

Change Control & Configuration Management: A Blueprint for Mesusa Corporation

Paul Romer

CSD370

Module 12.2 Assignment

March 9, 2025

Outline

- I. Introduction
 - a. What is change control? (High Level)
 - b. What is configuration management? (High Level)
 - c. Why should Mesusa implement change control and configuration management?
- II. Change Control
 - a. Detailed definition
 - b. Process overview
 - c. Benefits and drawbacks
 - d. Vendor options
- III. Configuration Management
 - a. Detailed definition
 - b. Process overview
 - c. Benefits and drawbacks
 - d. Vendor options
- IV. Relationship between change control and configuration management
 - a. Change control vs Configuration Management
- V. Implementation Recommendation
 - a. Organizational requirements
 - b. Personnel and team considerations
 - c. Tools and automation
- VI. Change control and configuration management in the SDLC
 - a. Plan
 - b. Design
 - c. Implementation
 - d. Testing
 - e. Deployment
 - f. Maintenance
- VII. Conclusion

Change Control & Configuration Management

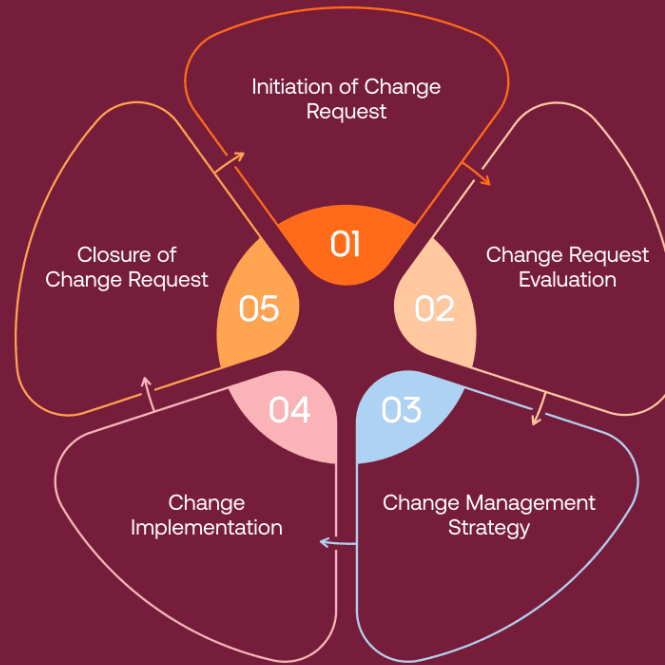
Introduction

Change control and configuration management are complementary processes that help an organization manage changes to their IT systems in a controlled and predictable way. Change control is a system, or set of processes and tools, to manage and reduce risks from software changes. It's a systematic approach that includes change proposal, change evaluation, and implementation. The goal of change control is to ensure that changes are necessary, well-understood, and do not introduce risk (WhatFix). Configuration management focuses on ensuring the consistency of a system's attributes and components throughout its life (Atlassian). It involves tracking all configurable items - like hardware, software, and settings - to ensure that they are in the correct state (RedHat). Implementing change control and configuration management processes are crucial for Mesusa Corporation as it scales its operations. Without these systems and processes, MeCo may struggle with untracked modifications, configuration drift, errors, and difficulty recovering from technical incidents. This paper will provide an overview of change control and configuration management fundamentals and how MeCo can implement these practices, including expected benefits, potential challenges, and available vendor tool options.

Change Control

Change control is a structured process for managing how modifications are introduced into a production environment. It ensures that every proposed change (like changes to software code, infrastructure, or procedure) is identified, evaluated, approved, implemented, and documented in a controlled manner (WhatFix). Because businesses rely on production systems to run day-to-day operations and to make money, changes should be made with care as to not disrupt operations. This is the purpose of change control, to prevent unnecessary or harmful changes, document all modifications, minimize service disruptions, and use resources efficiently (TechTarget). A change control process is a sequence of steps that each change should go through from the proposed change to its final implementation. While the exact change control process may vary depending on business needs, an example from WhatFix includes 5 steps.

5 Stages of the Change Control Process



1. **Initiation of Change Request:** Change requests can come from anyone (like users, stakeholders, project managers, and staff). The change request should include a description of a proposed modification. This includes details like what should be changed, the reason for change, who requested the change, affected systems or products, and an indication of priority. This request is entered into a tracking system like Jira.
2. **Change Request Evaluation:** A project leader or change coordinator reviews the request and may get additional clarification if needed. Urgent changes can be prioritized and fast tracked through this process, while bigger changes may require deeper evaluation. This is the part of the process where an analysis is done on the effect of the change to understand how it will impact current processes, affect project timelines, budgets, resources, and where risks like technical challenges and potential disruptions are understood. Key stakeholders are brought in to understand the change and share their expertise.

