure change approval has been obtained	
5. Coordinate a meeting between the requester and appropriate IT team, if necessary.	
6. Distribute weekly reports and coordinate the Weekly Change Review Meeting.	
7. Log weekly meeting outcomes	
8. Notify requester of scheduled rollout date(s)	
9. If necessary, work with the requester and other rollout management teams to determine and adjust the rollout schedule.	
10. Update change request status and include any additional notes	
11. Monitor and communicate the progress of the change request	
Weekly Change Meeting Participants	
1. Identify and resolve conflicts	
2. Prioritize requests and identify proposed schedule changes	
3. Approve, reject, defer or assign as pending each change request	
4. Send an alternate if the primary member can not attend the weekly meeting	

Change Control Risk/Impact Examples

Tasks	Risk/Impact		
	High	Medium	Low
Install a new server			X
Server replacement	X		
Server move		X	
Add/replace/remove server hardware	X		
Upgrade server system software (NT, UNIX, Mainframe, specify other)	X		
Reboot server hardware		X	
Modify/update "standard" applications on server (MS Office Suite, MS Mail, etc.)		X	
Modify/update "business" applications on server		X	
Remove "standard" applications on server			X
Remove "business" applications on server			X
Upgrade Dell and COMPAQ system software	X		
Upgrade Dell and COMPAQ workstation software			X
Reboot Dell and COMPAQ hardware		X	
Modify Dell and COMPAQ hardware		X	
Install router card		X	
Upgrade router software	X		
Install hub card			X
Upgrade hub firmware		X	
Change LAN printer queue name		X	
Modify system login script		X	
Call Center system	X		

6 Appendix C: Configuration Items List

C.1. Identification

Configuration Identification is the process of selecting, identifying, and naming configuration items (CI). A configuration item is an aggregate of hardware and/or software, developed and managed as a single item. Some configuration items are initiated and operated independently of other configuration items, others are dependent and perform functions that satisfy end-user requirements. A configuration item contains information that is created as part of the software engineering process. It also has specific functional and/or physical characteristics that have been defined in terms of a specific standard. The aggregation of such elements must be realizable in a product that can be observed and evaluated in terms of degree of satisfaction of the user approved requirements for form and function. In designating configuration items, the CM will establish a numbering scheme to correlate the configuration items and associated documentation. Configuration identification is performed by the appropriate functional area manager in coordination with the CM and is maintained during all phases of the project. Configuration identification provides the basis for applying management control over the system configuration. It allows for isolated items to be controlled, their status to be tracked, and their configuration to be reported.

Configuration identification is controlled through three evolutionary baselines. The three baselines are described below:

- The Functional Baseline will be established by government-approved standards. The Functional Baseline will be established at the end of the conceptual phase as marked by the system requirements review (SRR).
- The Allocated Baseline is the initial approved allocated configuration identification established by the authenticated development specification. The establishment of the allocated baseline for a CI normally signals the end of the requirements phase or the beginning of the preliminary design phase as marked by the SRR.
- The Production Baseline is established by the authenticated product specification or its equivalent. The Product Baseline is normally established by the customer at the end of the full scale development phase, which is marked by the completion of the functional and physical configuration audits. The Product Baseline defines the as-build product, consisting of the allocated baseline and all approved changes. This baseline can only be changed by a Change Request approved by the Program Management, the CCB and the client.

Configuration Item Hierarchy

The following template is used to define the SFA solution. It provides a hierarchic classification for configuration items. The “leaf” nodes of the hierarchy (shown in italics) identify the actual configuration items which must be defined in the description of a version of the SFA solution. It is critical the lowest significant configuration item level is defined for each type of configuration item. Note that it is the responsibility of the CIO IT Management Configuration Management Lead to define the configuration for each version of the SFA solution.

