# Configuration Control Protocol

**Project:** McCoy Speaks  
**Document Type:** Process / Quality Assurance  
**Version:** 1.0  
**Effective Date:** October 9, 2025

## 1. Purpose

This protocol defines how project documents, design artifacts, and code are created, reviewed, updated, and stored to ensure configuration control, traceability, and reproducibility. It replaces any dependency on AI “canvas” or auto-managed document systems. All configuration items are managed manually by the project owner in local and GitHub repositories.

## 2. Configuration Philosophy

* **Control = Trust.** No document or code file should change automatically or without deliberate approval.
* **ChatGPT acts as an assistant, not a repository.** It helps draft, review, and format but never maintains live files.
* **GitHub and local storage are the authoritative sources.**
* **Every version is human-reviewed before commit.**

## 3. Roles

| Role | Description | Authority |
| --- | --- | --- |
| **Project Owner** | Maintains all source files locally and on GitHub. Approves all revisions. | Final authority for commits and releases. |
| **ChatGPT Assistant** | Provides writing, editing, and technical help during sessions. | Suggests, never commits. |
| **GitHub Repository** | Stores official version-controlled artifacts. | Acts as the single source of truth. |

## 4. Document Handling Procedure

### 4.1 Working with ChatGPT

1. Upload or paste the latest document (Markdown, DOCX, etc.) into chat.
2. Discuss edits. ChatGPT assists with rewriting, reorganizing, or adding sections.
3. Approve the draft. ChatGPT produces a downloadable final file.
4. Download and save locally. The project owner verifies version before commit.

ChatGPT will not edit or save live copies; it only provides updated text in session.

### 4.2 Versioning

Each document includes: - Version number (e.g., v1.1, v2.0) - Date of issue - Summary of changes - Author initials or “Approved by” record

Commit tags in GitHub will match the version identifier in the document header.

### 4.3 Storage Locations

| Repository | Purpose |
| --- | --- |
| **Local Folder** | Working copies and in-progress edits. |
| **GitHub Repository** | Finalized and tagged releases. |
| **Backups** | Optional archive of exported .zip or .tar for redundancy. |

## 5. Change Control

| Step | Action | Responsibility |
| --- | --- | --- |
| 1 | Identify need for change | Project Owner |
| 2 | Draft revision with ChatGPT assistance | Project Owner |
| 3 | Review and approve changes | Project Owner |
| 4 | Update version number and changelog | Project Owner |
| 5 | Commit and push to GitHub | Project Owner |
| 6 | Verify sync and backup | Project Owner |

## 6. Changelog Example

| Version | Date | Description | Author |
| --- | --- | --- | --- |
| v1.0 | (today’s date) | Initial release of Configuration Control Protocol | MJT |