

ChatMaps Deliverable 5

Configuration Management

Stephen Goodridge, Clark LaChance, Nicholas Pease, Joseph Gallant, Aidan Bradley

COS420

21 April 2024

Github: <https://github.com/ChatMaps/ChatMaps>

Kanban: <https://trello.com/b/TaygvBv7/chatmaps>

Configuration Management

[Table of Contents]

1. Introduction	3
1.1. Purpose	3
1.2. Scope	3
1.3. Key Terms	3
1.4. References	4
2. SCM Management	4
2.1. Organization	4
The ChatMaps project team will implement the SCM activities outlined in this document.	4
2.2. Responsibilities	4
3. SCM Activities	4
3.1. Configuration Identification	4
3.2. Configuration Control	5
3.3. Configuration Status Accounting	5
3.4. Configuration Evaluation and Reviews	5
3.5. Interface Control	5
3.6. Subcontractor/Vendor Control	6
3.7. Release Management and Delivery	6
4. SCM Schedules	6
4.1. Sequence and coordination of SCM activities	6
4.2. Relationship of key SCM activities to project milestones or events, such as:	6
4.3. Schedule either as absolute dates, relative to SCM or project milestones or as sequence of events.	6
4.4. Graphical representations can be used here.	6
5. SCM Resources	7
5.1. Identifies environment, infrastructure, software tools, techniques, equipment, personnel, and training.	7
5.2. Key factors for infrastructure:	7
5.3. Identify which tools are used in which activity.	8
6. SCM Plan Maintenance	8
6.1. Who is responsible for monitoring the plan?	8
6.2. How frequently updates are to be performed?	8
6.3. How changes to the Plan are to be evaluated and approved?	8
6.4. How changes to the Plan are to be made and communicated?	8
6.5. Also includes history of changes made to the plan.	8

1. Introduction

This configuration management plan outlines the policies, tools and procedures that will be used to manage and track changes to the COS420 ChatMaps development project.

1.1. Purpose

The Configuration Management Plan (CMP) provides for the management of configuration items (CIs) in the ChatMaps mobile and web applications. This CMP will account for procedures and policies related to configuration identification, control, status accounting, evaluation, review, interface control, subcontractor management, vendor management, release management, and delivery.

1.2. Scope

This CMP is applicable to all configuration items within the ChatMaps application. This includes software, documentation, and other configurable items deemed necessary.

1.3. Key Terms

Configuration Item(CI): Any component or artifact that is to be managed by the configuration management.

Configuration Management Plan(CMP): A configuration management plan is a document that outlines the policies, procedures, and processes for managing configuration items throughout the software development process.

Version Control (VC): Systems that identify, store, and control access to the different versions of components.

Baseline: A baseline is a defined point in time in the software development process at which a configuration item or set of configuration items is approved for release. These establish a stable foundation for further development and to track changes made to the configuration items over the lifecycle.

Change Request: A formal proposal for a change to the configuration item or a set of configuration items.

Version and Release Identification: Managed versions of a component are assigned unique identifiers when they are submitted to the system. This allows for different versions of the same component to be managed, without changing the component name

1.4. References

Software Engineering- Ian Sommerville- 10th edition Ch 25 pg 731-753 (Configuration management)

2. SCM Management

2.1. Organization

The ChatMaps project team will implement the SCM activities outlined in this document.

2.2. Responsibilities

Project Manager

Responsible for managing the team, coordinating efforts between group members and managing member skills and abilities. Objectives: coordinate efforts, and utilize skills properly.

Configuration Management Team

Responsible for implementing the activities outlined in this document and execution of the SCM activities.

Development Team

Responsible for identifying and documenting CIs, change control, and ensuring delivery is per CMP regulation

3. SCM Activities

3.1. Configuration Identification

3.1.1 Configuration Items:

`./postcss.config.js,./src/components/app/map/geo.js,./src/components/app/friends/page.js,./src/components/app/friends/dm.js,./src/components/app/friends/friends.js,./src/components/app/profile/ProfileRoom.js,./src/components/app/profile/ProfilePanel.js,./src/components/app/profile/Interest.js,./src/components/app/profile/ProfileEdit.js,./src/components/app/datatypes.js,./src/components/app/page/home.js,./src/components/app/page/chat.js,./src/components/app/sidebar/home.js,./src/components/app/sidebar/chat.js,./src/components/app/sidebar/dm.js,./src/components/app/`

