



Inteum Company LLC Configuration Management Plan

Version Number: 2.0

Version Date: 2/10/2026



VERSION HISTORY

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
1.0	Preet Mood	12/20/2017	Justin Adair	1/8/18	Initial draft
1.1	Preet Mood	1/18/18			Signatures on Appendix A
1.2	Preet Mood	1/22/19			Reviewed the document to see if any changes are needed. No changes/updates needed.
1.3	Preet Mood	3/7/19			Corrected some spell checks
1.4	Preet Mood	10/31/19			Added revised steps for Bug case and Feature case process under "Building and Testing Changes"
1.5	Preet Mood	1/13/2021			Reviewed the document to see if any changes are needed. No changes/updates needed
1.6	Preet Mood	1/5/2022	Justin Adair	1/5/2022	Reviewed the document and No changes/updates needed
1.7	Preet Mood	1/23/2023	Matt Campbell	1/23/2023	Reviewed the document and No changes/updates needed
1.8	Preet Mood	1/16/2024	Matt Campbell	1/16/2024	Reviewed the document and no changes/updates are needed
1.9	Yavuz Alparslan	1/8/2025	Preet Mood	1/9/2025	Reviewed the document. No changes or updates are required.
2.0	Preet Mood	2/10/2026	Yavuz Alparslan	2/10/2026	Reviewed the document. No changes or updates are required.



TABLE OF CONTENTS

INTRODUCTION.....	4
Purpose of the Configuration Management Plan	4
ROLES AND RESPOSIBILITIES.....	4
Configuration Manager.....	4
System Engineer.....	5
CONFIGURATION MANAGEMENT	5
Configuration Management Tools	5
Training.....	5
CONFIGURATION MANAGEMENT ACTIVITIES.....	5
Configuration Items.....	5
Configuration Identification	6
Software engineering processes	6
Configuration Control.....	6
Configuration Baseline	6
Processing and managing change requests	6
Building and Testing changes.....	7
Identify the need for separation of duties	7
Identify how security impacts of changes will be evaluated	8
Configuration Auditing/Reporting	8
APPENDIX A: CONFIGURATION MANAGEMENT PLAN APPROVAL	10
APPENDIX B: REFERENCES	11
APPENDIX C: KEY TERMS	12



Introduction

Purpose of the Configuration Management Plan

The purpose of the Configuration Management Plan is to describe how configuration management (CM) will be conducted throughout the software development lifecycle. This includes documenting how CM is managed, roles and responsibilities, how configuration item (CI) changes are made, and communicating all aspects of CM to organization stakeholders. Without a documented configuration management plan, it is likely that CIs may be missed, incomplete, or unnecessary work is done because of a lack of version and document control.

The overall objective of a Configuration Management (CM) Plan is to document and inform project stakeholders about CM with the project, what CM tools will be used, and how they will be applied by the project to promote success. The Inteum Company CM Plan defines the project's structure and methods for

- Identifying and baselining configuration items (CI)
- Controlling modifications and releases of CIs
- Ensuring completeness, consistency, and correctness of CIs
- Delivery of the CIs

The intended audience of the CM Plan is the project team, project sponsor and any senior leaders whose support is needed to carry out communication plans.

ROLES AND RESPONSIBILITIES

Roles and responsibilities is an important part of any plan. In order to communicate a clear understanding of expectations, these roles and responsibilities must be clearly defined. Any work that will be performed as part of the plan must be assigned to someone and this section allows us to illustrate who owns these tasks and to communicate them to all project stakeholders.

The following roles and responsibilities pertain to the CM Plan for Inteum Company.

Configuration Manager

The Configuration Manager is responsible for:

- Overall management of the configuration baseline
- Identification of CIs
- Responsible for tracking software components, and building and promoting software releases
- Providing any required configuration training



System Engineer

Each Configuration Item will be assigned to a focus group consisting of one or more engineers.

