

# **Configuration Management Plan**

**Cooking Companion**

**CCP Inc**

**April 23, 2023**

## **[Table of Content]**

- 1. Introduction**
  - 1.1 Purpose**
  - 1.2 Scope**
  - 1.3 Key Terms**
  - 1.4 References**
- 2. SCM Management**
  - 2.1 Organization**
  - 2.2 Responsibilities**
  - 2.3 Applicable Policies, Directives, and Procedure**
- 3. SCM Activities**
  - 3.1 Configuration Identification**
  - 3.2 Configuration Control**
  - 3.3 Configuration Status Accounting**
  - 3.4 Configuration Evaluation and Reviews**
  - 3.5 Interface Control**
  - 3.6 Subcontractor/Vendor Control**
  - 3.7 Release Management and Delivery**
- 4. SCM Schedules**
  - 4.1 Sequence and coordination of SCM activities**
  - 4.2 Relationship of key SCM activities to project milestones**
  - 4.3 Schedule relative to SCM**
  - 4.4 Graphical representation**
- 5. SCM Resources**
  - 5.1 Identifies environment, infrastructure, software tools**
  - 5.2 Key factors for infrastructure**
  - 5.3 Identify tools used**
- 6. SCM Plan Maintenance**
  - 6.1 Who is responsible for monitoring the plan?**
  - 6.2 How frequently are updates to be performed?**
  - 6.3 How changes to the Plan are to be evaluated and approved?**
  - 6.4 How changes to the Plan are to be made and communicated?**
  - 6.5 History of changes made to the plan.**

## 1. Introduction

This Configuration Management Plan outlines the policies, procedures, and tools that will be used to manage and track changes to the software development project. Our team has tailored the plan downward as some of the components do not apply to our project.

### 1.1. Purpose

This Configuration Management Plan(CMP) provides a framework for the management of the configuration items (CIs) within the Cooking Companion web application. The CMP outlines the procedures, policies, and responsibilities for configuration identification, control, status accounting, evaluation and review, interface control, subcontractor/vendor control, release management and delivery.

### 1.2. Scope

This Configuration Management Plan is applicable to all configuration items within the Cooking Companion web application. This includes software, documentation, and any other artifacts that are deemed configuration items.

### 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 Cooking Companion project team will be responsible for the implementation of the SCM activities laid out in this document. The project manager will provide guidance and support for all SCM activities.

### 2.2. Responsibilities

- **Project Manager:**
  - Responsible for overall project management and ensuring that SCM activities are being conducted per the CMP.
- **Configuration Manager:**
  - Responsible for the implementation and execution of SCM activities within the project repository.
- **Development Team:**
  - Responsible for identifying and documenting CIs, QA testing, controlling changes to CIs, and ensuring that everything is delivered as per the CMP.

### 2.3. Applicable Policies, Directives, and Procedures

- Our team has chosen to omit this section as it does not apply to our project.

## 3. SCM Activities

### 3.1. Configuration Identification

3.1.1 Configuration Items: cookingCompanion(0.0.0), app.py, DBAccess.py, requirements.txt, banner.html, bigSearch.html, home.html, layout.html, login.html, query.html, recipe\_page.html, app.css, basic.css.

3.1.2 The Cooking Companion app will initially launch at (0.1.0), our demo version will be version (0.0.5).

3.1.3 Acquiring configuration items will be done through the team's repository on GitHub. Work will be done on separate branches and merged back into the main branch of the repository by the Configuration manager or Project manager. All changes will be tracked through the team GitHub.

### **3.2. Configuration Control**

3.2.1 Change requests will be submitted through a change request form. This change request will be sent to the configuration manager to task the changes to the development team.

3.2.2 Changes will be verified and validated by a QA team.

3.2.3 Approving or disapproving changes will be implemented by the configuration manager.

3.2.4 Baselines will be established at significant milestones for version control by the configuration manager.

### **3.3. Configuration Status Accounting**

3.3.1 All configuration status accounting will be done through the team GitHub. All commits and changes will be tracked and reported to the project manager.

3.3.2 Access to this data will be freely available on the team GitHub. All data will be stored in the team's GitHub repository

### **3.4. Configuration Evaluation and Reviews**

3.4.1 An audit on a CI will take place prior to the release of a major update.

3.4.2 These audits will be processed by our QA team on an as needed basis and prior to any update to the codeline. The results of these will be contained within the system configuration management software used for this project. Change requests will be submitted through a formal change management process and approved by the team's configuration manager.

