***Configuration Management Plan***

***Version 3***

**TABLE OF CONTENTS**

[**INTRODUCTION**](#_aeotdcjovxxt) **3**

[**SOFTWARE CONFIGURATION ITEMS**](#_7fa8l2mz6f0c) **4**

[**OTHER SOFTWARE**](#_bto7hp9gwshs) **4**

[**CONFIGURATION AND PROJECT RELATIONSHIP**](#_ysi1czqomubm) **4**

[**CONFIGURATION APPLICATIONS**](#_5nkx0a7bsl09) **4**

[**CONFIGURATION ASSUMPTIONS**](#_3cn59rjfvzva) **4**

[**CONFIGURATION MANAGEMENT**](#_dn67788tjmom) **4**

[**ORGANIZATION**](#_j7kqxk2h3vq8) **5**

[**RESPONSIBILITIES**](#_7v2srbgkw2wy) **5**

[**APPLICABLE POLICIES, DIRECTIVES, AND PROCEDURES**](#_c21jux8t2ayv) **5**

[**CONFIGURATION ACTIVITIES**](#_ttl3s69dgvr0) **5**

[**CONFIGURATION IDENTIFICATION**](#_wi3v515hfic9) **6**

[**CONFIGURATION CONTROL**](#_lv095b6iywga) **6**

[**CONFIGURATION STATUS ACCOUNTING**](#_vibygegrjb79) **6**

[**CONFIGURATION EVALUATION AND REVIEWS**](#_rvggkkb4f0iy) **7**

[**INTERFACE CONTROL**](#_paxeq9z4tome) **7**

[**VENDOR CONTROL**](#_w0q5pdl2645a) **7**

[**RELEASE MANAGEMENT AND DELIVERY**](#_pxoj2vop8u81) **7**

[**CONFIGURATION SCHEDULE**](#_f8vwzv6tnfwv) **7**

[**CONFIGURATION ACTIVITIES AND MILESTONE RELATIONSHIP**](#_s5v77a6ueyib) **7**

[**SOFTWARE CONFIGURATION RESOURCES**](#_qq5mezabwail) **8**

[**CONFIGURATION PLAN MAINTENANCE**](#_ve8ai4n2lkqa) **8**

# **INTRODUCTION**

Our software project is building a web application, InvestiMapp, from which users can sign in to track real-time progress for their stock market portfolio.

## **SOFTWARE CONFIGURATION ITEMS**

* Stock Market data API (real-time and historical data)

## **OTHER SOFTWARE**

* Amazon Web Services’ CodeStar (Version/Release Control)
* Amazon Web Services’ BeanStalk (Application Hosting)

## **CONFIGURATION AND PROJECT RELATIONSHIP**

This configuration plan details the various software, components, and libraries that will be used in our project. The goal is to control development, codeline and baseline versions and releases.

## **CONFIGURATION APPLICATIONS**

This Configuration Management Plan’s aim is to set a general direction for our application development. With an informal depth of control, the developers should follow this plan as guidelines for managing versions and releases of the project.

## **CONFIGURATION ASSUMPTIONS**

This plan assumes a high level of flexibility due to the timeframe of the project, as well as the schedules of those involved. Customer interaction will be limited to Deliverable feedback.

# **CONFIGURATION MANAGEMENT**

## **ORGANIZATION**

* Code pipelines (codeline, baseline, and mainline) are handled by Amazon Web Services
* Developers commit code, from which we can test in a closed environment. When a new version release is ready, we can release all added changes simultaneously with the click of a button in the CodeStar environment.

## **RESPONSIBILITIES**

* Product Owner
* To maintain the list of product features, and mark their progress
* Part of the development team, but posses more of an overseer’s role
* In effect for the duration of one sprint.
* Scope includes the main features being implemented during each sprint.
* Maintain, share, and advises on product features and implementations
* Scrum Master
* To keep track of team progress, overseer and handler of each sprint.
* Part of the project team, manages overhead.
* In effect for the duration of one sprint.
* Handles the scrum duties; conducts meetings and promotes teamwork for solving issues.
* Hold daily scrum meetings
* Developers
* To develop the application and implement project features.
* Core of the project development team.
* Duration of the entire project
* Develop features and commit changes.
* Implement features and maintain the sprint backlog document.

