***Algorithmic Trading System***

***Software Project Management Plan***

Algorithmic Trading Software

2024-04-09

Version 1.1

Authors: Jacob Rawlings,Jake Fischer**,** Alan Abdollahzadeh, Devon Volberg, William Blake, Benjamin Carrier, Vanessa Dubouzet, Dakota Flath, Parker Green, Yuan Hu, Jacob Labelle, Isaac Lengacher-Bergeron , Dominic Presch, Dylan Soares, Jaeden Soukoroff

[**1 Introduction 1**](#_heading=h.3znysh7)

[**1.1 Project Overview 1**](#_heading=)

[**1.2 Project Deliverables 1**](#_heading=)

[**1.3 Document Overview 1**](#_heading=)

[**1.4 Acronyms and Definitions 1**](#_heading=)

[**1.5 References 1**](#_heading=h.buo7f2ej8upx)

[**2 Project Organization 3**](#_heading=)

[**2.1 Organizational Policies 3**](#_heading=h.lnxbz9)

[**2.2 Process Model and Organizational Structure 3**](#_heading=)

[**2.3 Organizational Boundaries and Interfaces 3**](#_heading=)

[**2.4 Project Responsibilities 4**](#_heading=)

[**3 Managerial Process 4**](#_heading=h.z337ya)

[**3.1 Management Objectives and Priorities 5**](#_heading=)

[**3.2 Assumptions, Dependencies, and Constraints 5**](#_heading=)

[**3.2.1 Assumptions 5**](#_heading=h.4i7ojhp)

[**3.3 Risk Management 5**](#_heading=)

[**3.4 Monitoring and Controlling Mechanisms 5**](#_heading=)

[**3.4.1 Project Monitoring 5**](#_heading=h.90vd73qyueef)

[**3.4.2 Project Controlling 6**](#_heading=h.wqm8i5t07tzv)

[**3.5 Staffing Plan 6**](#_heading=)

[**4 Technical Process 6**](#_heading=)

[**4.1 Tools, Techniques, and Methods 7**](#_heading=)

[**4.2 Software Documentation 7**](#_heading=)

[**5 Work Packages 8**](#_heading=)

[**5.1 Work Packages 8**](#_heading=)

[**5.2 Dependencies 9**](#_heading=)

[**5.3 Resource Requirements 10**](#_heading=)

[**5.4 Budget and Resource Allocation 10**](#_heading=)

[**5.5 Schedule 11**](#_heading=)

[**6 Additional Components 11**](#_heading=)

[**Configuration Management plan 11**](#_heading=h.v5doqv8l6sc9)

[**Software Test Plan 11**](#_heading=h.qskv644v2ucz)

[Project Plan 11](#_heading=h.w1bz75snb71z)

**Revision Sheet**

| Revision Number | Date | Brief summary of changes |
| --- | --- | --- |
| 001 | 2023-11-19 | Initial baseline draft |
| 002 | 2023-12-07 | Review and Revisions (Release 1.0) |
| 003 | 2024-01-26 | Updated sections with new design and PM considerations |

# **Introduction**

## **Project Overview**

The ATS project is tasked with gathering and storing large quantities of financial market data for future analysis by market researchers. The project is sponsored by Okanagan College, and will operate on IONOS which is a web based cloud server. This project’s initial release will be delivered by 7th December, 2023. Following this, there will be a maintenance and continuous integration period where subsequent iterations will be released until 10th April, 2024. The requirements for our project include pulling data from a financial market tracking source, then storing that data so that it can be used with a ML model for market forecasting.

## **Project Deliverables**

The ATS must deliver a working database containing current and historical financial market data. The ATS configuration must be modifiable, and must be capable of supporting alternative REST API’s. A User Interface which allows for managing this system configuration must also be delivered. Data must be retrievable from the database by the customer in common formats such as .csv and .sql through the web interface of IONOS.

## **Document Overview**

This document summarizes and defines project organization, managerial process, technical process, work packages, and additional management resources related to the production phase of the ATS. The project organization details the policies, boundaries, and responsibilities of the members of the ATS. The managerial process outlines risks, objectives, and constraints of the project and its creators. The technical process outlines the tools and documentation used to create and track changes to the ATS. Work packages outline the process of workflow throughout the ATS creation and details of its steps.