The System Engineer is responsible for:

- Monitor development for technical integrity and consistency, and compliance with the system requirements and architecture
- Review change requests as they are received and provide technical assessment to the Configuration manager
- Support the Configuration Manager in review and dispositions of change requests

Configuration Management

Configuration Management is the process of systematically controlling and managing all steps of configuration throughout the project lifecycle. In order to effectively handle project CM, it is important to use a process which ensures only necessary configuration changes are made.

Configuration Management Tools

Inteum uses Axosoft, Team Foundation Server (Aka Source Control) for library control, configuration inventory and change history, and status reporting.

Training

This section describes CM training for all project personnel.

Inteum provides on-the-job training regarding configuration controls during the change management process. Roles and responsibilities are reviewed with the following groups.

- Development teams across Inteum
- Business sponsors
- Testing team across Inteum
- Account Managers across Inteum
- Business development team across Inteum

Configuration Management Activities

Configuration Items

Inteum Company Configuration Items are:

- Hardware changes
- Bug fixes
- Features
- Config files
- Database Schemas



Configuration Identification

Identification involves categorizing the type of configurations that are to be controlled. The current configuration items that Inteum CM is tracking include:

- Operating Systems(s) (Windows, Mac, etc.)
- Bug and Security fixes
- Features enhancements
- Support Cases
- Maintenance release changes
- Major Revision Changes
- Database Changes

Software engineering processes

Inteum employ architectural designs, software development techniques, and systems engineering principles that promotes effective information security within organization information systems.

Configuration Control

Configuration control is the systematic evaluation, coordination, approval or disapproval, and implementation of all changes after formal establishment of an item's baseline.

Configuration Baseline

A baseline is a collection of information describing the technical characteristics of each Configuration Item. Baselines serve as technical control points in the life cycle for the evaluation of proposed changes to these technical characteristics. The baseline and the approved changes or modifications provide a current description of the system.

Inteum documents the system baselines that comprise the configuration of the system in the production request system, e.g., Axosoft, Team Foundation Server (Aka Source Control) etc.

Processing and managing change requests

A new feature enhancement request or any issues/bugs identified by the customer or a new software installation are submitted in Axosoft as a Support case under "Help Desk".

- If it is a new "Feature" enhancement request Configuration Manager will review and provide the details to QA Lead to open a related Feature(s) in Axosoft.
- If it is a "Bug/Issue" Configuration Manager will assign the support case to QA Lead. QA will try to reproduce the issue and if the issue was reproduced a new related BUG will be opened in Axosoft.
- If it is a new software installation system administrator will open a related Feature(s) in Axosoft and once the new software installation is done respective "Feature" will be updated and closed by system administrator.

Inteum CM and QA will make sure the support cases are tracked till the closure.



Building and Testing changes

Below are the detailed steps for **bug case** filed in QA environment till deployed to Mainline:

1. Configuration Manager assigns the support case to QA when a support case identifies a bug or requires confirmation as to whether a bug exists.
2. QA will then write up the bug case and then assign that bug case to the appropriate developer. At this point the Workflow Step in the bug case will be "Defect Found"
3. The developer fixes the bug case and keeps the bug case assigned to them and change the Workflow step to "Defect Ready for Code Review" and update the "Code Reviewer" field with developer who would be doing the code review.
4. The code is reviewed following the OWASP coding standards and once the code is reviewed by the developer they will change the Workflow step to "Defect Ready for Testing (Code Review has been Completed)" and change the "Assigned To" field to QA tester.
5. QA will complete the testing and if the bug case is passed QA will updated the Workflow step to "Defect Passed QA (Code has been reviewed and tested)" and assign the case to deployer.
6. During QA testing if QA identifies conditions where the bug is not fixed, QA will add notes to the "Description" field and reset the workflow step to "Defect Re-Opened" and assign it back to the developer who worked on the fix.
7. The developer will start the process again at Step 3 above.
8. When the bug case is passed and QA updates the bug case to "Defect Passed QA (Code has been reviewed and tested)" and assign it to deployer as noted in Step 5 above. The deployer will get notification.
9. Once the deployer deploys the QA passed bug cases from QA to Mainline the deployer will change the Workflow step to "Defect Deployed (Code has been reviewed, tested and deployed)".

