**Julio Pochet**  
**CSD370 – Module 9.4 Assignment**  
**Rough Draft: Change Control & Configuration Management at MeCo**

**Introduction**

Mesusa Corporation (MeCo) is growing rapidly, and with that growth comes the need for more control over how software changes are planned, tested, and released. Currently, the company lacks a formal structure to manage system modifications, updates, and patches. As the senior developer, George Everett has asked me to propose a blueprint for implementing a change control and configuration management process to improve system reliability, traceability, and security. This report outlines the structure, benefits, challenges, and recommendations for putting that process in place, along with its connection to MeCo’s secure software development lifecycle (SSDLC).

**Understanding the Processes**

**Change Control**

Change control is the structured approach to managing proposed changes to software, hardware, documentation, or systems. This process ensures that all modifications are evaluated for impact, approved by relevant stakeholders, documented, and implemented in a way that minimizes disruption or unintended consequences.

**Configuration Management**

Configuration management keeps track of all components that make up a system, including code versions, documentation, infrastructure settings, and environments. It ensures that all team members are working with the same system setup, helping eliminate issues like inconsistent builds or environment mismatches.

**How They Work Together**

Change control ensures the right changes are being made for the right reasons, while configuration management ensures those changes are applied consistently and tracked properly. Together, they give MeCo full visibility and control over how its software evolves, which is critical for both quality and security.

**Benefits and Drawbacks**

**Benefits**

* **Improved Stability** – When changes are reviewed and tested before going live, systems experience fewer crashes and rollbacks.
* **Better Security** – Tracking all changes helps detect unauthorized activity and ensures patches are properly handled.
* **Transparency and Accountability** – Everyone knows who made each change, when, and why.
* **Support for Bug Bounty Programs** – If MeCo adopts a bug bounty program, this process will help securely manage externally reported vulnerabilities.

**Drawbacks**

* **Slower Response Times** – Emergency fixes might be delayed if the approval process isn’t streamlined.
* **Learning Curve** – Teams may need training on tools and new documentation procedures.
* **Increased Overhead** – Without automation, managing change logs and version history manually can take up valuable developer time.

**Recommended Implementation for MeCo**

To roll out change control and configuration management efficiently, I recommend the following:

* **Create a Change Advisory Board (CAB)** consisting of a project manager, lead developer, security representative, and QA lead.
* **Adopt a tool like Jira or GitHub Issues** with workflows for submitting, reviewing, and approving change requests.
* **Assign a rotating configuration manager** from the dev team to update version control logs and maintain documentation.
* **Use templates for change requests, rollback plans, and release notes** to simplify the process.

**Integration Across the Secure Software Development Lifecycle (SSDLC)**

| **SDLC Phase** | **Activities Using Change/Config Management** | **Estimated Requests** |
| --- | --- | --- |
| **Requirements** | Document and approve new feature/change requests | 5–10 per sprint |
| **Design** | Review impact of changes on architecture and dependencies | 3–5 per sprint |
| **Development** | Track code changes, create pull requests, update configuration items | 10–15 per sprint |
| **Testing** | Log test environment changes, track bug-related changes | 5–8 per sprint |
| **Release** | Final approval of all release components and deployment changes | 1–3 per release |
| **Maintenance** | Emergency fixes, updates, environment changes | 3–7 per month |

**Conclusion**

A formal change control and configuration management process is essential for MeCo’s next phase of growth. It reduces risk, boosts software quality, and enables the company to react faster and more safely to both internal needs and external reports (like bug bounty submissions). With the right tools and structure, this system can be rolled out efficiently and provide immediate value.

**References**

* Atlassian. (2024). *Change management in Jira*. <https://www.atlassian.com/itsm/change-management>
* Microsoft. (2023). *Configuration Management in DevOps*. <https://learn.microsoft.com/en-us/shows/devops-fundamentals/configuration-management>
* OWASP. (2023). *Secure Development Foundations*. <https://devguide.owasp.org/en/02-foundations/02-secure-development/>
* Red Hat. (2024). *Driving change with a human-centered approach*. <https://www.redhat.com/en/blog/driving-change>
* Tonic.ai. (2024). *Guide to synthetic test data generation*. <https://www.tonic.ai/guides/guide-to-synthetic-test-data-generation>
* Webvizio. (2024). *How to write an effective bug report*. <https://webvizio.com/blog/how-to-write-an-effective-bug-report/>