notifications/notifications.js,./src/components/app/header.js,./src/app/page.js,./src/app/chat/page.js,./src/app/chat/layout.js,./src/app/register/page.js,./src/app/register/layout.js,./src/app/user/page.js,./src/app/user/layout.js,./src/app/onboarding/page.js,./src/app/onboarding/layout.js,./src/app/login/page.js,./src/app/login/layout.js,./src/app/dm/page.js,./src/app/dm/layout.js,./src/app/app/page.js,./src/app/app/layout.js,./src/app/globals.css,./src/app/layout.js,./package.json,./firebase-config.js,./tailwind.config.js,./capacitor.config.ts,./package-lock.json./public/manifest.webmanifest,./jsconfig.json,./next.config.mjs,./eslintrc.json,./vercel.json

3.1.2 The application will initially launch at V1.0.0, denoted by major releases, minor releases, and bug/security patches. All other releases will follow an incremental model.

3.1.3 Procedures for the acquisition of configuration items will be as follows. Members will make a branch of main to develop a new feature. Once that feature is ready to merge, a request will be opened that must be reviewed by a member of the development team, the project manager or the configuration management team. All changes will be tracked and documented through Github. Documentation of methods and classes will be in the jsdoc format.

3.2. Configuration Control

3.2.1 Change requests must be submitted in writing to the configuration management team and no changes will be implemented without written approval.

3.2.2 Before implementing requested changes, the configuration management team will review all changes and documentation before those changes are completely implemented.

3.2.3 Approvals and Disapprovals will all route through the configuration management team or the project manager.

3.2.4 The configuration team will establish baselines for releases by major milestones.

3.3. Configuration Status Accounting

3.3.1 All configuration status accounting will be completed on Github. Commits will be tracked and reported to the project manager.

3.3.2 All data will be freely available and stored on the team Github repository and organization.

3.4. Configuration Evaluation and Reviews

3.4.1 An audit on a CI will be conducted before a merge to main or a major release

3.4.2 Both the development and configuration management teams will be involved in the process of review before a change to the status data. To approve changes, all verification tests must be passed, both automated and manual.

3.5. Interface Control

3.5.1 Any changes to CIs outside the scope of this document will be conducted as required.

3.6. Subcontractor/Vendor Control

3.6.1 Vercel Hosting managed by the Vercel Company

3.6.2 Firebase Authentication, Database, and Storage services managed by Google Cloud

3.7. Release Management and Delivery

3.7.1 The web application will be automatically built and deployed to Vercel by an automated system on merge to the main branch of the repository

3.7.2 Deployment to the Google Play app store will be conducted by the project manager as required.

4. SCM Schedules

4.1. Sequence and coordination of SCM activities

4.1.1 SCM Activities will be coordinated by the Project Manager.

4.2. Relationship of key SCM activities to project milestones or events, such as:

4.2.1 After initial release, the web application will be released as new features and bug reports are reviewed, tested, and merged to main. Mobile applications will be designed to update with a similar update speed.