## **Acronyms and Definitions**

ATS - Algorithmic Trading System

SRS - System Requirements Specification

SDD - Software Design Document

CMP - Configuration Management Plan

CI - Continuous Integration

XP - Extreme Programming

## 1.5 References

[System Requirements Specification](https://docs.google.com/document/u/0/d/1CbAF2mOfkrObv0FZ-JKMiEawvmQiZl1uLMKGMOtAUPg/edit) [R1]

[Software Design Document](https://docs.google.com/document/u/0/d/1kH4S7RFlHHq6SzOlhkYMcuc-0IA7yYNazzqXoAM9lVY/edit) [R2]

[Software Test Plan](https://docs.google.com/document/u/0/d/1G8IDT8RH3BMlVuHbDmPxUAnFdIvpo8_n-6HjcZbuiF0/edit) [R3]

[Configuration Management Plan](https://docs.google.com/document/u/0/d/1fIw9oy5tjoy4-G9NGvU28-ZwWCqFCC63k-rCmbOEJbo/edit) [R4]

[Risk Management Plan](https://docs.google.com/document/d/1SfLjt5y5TFBPU2QsOqfKHZAo21i6bhmHX7k90aghkqU/edit) [R5]

[Team Contract](https://docs.google.com/document/u/0/d/1jcJq0qhL_UHWdKtmhirTRYhIrrTmR3fKwvvt-EZeRP0/edit)[R6]

[Change Request Form](https://docs.google.com/document/d/15cJJYKJDISOaI6XpkidXRFpBuVadP0pqV-8OSKZeWto/edit)[R7]

# **Project Organization**

## **Organizational Policies**

The ATS team members must adhere to the team contract policies outlined in the Team Contract [R6]. The ATS organizational policies are located in the Risk Management Plan [R5], Configuration Management Plan [R4], and Software Test Plan [R3].

## **Process Model and** Organizational Structure

The ATS organization operates on an Agile methodology with a primary focus on Scrum, which is a horizontal structure which assigns Product Owners, Scrum Masters and developers. Product Owners act as an interface between the team and clients. They keep frequent communications with the client, receiving feedback, and translate requirements to the development team via prioritizing backlog items. Product Owners are also responsible for reviewing change requests and assessing risks. Scrum Masters recognize tasks, assign tasks to the development team members and stay informed of development progress. Developers implement the functionality of the system and document any unresolved queries or concerns that may arise during the process. Developers also engage in XP practices such as pair programming. Each sprint, developers will form pairs and work with each other to improve code quality, allow for real-time review, and collaborative problem solving.

## **Organizational Boundaries and Interfaces**

* Boundaries between ATS organization and college
  + Okanagan college supplies the ATS project with API keys to collect data
  + Okanagan college supplies the ATS project with Cloud server access
* API interface
  + API access provided by Okanagan college retrieves data from Financial Modelling Prep
* Customer interface
  + Customers access the OLTP database from IONOS interface
  + Customers interact with the system configuration from the ATS UI
* Jira
  + Provides the workflow interface for the development teams.
  + Tracks work duration and dates.
  + Assigns tasks to developers.
* Google Drive
  + Cloud Storage for system documentation.
  + Collaboration on documents.
* GitHub
  + Version control for the ATS project
  + Git Actions for CI
* IONOS
  + Interface used to host cloud servers for data storage.
* PhpMyAdmin
  + Interface used to access the database.

## **Project Responsibilities**

* **Requirements Gathering:** Product Owners communicate with the clients and define the requirements, potential features and scope of the project.

**Responsible Actor(s):** Product Owner

* **Design:** Scrum Masters and Product Owners describe the product/system and assign documentation tasks to developers

**Responsible Actor(s):** Product Owner, Scrum Master, Developer

* **Implementation:** Scrum Masters create and create and assign tasks for developers that they choose from to complete.

**Responsible Actor(s):** Scrum Master, Developer

* **Testing:** Developers test the functionalities through unit tests and Product Owners conduct acceptance tests with the clients prior to the release.

**Responsible Actor(s):** Product Owner, Developer

* **Documentation:** Developers create IEEE documents for code, architecture, design, and the end user.

**Responsible Actor(s):** Developer

# **Managerial Process**

## **Management Objectives and Priorities**

Product Owners and Scrum Masters will be tasked with the responsibility of overseeing and managing the project schedule. Their duty is to ensure that all project tasks are completed adequately and on schedule. Additionally, these leaders will take charge of generating and organizing all project deliverables, as well as conducting meetings as part of their responsibilities.