CLASS

CONFIGURATION ITEM

REQUIREMENTS

- [*Requirement Statement*]*n

SOFTWARE

APPLICATION SOFTWARE

COTS

- [*COTS Package*]*n

Modified COTS

- [*COTS Package*]*n
- [*Software Module*]* n

Custom

- [*Software Module*]*n

INTERFACE SOFTWARE

COTS

- [*COTS Package*]*n

Modified COTS

- [*COTS Package*]*n
- [*Software Module*]n

Custom

- [*Software Module*]*n

TECHNICAL ARCHITECTURE (DEVELOPMENT, EXECUTION, OPERATIONS)

COTS

- [*COTS Package*]*n

Modified COTS

- [*COTS Package*]*n
- [*Software Module*]*n

Custom

- [*Software Module*]*n

DATA CONVERSION SOFTWARE

COTS

- [*COTS Package*]*n

Modified COTS

- [*COTS Package*]*n
- [*Software Module*]*n

Custom

- [*Software Module*]*n

DATABASE

- [*Database Module*]*n

HARDWARE

- [*Hardware Component Specification*]*n

DESIGN SPECIFICATIONS

- [*Software Module Design Specification*]*n
- [*Database Design Specification*]*n

TEST SPECIFICATIONS

- [*Test Condition*]*n
- [*Test Script*]*n

TRAINING

- [*Training Module*]*n

USER PROCEDURES

- [*User Procedure*]*n

OPERATIONS PROCEDURES

- [*Operations Procedure*]*n

INSTALLATION PROCEDURES

- [*Installation Procedure*]*n

DATA CONVERSION PROCEDURES

- [*Data Conversion Procedure*]*n

Configuration Item Descriptions

Configuration Item	Description	Identifier
Requirement Statement	Each version of the SFA solution is intended to meet a set of requirements.	TBD
COTS Package	COTS packages are solution components pre-assembled by the package vendor. The vendor is responsible for the configuration management of the parts which make up the package. SFA need only manage the package as a single configuration item.	TBD
Software Module	This is the lowest configuration-managed building block of software. Configuration management at the module level is required for custom and modified COTS software. Modules are classified by type: Window, Batch, Report, Interface, Tech Arch, Data Conversion.	TBD
Database Module	This is an instance of database software which defines the structure of the SFA database e.g. a DDL module	TBD
Hardware Component Specification	This is a specification of the characteristics of the hardware required to support SFA	TBD
Software Module Design Specification	This is a specification of the design of a software module. It includes functional and detailed design. Software modules are either windows, reports, batch processes, interfaces, technical architecture or data conversion.	TBD
Database Design Specification	This is a specification of the data base design. It defines data entities, elements and relationships.	TBD
Product Test Condition	This is a single, atomic statement of requirement which can be tested.	TBD

Product Test Script (including test data)	This is a series of steps through which the system is exercised in order to determine if it meets test conditions and thus the requirements. Test scripts define input data, sequential steps and expected results.	TBD
Training Module	This is a unit of training e.g. “Creating A Work Order”	TBD
User Procedure	This is a set of instructions designed to assist users with operation of the system	TBD
Operations Procedure	This is a set of instructions designed to assist systems support personnel with operation of the system e.g. data backup and recovery procedures	TBD
Installation Procedure	This is a set of instructions designed to assist fielding personnel with the installation of SFA at a site e.g. infrastructure site survey procedure.	TBD
Data Conversion Procedure	This is a set of instructions designed to assist fielding personnel and users with the clean up of site data and the execution of the final data conversion.	TBD