3. **Change Management Strategy:** This is where a strategy to effectively execute the change is developed. You should have a clear outline for what the change is and who is impacted. You should have a plan for the resources needed (like personnel, tools, and funding). Deadlines and milestones for implementing change should be decided on, and a plan for potential issues or setbacks should be finalized.
4. **Change Implementation:** In this step, the planned change is implemented. As the change is carried out, progress is tracked to ensure progress as planned. Key stakeholders are continuously informed of progress and setbacks, and the changes are quality checked to ensure standards are met. When the work to complete the change request is completed in a development environment, it should be approved by appropriate change board, stakeholders, or project lead before being pushed to production to ensure that the change meets expectations.
5. **Closure of Change Request:** Sometimes called 'Landing the Plane', after each change request is completed, it should be evaluated and documented. The initial objectives should be compared to the actual outcomes. Feedback should be received from stakeholders, which can sometimes drive additional change requests. A retrospective should be completed celebrating success, reflecting on lessons learned, and recommendations for future changes (WhatFix).

At any point in this change control process, if it's determined that the change is too risky or not beneficial, it can be modified, delayed, or canceled. Every person working within this change control system should feel free to raise concerns as they arise. This controlled approach to system changes prevents chaotic, ad-hoc changes from reaching production, ensuring that changes are made in a controlled risk minimized way (TechTarget).

Benefits	Drawbacks
Reduced Risk of Disruption	Potentially Slower Paced
Documented Accountability	Overhead and Costs to Manage the Changes
Improved Coordination and Communication	Risk of Reduced Flexibility
Enhanced Productivity and Focus	Implementation Requires Training and Cultural Adjustment
Better Tracking of Changes	

MeCo has a few vendor tools and platforms to choose from to support the implementation of a formal change control process. One of the most popular and widely used is Jira, by Atlassian. Their change control offering is called Jira Service Management. It is known for being user and developer friendly. It integrates with agile workflows and has

integration with source management tools like GitHub (Atlassian). I'm writing this with the assumption that the developers at MeCo already use a version control system like Git. If not, that is another important aspect of control management that should be implemented. Another option for change control is Freshservice by Freshworks. They market themselves as a Jira alternative with similar change control functionality including templates for change requests, automatic notifications, and meeting management.

Configuration Management

Configuration management is the process of systematically recording and maintaining information about an organization's IT configuration and controlling changes to these in a consistent manner. This includes knowing your IT environment including the hardware used, software, firmware, settings, and documentation. This requires keeping system and configuration information up to date and ensuring that any changes are done in a controlled way (Atlassian). Configuration management helps answer questions like: What is the current setup of our systems? What versions and settings are we running? Where are our servers located and how are they set up? Without a system to track configuration, configuration drift can introduce unknown risks and make future changes extremely difficult. By having a configuration management process, it ensures consistency and visibility into its IT environment (RedHat).

The first step to establish a configuration management system, you first have to determine what needs to be tracked and how it will be tracked. This might include creating a list of configuration items in the organization (like servers, network devices, software applications, OS versions) and gathering data about their key attributes (like their location, versions, and who's responsible for them). This information gathered as part of the planning process is important to establish a baseline of what should be tracked (Wikipedia).

Configuration control is the next step in the process of establishing a configuration management system. This step is closely linked with the change control processes you'll establish. Any change to a configuration item (such as updating to a new OS version, changing a configuration setting, replacing a server component, or making a cloud configuration change) should go through a controlled procedure like change control to ensure that the configuration management documentation is updated accordingly. Over time, the organization will have constant visibility into the current state of the system and a history of all the changes made. This is valuable documentation to reduce chaos in a dynamic and growing organization, maintain flexibility for future changes, and to reduce risk when breaking changes are made as it's easy to revert back to the last working state (Wikipedia).

A configuration management tool should be flexible enough to meet business needs, tracking changes whether you need to manage physical datacenters, public clouds, and edge environments. It should include support for a variety of use cases like networking, security, applications, and provisioning. Integration support for third-party platforms can also be an important consideration when choosing a vendor, along with automated drift detection. These components often operate continuously. When MeCo sets up a configuration management system, it will likely use tools to automate identification, control, and status accounting. As changes happen through change control, the configuration management process captures those changes into the record, maintaining an accurate picture of the IT landscape (RedHat).