## **Assumptions, Dependencies, and Constraints**

### 3.2.1 Assumptions

A number of assumptions have been made at the outset of the project.

| No. | Assumption | Implications |
| --- | --- | --- |
| 1. | Dynamic Requirements | It is assumed that project requirements are subject to modification. Changes to requirements may impact the project timeline.scope, product delivery delays, and staff workload. Accounting for this in the project management activities is vital to success. |
| 2. | Availability of resources | Changes to the availability of resources (staff, budget) will impact the project timeline. Delays to task completion as a result of increased workload on available staff, as well as the need to reallocate resources. |
| 3. | Stability of technology | Changes to technology availability will impact the project timeline. Unavailability of the chosen tools and technologies will cause delays and fiduciary concerns while acquiring alternatives. Lesser alternative tools may also have an impact on final product quality. |

## **Risk Management**

Please refer to the ATS Risk Management Plan [R5]

## **Monitoring and Controlling Mechanisms**

### 3.4.1 Project Monitoring

Progress tracking will primarily be performed using Jira. Jira acts as a centralized platform for project management with tasks, user stories, and issues representing work to be done. Jira will provide a clear method to track project progress during each sprint and throughout the project. During sprint planning, tasks, stories, and issues will be created, estimated, and prioritized, with story points assigned to each. Members of the development team will be assigned to each during sprint planning. Tasks, user stories, and issues will be updated by the assignees to reflect current status and completion.

In addition to Jira, regular code reviews and testing will be performed throughout the development lifecycle to monitor the quality of components, as specified in the Software Test Plan.

### 3.4.2 Project Controlling

Establishing a project controlling framework ensures that the project is effectively monitored and remains adaptable throughout its lifecycle.

A formal change control process will be implemented. All changes to project scope, schedule, or resources must be submitted through the Change Request Form [R7]. Changes submitted through the Change Request Form [R7] will be reviewed by the Change Control Board for approval or rejection. Approved changes will be documented and integrated into the project plan.

Jira provides issue tracking that will be used to log and manage project issues. The Product Owner will review completed issues, bringing them to relevant stakeholders for resolution.

Git Actions will be leveraged during this project's maintenance phase as a CI tool, in order to automate the integration and testing process. This enables code changes to be seamlessly implemented into a shared repository on a regular basis. Essentially, automated testing is done with each commit, allowing for issues to be detected early and maintaining a stable system.

## **Staffing Plan**

**Project Owner**

**Responsibilities:**

* + Act as a liaison between clients and developer team
  + Provide a vision for the project

**Scrum Master**

**Responsibilities:**

* + Manage Jira and assign tasks to developers
  + Run Scrum meetings

**Developers**

**Responsibilities:**

* + Implement data retrieval and data storage functionality
  + Ensure code quality and maintainability
  + Collaborate with other team members

# **Technical Process**

## **Tools, Techniques, and Methods**

**Methods**

* Agile Development - this methodology was chosen due to its incremental approach and adaptability to dynamic requirements.
* Modular Design - software is separated into independent components that each represent a specific functionality. Allows for easier scalability and maintenance. For the ATS, code modularity comes in the form of:
  + Separate configuration files
  + Independent data collection components
  + Independent database insertion/update scripts
* Gitflow - To facilitate version control for the development process, the Gitflow architecture was chosen.
* REST - The system employs RESTful principles, to integrate with external RESTful APIs
* Rapid Prototyping - Script-based design is heavily utilized to facilitate rapid prototyping and automation

**Tools**

* Python is the primary programming language used in development of the ATS.
* PhpMyAdmin acts as a GUI tool for managing the OLTP database.
* GitHub is utilized as a repository for source code to allow for version control and collaborative development.
* Jira serves as the task tracking and Agile project management tool.
* Google Drive is utilized for collaborative work and storage of documentation, diagrams, and other non-software work products
* Git Actions acts as a continuous integration tool

For a detailed breakdown of software design, please see the SDD [R2]. For testing and integration details, see the STP [R3]. To see the methods used for modifying project deliverables, see the CMP [R4].

## **Software Documentation**

**Required Documentation and Milestones:**

* System Requirements Specification: Expected by November 5, 2023 *(DONE)*
* Software Design Document: Expected by November 5, 2023 *(DONE)*
* Risk Management Plan: Expected by November 5, 2023 *(DONE)*
* Configuration Management Plan: Expected by November 5, 2023 *(DONE)*
* Acceptance Test Document: Expected by November 5, 2023 *(DONE)*
* Project Management Plan: Expected by November 19, 2023 *(DONE)*
* ATS User Manual: Expected by December 7, 2023 *(DONE)*
* Developers Guide: Expected by December 7, 2023 *(DONE)*
* Business Case Document: Expected by December 7, 2023 *(DONE)*
* Environment Configuration: Expected by January 31, 2024 *(IN PROGRESS)*
* UI Prototype: Expected by February 14, 2024 *(TO DO)*
* Redundant Data Source Support: Expected by February 28, 2024 *(TO DO)*
* Improved Logging System: Expected by March 13, 2024 *(TO DO)*
* Functioning CI: Expected by March 13, 2024 *(TO DO)*
* Finalized UI: Expected by March 13, 2024 *(TO DO)*
* End-User Testing: Expected by March 27, 2024 *(TO DO)*
* Final Release: Expected by April 10,2024 *(TO DO)*

Documentation for the ATS Project is constructed based on IEEE standard templates. Baseline versions of these documents are created collaboratively by the development team, with sections being written in pairs. When these sections are ready for review, another pair will review and provide feedback or suggestions. The ATS Product Owners have the final sign-off on software documentation before it is released.