## **APPLICABLE** POLICIES**, DIRECTIVES, AND PROCEDURES**

* User Data Protection Requirements
* Federal Regulations for the Stock Market

# **CONFIGURATION ACTIVITIES**

## **CONFIGURATION IDENTIFICATION**

* Amazon Web Services’ CodeStar
* Pipeline
* Version Control
* Repository
* Rollbacks

## **CONFIGURATION CONTROL**

* Requesting changes:
* Create a pull request to the repository linked with Amazon Web Services CodeStar
* Evaluating changes:
* A developer (differing from the pull request originator) will evaluate the pull request, and upon evaluation, will accept the request into the test environment.
* Change approval:
* From the test environment, begin verification and validation processes.
* If the change is declined, revert to baseline prior to the pull request.
* Implementing changes
* Once new features have been tested and validated, the development team will collectively decide to release the newest mainline to the public URL of our web app through CodeStar.

## **CONFIGURATION STATUS ACCOUNTING**

* All software, systems, and components will be tracked through AWS CodeStar, and our GitHub configuration repository.
* All included configuration components will have details noting where, what, and why each configuration item is used.
* Each developer will automatically have access to control and view the status of all configuration items through CodeStar. Those outside of the development team will be invited to the repository with all relevant documentation.

## **CONFIGURATION EVALUATION AND REVIEWS**

* Prior to release, all configuration items need to be inspected by 3 individuals. The objective of these inspections is to ensure the addition of configuration items (CI) that benefit the application, progress development, and do not compromise existing systems:
* Product Owner: Judge the CI based on its effectiveness towards implementing a feature
* Scrum Master: Judge the CI based on its contributions to reaching a sprint goal
* Developer: Judge the CI based on its merits to the overall health of the application, and compatibility with the existing system.

## **INTERFACE CONTROL**

* All configuration items will be heavily tested to ensure no existing items outside of this plan lose functionality.

## **VENDOR CONTROL**

* Due to the scope and budget of the project, all included items originating from outside the project environment will be open source packages, components, APIs, etc.
* Only those that can be worked in to the project in their initial state will be included, and no contact with outside vendors is expected.

## **RELEASE MANAGEMENT AND DELIVERY**

* Control of build versions is handled through CodeStar. Version pipelines are designated as private or public. Private pipelines include features in development, and features currently being tested.
* Once a private pipeline has been validated and verified for release, no less than three project team members will collectively decide to push the private pipeline for public release through CodeStar’s UI.

# **CONFIGURATION SCHEDULE**

## **CONFIGURATION ACTIVITIES AND MILESTONE RELATIONSHIP**

* Sprint 2: Implemented the UI baseline and included SDKs.
* Sprint 3: Completed database schema

# **SOFTWARE CONFIGURATION RESOURCES**

* Our central environment is Amazon Web Services; CodeStar. CodeStar is a simple way that Amazon has set up some simple functionality for quick implementation. From here we can access source code and mainlines. It also allows for real-time updates and releases.
* Individual developers can use whichever IDE they wish, so long as they are capable to pushing to the repository.
* The infrastructure of our application results from AWS pipelines and BeanStalk, which handles everything natively.
* Functionality maintained through CodeStar
* Performance and availability handled by AWS servers
* Security and Safety handled by browser cache
* Equipment and Costs are all managed through AWS
* No extraneous equipment is required.

# **CONFIGURATION PLAN MAINTENANCE**

* Plan monitoring
* Handled by entire development team, Product Owner and Scrum Master to take majority of responsibility.
* Updates
* Should be updated at least once every sprint.
* Update approval
* Updates should be evaluated by at least 3 members of the team, and approved based on the health and contributions to the project plan.
* Update communication
* Once an update has been approved, an announcement shall be made during the next scrum meeting, and the update originator shall make the changes.

**DOCUMENT CONTROL LOG**

|  |  |
| --- | --- |
| **Title:** | Configuration Management Plan |
| **Deliverable:** | Deliverable 2 |
| **Date:** | March 7th, 2018 |
| **Author:** | Wesley Scott |
| **Distribution:** | InvestiMapp Group |
| **Reference:** | Lecture 11 & 12 |
| **Filename:** | InvestiMapp\_Deliverable\_3\_CMP\_2.pdf |
| **Control:** | Reissue as complete document only |

**DOCUMENT SIGNOFF**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nature of Signoff** | **Person** | **Signature** | **Date** | **Role** |
| Author | Wesley Scott |  | 3/7/2018 | Scrum Master |
| Reviewer | Ryan Hutchinson |  | 3/8/2018 | Developer |
|  |  |  |  |  |

**DOCUMENT CHANGE RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| 4/14/2018 | 2 | Ryan Hutchinson | Changed from Cognito to AWS, added some more details. |
| 5/10/2018 | 3 | Wesley Scott | Changed user authentication to cached/persistent data. |
|  |  |  |  |
|  |  |  |  |