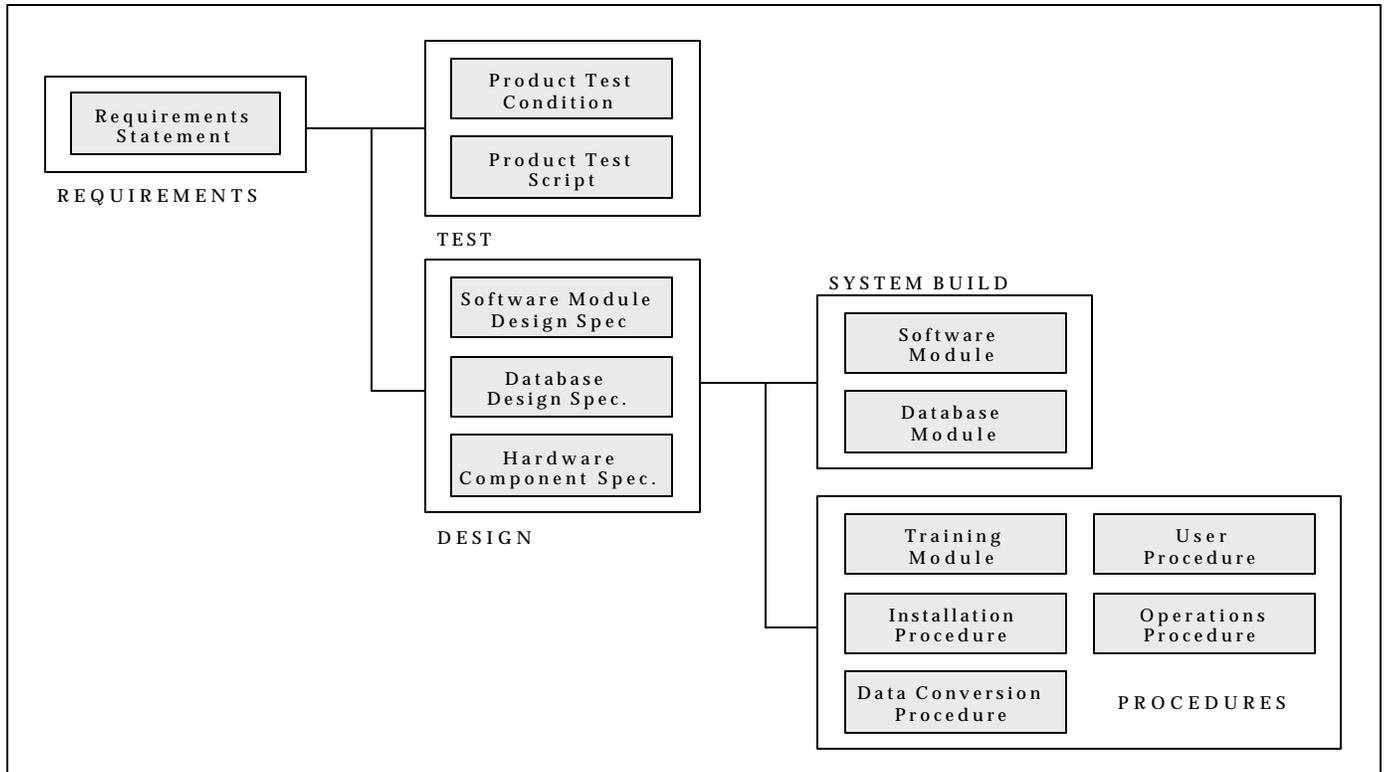
Note that approach papers and plans created during Solution Development (e.g., Increment 1 Development Plan, Beta Test Plan etc.) are not considered configuration items. These documents are intended to support informal communication between a small group of developers and do not require configuration management attention (i.e., no approval, change control, or version control)

Configuration Item Relationships

Configuration items are related to each other. The SFA Modernization Blueprint approach systematically decomposes the system requirement - through the design - to the system code and then tests the solution through a series of stages which ultimately demonstrate that the solution meets the original requirements. In parallel with this activity, user training and procedures, operational procedures and data conversion procedures are developed based on the solution design.

In this development approach, there is a hierarchic (or “waterfall”) relationship between configuration item types. As the development proceeds, “downstream” configuration items are developed based on the content of “upstream” items, e.g., software designs are based on requirements statements. This relationship is recorded in the description of each instance of a configuration item and provides the vehicle for traceability back to requirements and impact analysis when a configuration item is changed.

The relationship between configuration items is shown below.



C.2. Select Configuration Items

The purpose of selecting a configuration item is to manage their development and subsequently to manage their change. Each configuration item is a delivered item to which specific functional, performance, and physical characteristics are allocated. An effective configuration management program requires defining configuration items and placing them under configuration control at the appropriate time. To control and manage configuration items, each must be separately named and then organized using an object oriented approach. Two types of objects can be identified: basic and composite objects. A basic object is a “unit of text” that has been created by a software engineering during analysis, design, code, or test. Basic objects are items such as requirements specification, a source listing for a module, or a suite of test cases. A composite object is a collection of basic objects and other composite objects. Each object has a set of distinct features that identify it uniquely: a name, a description, a list of “resources,” and a realization. The identification task must achieve the following objectives:

- Item Name: A name that identifies the object unambiguously.
- Comprehensive Description: A description that is a list of data times that identify the configuration item type (e.g., document, program, data) that is represented by the object.