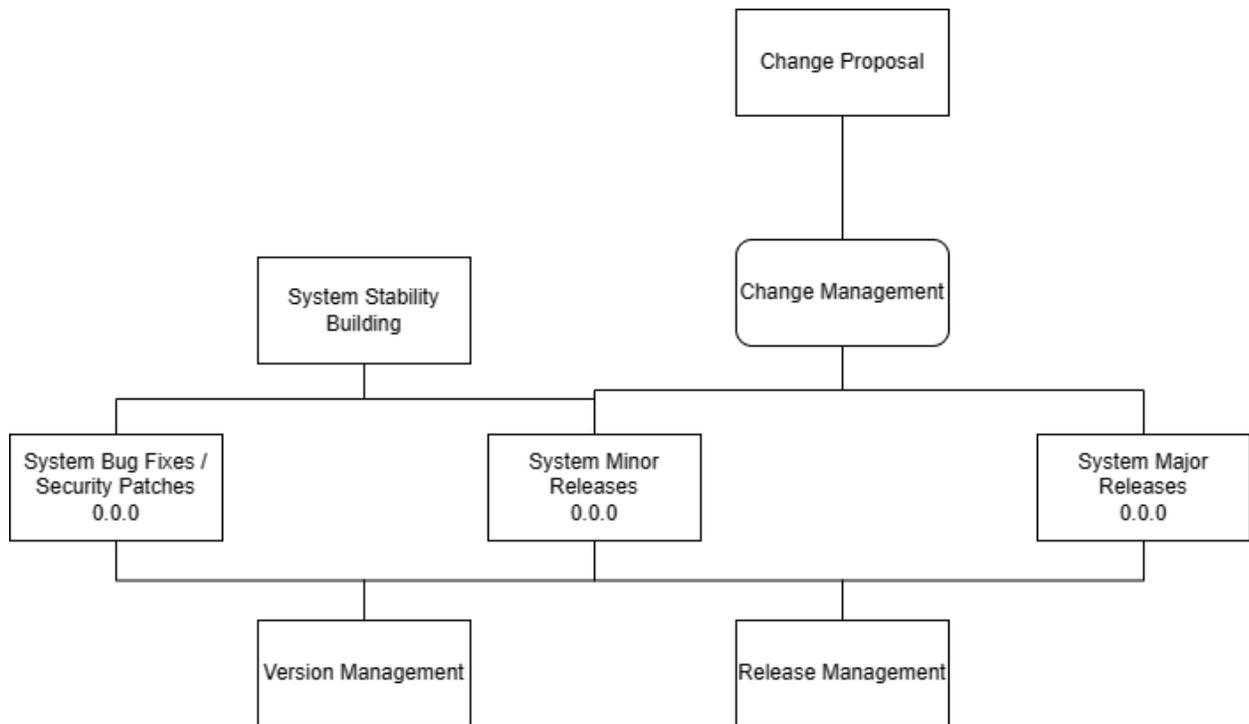
4.2.2 Change control procedures will be determined by the project manager and development team

4.2.3 Configuration audits will have 10 days to be completed as requested by a project manager.

4.3. Schedule either as absolute dates, relative to SCM or project milestones or as sequence of events.

4.3.1 SCM Activities will be scheduled every 6 months after initial release.

4.4. Graphical representations can be used here.



5. SCM Resources

5.1. Identifies environment, infrastructure, software tools, techniques, equipment, personnel, and training.

All change management will be conducted through Git and Github. Vercel will handle hosting the web application. Google Play will handle distributing the mobile application. The frontend and backend will be handled using React with the NextJS framework. Authentication, storage, and database management will be handled by Firebase. All members of all teams will be trained on each of these key resources.

5.2. Key factors for infrastructure:

The system will utilize Firebase for database, storage, and authentication management. This allows our application to scale easily to handle larger user demand as well as higher storage requirements. We anticipate the system will be able to handle 100 concurrent users at our V.1.0.0 release.

Dependencies @capacitor/android==^5.7.4, @capacitor/core==^5.7.4, @capacitor/geolocation==^5.0.7, @emotion/react==^11.11.4, @emotion/styled==^11.11.5, @headlessui/react==^1.7.18, @mui/icons-material==^5.15.14, @mui/material==^5.15.14, firebase==^10.6.0, next==^14.1.0, pigeon-maps==^0.21.3, react==^18.2.0,

react-beforeunload==^2.6.0, react-dom==^18.2.0, react-firebase-hooks==^5.1.1,
react-hook-form==^7.50.1, @capacitor/assets==^3.0.5, @capacitor/cli==^5.7.4,
autoprefixer==^10.0.1, eslint==^8, eslint-config-next==14.1.0, postcss==^8, tailwindcss==^3.3.0

5.3. Identify which tools are used in which activity.

Github (Distributed Control Management System)
Vercel (Preview, CI/Cd, Web Application Distribution)
Firebase (Database, Authentication, File Storage)
React (Frontend, Backend)

6. SCM Plan Maintenance

6.1. Who is responsible for monitoring the plan?

The project manager and configuration management team will be responsible for monitoring the plan

6.2. How frequently updates are to be performed?

Updates will be made every 6 months or as required

6.3. How changes to the Plan are to be evaluated and approved?

Changes to this plan will be submitted in writing to the configuration management team or the project manager for approval.

6.4. How changes to the Plan are to be made and communicated?

Any changes made to the plan will be communicated to the necessary parties by email, updating documentation stored in the configuration management software, as well as, notifying team members on the project Discord.

6.5. Also includes history of changes made to the plan.

Changes	Date	Initials
Initial Plan Created (0.1)	10 APR 2024	NP, CL, SG, JG, AB