Below are the detailed steps for **feature case** filed in QA environment till deployed to Mainline:

1. Configuration Manager assigns the support case to QA when a support case identifies a feature request.
2. QA will then write up the feature case with detail steps and assign that feature case to the appropriate developer. At this point the Workflow Step in the feature case will be "Feature Requested"
3. The developer fixes the feature case and keeps the feature case assigned to them and change the Workflow step to "Feature Ready for Code Review" and update the "Code Reviewer" field with developer who would be doing the code review.



4. The code is reviewed following the OWASP coding standards and once the code is reviewed by the developer they will change the Workflow step to "Feature Ready for Testing (Code Review has been Completed)" and change the "Assigned To" field to QA tester.
5. QA will complete the testing and if the feature case is passed QA will updated the Workflow step to "Feature Passed QA (Code has been reviewed and tested)" and assign the case to deployer.
6. During QA testing if QA identifies conditions where the feature is not fixed, QA will add notes to the "Description" field and reset the workflow step to "Feature Re-Opened" and assign it back to the developer who worked on the fix.
7. The developer will start the process again at Step 3 above.
8. When the feature case is passed and QA updates the feature case to "Feature Passed QA (Code has been reviewed and tested)" and assign it to deployer as noted in Step 5 above. The deployer will get notification.
9. Once the deployer deploys the passed feature cases from QA to Mainline the deployer will change the Workflow step to "Feature Deployed (Code has been reviewed, tested and deployed).

Identify the need for separation of duties

Whenever a production build is ready for deployment dev will hand it over to production deployment manager

Identify how security impacts of changes will be evaluated

The security impacts of changes will be evaluated based on the following criteria:

Area of change	Security Impact
Database change	High
Data Access change	Medium
Business Logic Server Side	Medium
Web Service change	High
User Interface change	Low

Configuration Auditing/Reporting

Audits are an important part of project and configuration management. The purpose of an audit is to ensure that established processes are being followed as intended and to provide an opportunity to correct any deviations from these processes. Configuration audits will be an ongoing part of the configuration management lifecycle. The purpose of the configuration audit is to ensure all team members are following the established procedures and processes for configuration management.



At Inteum auditing is part of the standard SDLC process and is performed during testing lifecycle. Deficiencies founded during informal audits are reviewed and corrective action is taken accordingly. The purpose of the audit is to certify that the design, development, and integration meet the system's technical requirement; that they are accurately documented; and that they do not include unauthorized changes.



Appendix A: Configuration Management Plan Approval

The undersigned acknowledge that they have reviewed the Inteum Company Configuration Management Plan and agree with the information presented within this document. Changes to this Configuration Management Plan will be coordinated with, and approved by, the undersigned, or their designated representatives.

Signature:	Tim Hollobon	Date:	1/23/18
Print Name:	Tim Hollobon		
Title:	VP & Chief Technology Officer		
Role:	Dev Lead		
Signature:	Preet Mood	Date:	1/18/18
Print Name:	Preet Mood		
Title:	Senior Quality Assurance Lead		
Role:	QA Lead		
Signature:	Britt D'Andrea	Date:	1/18/18
Print Name:	Britt D'Andrea		
Title:	VP Business Development		
Role:	Account Manager		



APPENDIX B: REFERENCES

[Insert the name, version number, description, and physical location of any documents referenced in this document. Add rows to the table as necessary.]

The following table summarizes the documents referenced in this document.

Document Name	Description	Location
<i><Document Name and Version Number></i>	<i><Document description></i>	<i><URL or Network path where document is located></i>



APPENDIX C: KEY TERMS

The following table provides definitions and explanations for terms and acronyms relevant to the content presented within this document.

Term	Definition
CM	Configuration Management
CI	Configuration Item
SDLC	Software Development Life Cycle