Benefits	Drawbacks
Consistent Systems	Initial Setup Complexity
Faster Recovery and Troubleshooting	Maintenance
Easier Change Impact Analysis	Learning Curve/Cultural Change
Auditability	Cost
Efficiency and Automation	

Popular vendor options for establishing a configuration management system are Ansible, Puppet, and Terraform (TechTarget).

Relationship Between Change Control and Configuration Management

Change control and configuration management are closely intertwined processes that complement each other to ensure maintainability, integrity, and reliability of IT systems as a business grows and changes. Both are about introducing structure, control, and visibility into the complex systems required to serve customers digitally. They require documentation, approvals, and review to reap the benefits of knowing the who/what/when/why of the system when it's needed.

The primary differences between change control and configuration management are their focus. Change control is event driven. A change request is created and change control determines how the change is implemented in a trackable controlled way. Configuration management is state driven. Its focus is on understanding the state of the system at any given time (TechTarget).

Implementation Recommendation

In order to implement control change and configuration management, the organization has to consider its requirements. You should consider the needs of the business currently, and what needs it may have in the future to justify this investment. Consider that this will require support from management both in time and resources. The

teams will need to implement the changes and be trained on how to use them. This can influence the culture of the team as new processes introduce additional steps required to complete projects. Team structure may need to be change, and new teams that specialize in and maintain these processes may need to be created. Integrations into current workflows and automations may need to be established.

Change Control and Configuration Management in the SDLC

Integrating change control and configuration management into each phase of the Software Development Life Cycle (SDLC) ensures that best practices are followed from project inception to maintenance. In the planning phase, it will be important to plan for change control (like approvals) and configuration management (like ensuring version updates are tracked). During the design phase, plan alterations will need change control approvals. Documentation and approvals should be tracked under configuration management. During the development phase, code changes should be tracked with version control (like Git) and configuration changes should be documented. The development phase tends to be highly iterative, tracking and approvals ensure ad-hoc changes aren't introduced without documentation. As bug are found in the testing phase, fixes should be tracked. During the deployment phase, it's especially important to ensure that change control and configuration management are updated and available in case any unforeseen issues require the system is reverted to a stable state. During the maintenance phase, audits might be done, patches may be developed, and all of this should be tracked with change control and configuration management.

Conclusion

Establishing dedicated change control and configuration management processes is a strategic imperative for Mesusa Corporation as it grows and scales. Change control provides a structured way to evaluate and implement changes while minimizing risk, ensuring that projects are justified, approved, and clearly communicated. It increases long term reliability and flexibility as the business needs to make changes to adapt to the ever-changing business landscape. Configuration management maintains the integrity and consistency of systems by tracking all components and their states throughout the IT environment. Together, these processes form a strong foundation for reliable operation and future flexibility.

References

1. <https://www.atlassian.com/microservices/microservices-architecture/configuration-management-tools>
2. <https://www.geeksforgeeks.org/project-management-configuration-management-and-change-control/>
3. <https://www.axelos.com/certifications/itil-service-management>
4. <https://csrc.nist.gov/pubs/sp/800/128/upd1/final>
5. <https://linfordco.com/blog/change-control-management/>
6. <https://aws.amazon.com/what-is/sdlc/>
7. <https://whatfix.com/blog/change-control-process/#:~:text=A%20change%20control%20process%20is,stakeholder%20collaboration%2C%20and%20thorough%20documentation>
8. <https://www.redhat.com/en/topics/automation/what-is-configuration-management#:~:text=Configuration%20management%20is%20a%20process,as%20expected%20throughout%20its%20lifecycle>
9. <https://www.techtarget.com/searchdisasterrecovery/definition/change-control#:~:text=Change%20control%20is%20a%20systematic,a%20component%20of%20change%20management>
10. https://www.atlassian.com/software/jira/service-management?campaign=17732529748&adgroup=140639872753&targetid=kwd-362168546815&matchtype=e&network=g&device=c&device_model=&creative=647826614058&keyword=jira%20service%20management&placement=&target=&ds_eid=700000001721198&ds_e1=GOOGLE&gad_source=1&gbraid=0AAAAADAVAKeFRIGrGxK5cmrlvm25qHSQ6&gclid=Cj0KCQiAlbWBhCMARIsADnwasrqJ8iPz4u6SvZtJQ3n1YYRO8hYR5Ja2-n7zSTPLXpbOwGngo4g2FUaAq7KEALw_wcB
11. <https://www.freshworks.com/freshservice/it-service-management/change-management/>
12. https://en.wikipedia.org/wiki/Software_configuration_management#:~:text=%2A%20Configuration%20identification%20,that%20configurations%20contain%20all%20their
13. <https://www.techtarget.com/searchitoperations/feature/Review-the-top-configuration-management-tools-in-DevOps>