# **Work Packages**

## **Work Packages**

Work Package 1: Define the project scope

* Develop Project Charter
* Identify Stakeholders
* Define project objectives
* Create Risk Management Plan

Work Package 2: Requirements Gathering

* Collect User Stories
* Create a Software Requirements Specification document
* Review requirements with stakeholders

Work Package 3: Design Phase

* Database Design
* Data warehouse Design
* Design Configuration Files
* Design the system architecture
* Create Software Design document
* Review and approval of the design

Work Package 4: Development Phase

* Creating query scripts
* Creating database insertion scripts
* Testing(unit and integration)
* Debugging and code reviews
* Development documentation

Work Package 5: Quality assurance

* Functional testing
* Usability testing
* Performance testing
* Bug fixing

Work Package 6: Deployment

* User training
* System review with stakeholders
* Software release

Work Package 7: Maintenance

* Bug fixing
* System observation
* Customer support

Work Package 8: Inception

* Review and revise existing documentation
* Review and revise system analysis and design
* Requirements gathering for new functionality

Work Package 9: Elaboration

* Configure updated development environment
* Review project architecture
* Investigate Continuous Integration Tools
* Initial design for new functionality

Work Package 10: Construction 1

* Implement Continuous Integration
* UI Prototype Development
* Test development

Work Package 11: Construction 2

* Update system configuration processing
* UI prototype feedback and continued development
* Alternate data source support

Work Package 12: Construction 3

* User manual update
* Developers guide update
* Logging system elaboration

Work Package 13: Transition and Refactoring

* End-User testing
* Performance testing and reports
* Bug reporting and fixes

Work Package 14: Maintenance

* Live system monitoring
* Address and resolve post-deployment issues
* Active bug fixing and stress testing
* Finalize project for client delivery

## **Dependencies**

* Project Scope must be defined before the Requirements Gathering Phase can be started
* Design Phase can start being developed near the end of the Requirements Gathering
* Development Phase can be started once the Design Phase has started
* Quality Assurance will be started 1 week after Development Phase and continue after development has completed
* The Deployment will be reviewed with customer to meet Quality Assurance, once it meets the quality expectation the Deployment phase will conclude
* Maintenance depends on the system being deployed but, will overlap with deployment to ensure the system functions as intended

## **Resource Requirements**

Project Duration: 22 weeks

Personnel:

* Project Owner: Full-Time throughout the project
* Scrum Master: Full-Time throughout the project
* Development Team: Full-Time throughout the project
* Customer: 22 weeks Part-Time during Scope Definition, Requirements Gathering, and Deployment

Work hours:

* Development and testing environments:
  + Throughout the project: Computer access required for development and testing

Computer Hardware:

* Server and Hosting
  + Throughout the project: Required server hosting for the development and deployment of the project

Office and Laboratory Facilities:

* Meeting Rooms:
  + Throughout the project: Required meeting rooms for discussion and presentations

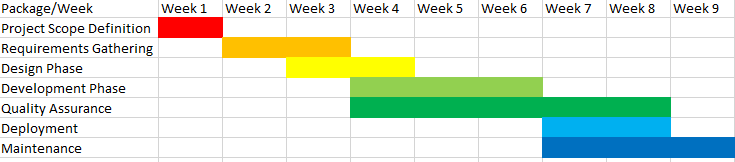
Maintenance:

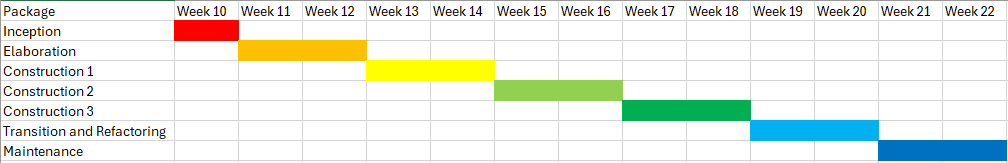
* Ongoing support for the project’s software and infrastructure

## **Budget and Resource Allocation**

Not Applicable to our project (Student lead project)

## Schedule





# **Additional Components**

## Configuration Management plan

This plan defines version control management, system formation, change management and release management during the development of the ATS project.

## Software Test Plan

The development team follows the framework to test and recast that the systems’ functionalities are in line with the requirements.

## Project Plan

This document acts as a plan for the project execution and schedule. It outlines the scope, milestones, and phases of the project.