### **3.5. Interface Control**

3.5.1 Coordination in changes to CIs interfacing items outside of the scope of the plan will be conducted as needed.

### **3.6. Subcontractor/Vendor Control**

3.6.1 MongoDB managed by their Mongo Atlas Service

### **3.7. Release Management and Delivery**

3.7.1 The product will be containerized and released through Docker by the Project Manager.

## **4. SCM Schedules**

### **4.1. Sequence and coordination of SCM activities**

- Coordination of SCM activities will be handled by the project manager and assigned to the configuration manager or the development team.

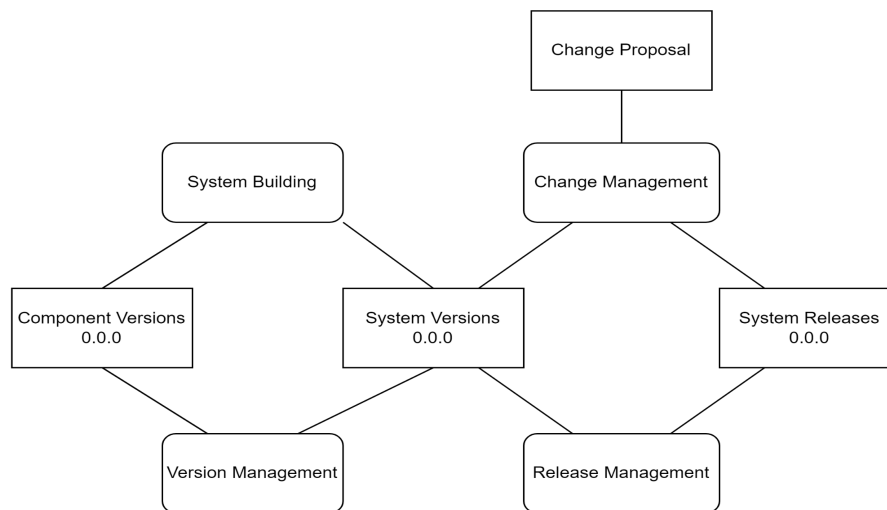
#### 4.2. Relationship of key SCM activities to project milestones:

- After initial launch the configuration baseline is established whenever the configuration manager merges all branches into the main branch. Then to be packaged and released.
- Implementation of change control procedures will be handled by the project manager and configuration manager.
- Once a configuration audit is deemed necessary and assigned by the project manager, it should be completed within 10 days.

#### 4.3. Schedule

- SCM activities will be scheduled every 12-15 weeks after the initial baseline is released.

#### 4.4. Graphical representation



### 5. SCM Resources

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

- All change management will be conducted through Git, MongoDB will handle the database functions, Python and Flask are used to form the backend of the application. While the frontend will be handled in HTML and CSS. Training on all of these systems will be necessary for any project manager, configuration manager, or member of the development team.

#### 5.2. Key factors for infrastructure:

- The system will utilize MongoDB to handle data storage so that our infrastructure can scale with an increasing user base. The system will be built to handle a recipe database of approximately 200 recipes and will scale with its user base. We anticipate that the system will handle approximately 1000 users at release of version 1.0.0.
- *Dependencies:* Flask==2.2.3; pymongo==4.3.3; python-dotenv==1.0.0; python==3.11; unittest

### 5.3. Identify which tools are used in which activity.

- GitHub(Distributed Control Management System)
- Docker(Packaging/Release)
- MongoDB Atlas(Database)
- Python(Backend)
- Flask(Backend)
- HTML(Frontend)
- CSS(Frontend)

## 6. SCM Plan Maintenance

### 6.1. Who is responsible for monitoring the plan?

- The Project Manager and the Configuration Manager will be responsible for monitoring the plan.

### 6.2. How frequently updates are to be performed?

- Updates will be performed every 12-15 weeks or as required.

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

- Changes to this plan will be submitted to the configuration manager and 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. History of changes made to the plan.

- No changes have been made to this plan as of April 23, 2023.

Changes	Date	Initials
Initial Plan Created(0.1)	4/23/23	HM