The descriptor provides a generic description of the function or purpose of the object and its role in a future release. Descriptors must be carefully selected and approved by the CM.

- **Resource Identification:** Resources are “entities” that are provided, processed, referenced, or otherwise required by the object. Data types, specific functions, and variable names are considered object resources.

The configuration items selected will form the project baselines.

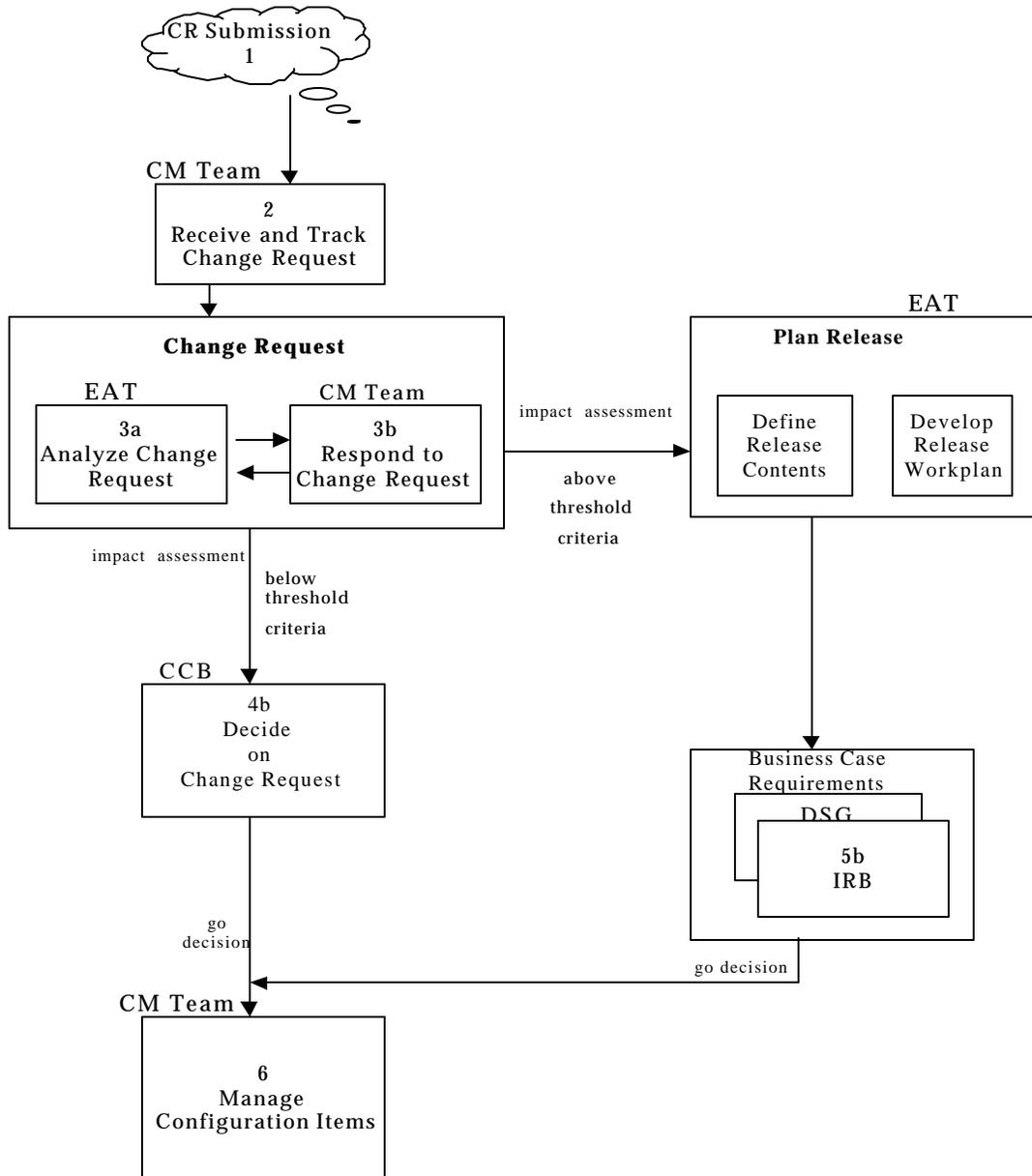
Independent Configuration Items

There are a number of release-independent documents which are subject to configuration management. Here is a sample of the typical deliverables defined in the Statement of Work, including:

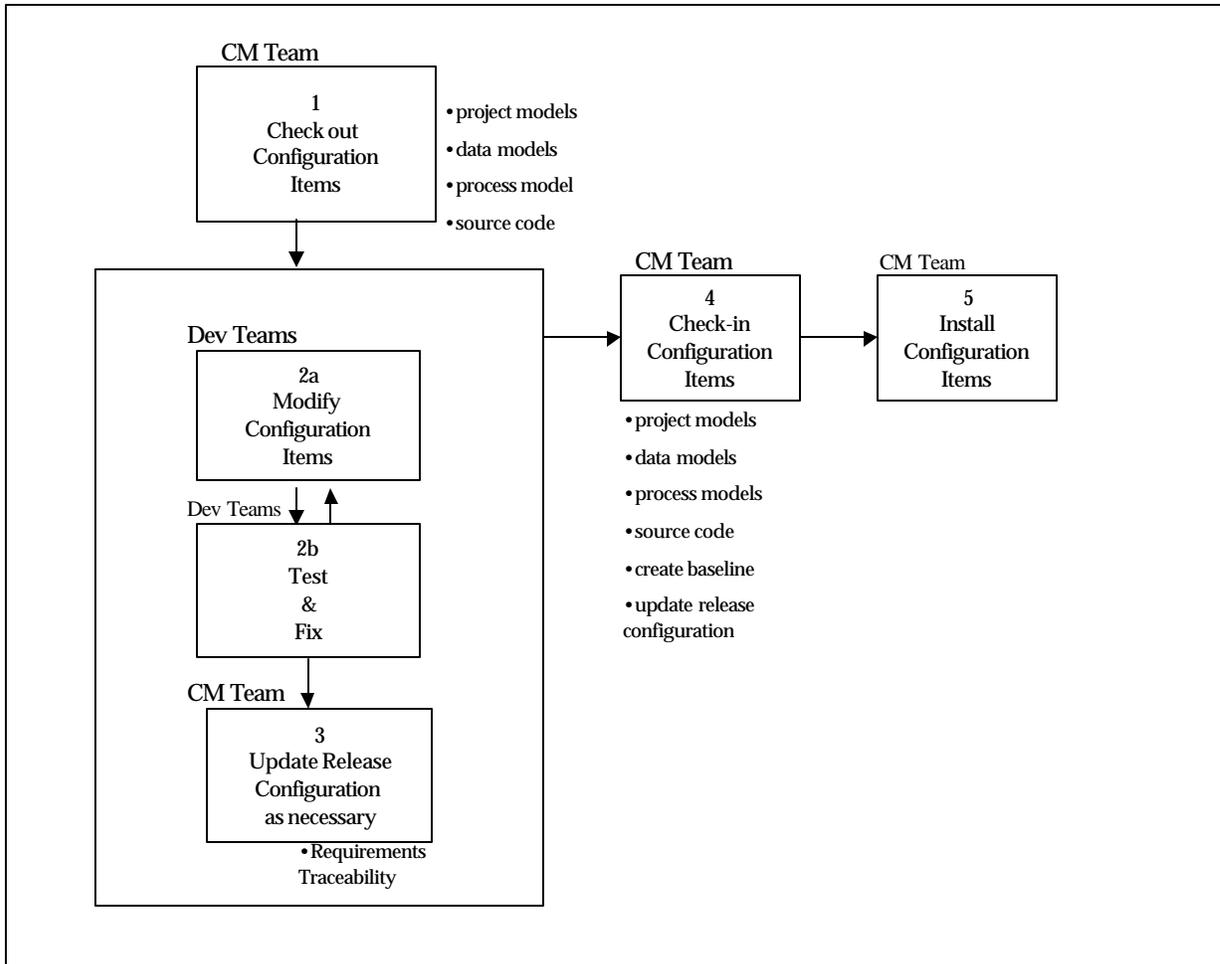
- Software Release Plan
- Legacy System Migration Plan
- Configuration Management Plan
- Quality Plan

7 Appendix D: Detailed CM Process Flows

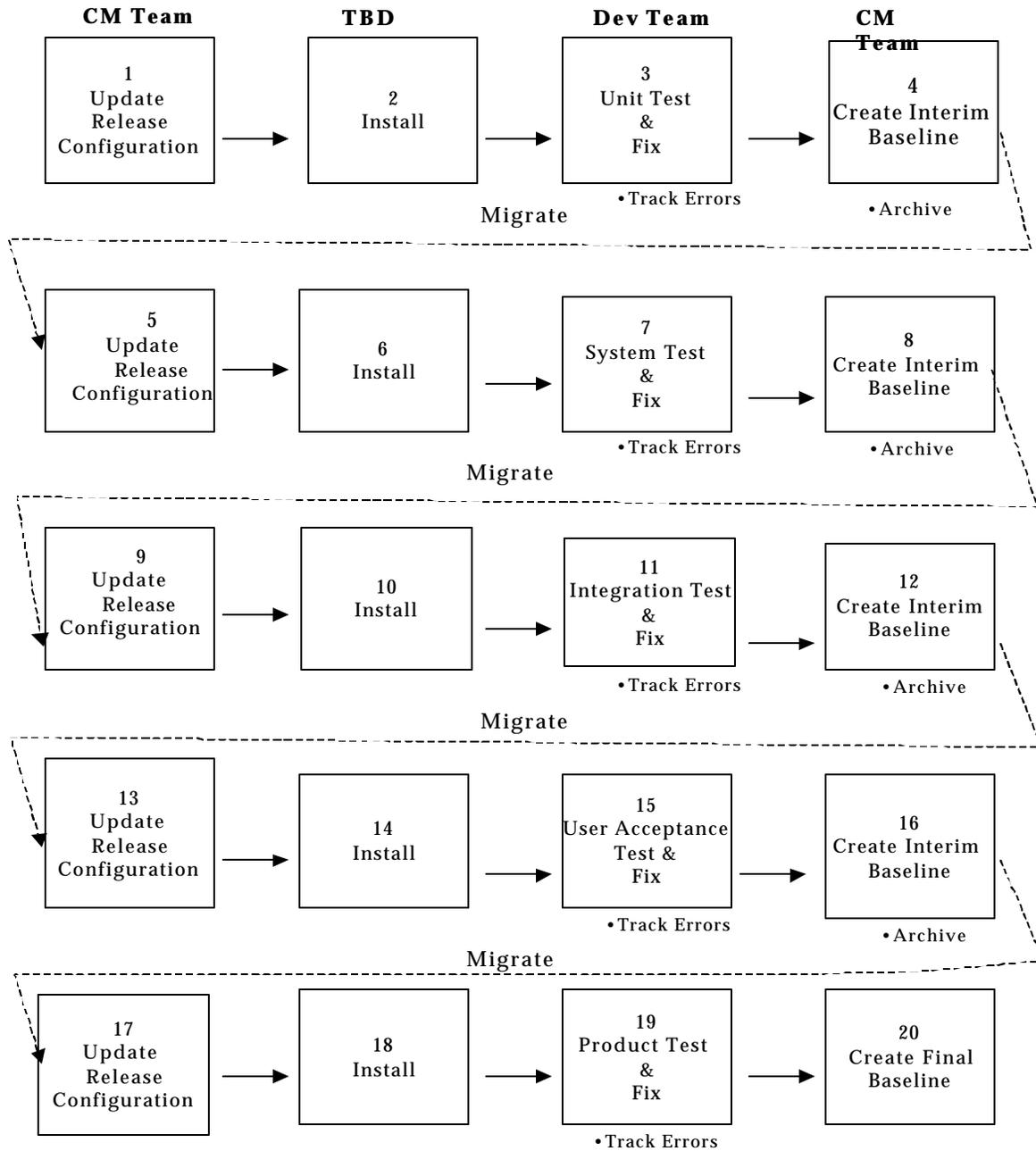
Manage Change Process



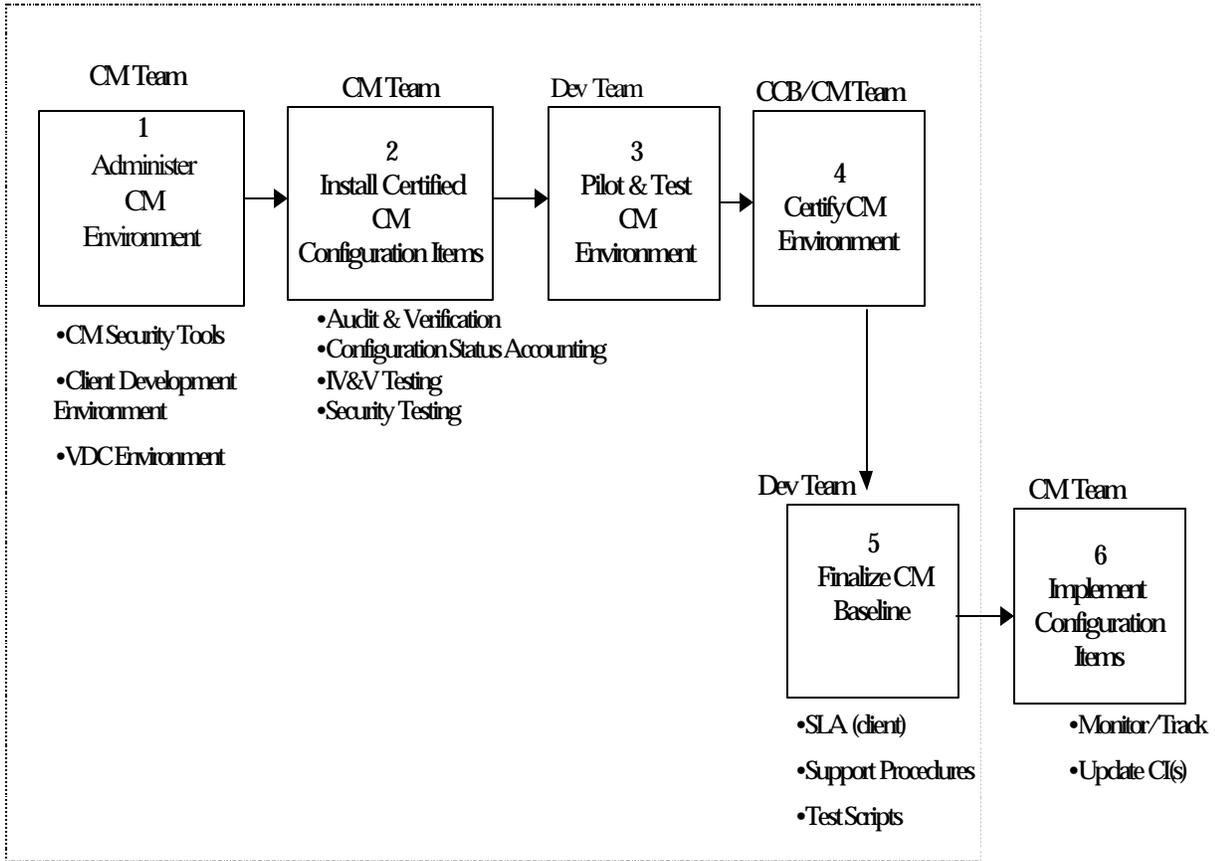
Modify Configuration Items Process



Manage Configuration Item Process



Install CM Items



Implement Configuration Item Process

