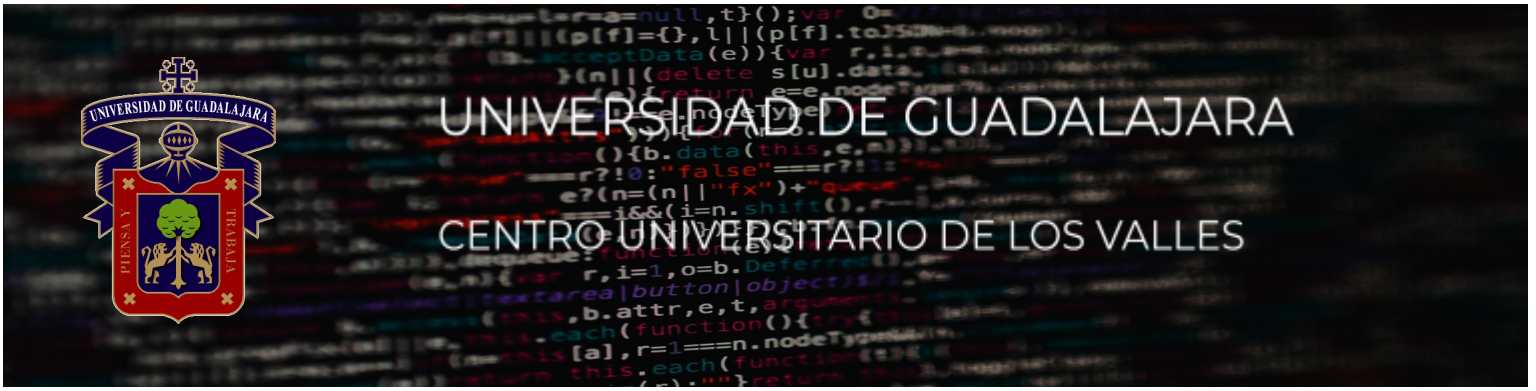


Software Configuration Management

Author: Josué Isaías Zepeda Martínez

Degree: Master in Software Engineering

Professor: Phd. Omar Ali Zatarain Duran

3rd Semester 2023-B

December 1st, 2023

How to read this document

This document has been written with some documents attached. The attached documents correspond to the final version of each document. Past versions of these documents are referenced in the Status Accounting part, which are not attached but can be found in the GitHub repository with the links found in the references of this document.

Table of Contents

Introduction

In this document, the development journey of the Algorithmic Trading Software project is chronicled, from its conceptualization to the refined and finalized version. This narrative encapsulates pivotal moments, strategic decisions, and the iterative process that has shaped the project.

The journey commenced with the initiation of a unique software development project. The vision was to craft an Algorithmic Trading AI with machine learning capabilities, empowering intelligent investments across diverse markets. Responding to valuable recommendations, the project scope was improved to focus on creating a configurable algorithmic trading software, ensuring a more precise alignment with the objectives. Complementary visual representations, available on GitHub as use case diagrams, succinctly outline the intended functionality of the software.

The initiation of the first "Change Request" prompted essential adaptations. Crucial project context was established, detailing team composition, financial considerations, and budgetary allocations. A critical milestone was achieved with the formulation of company policies. These policies defined Committee Board members, outlined a decision-making hierarchy, and established the groundwork for effective governance. Feedback received during class sessions prompted iterative refinements, applied not only to the policy document but also to specific "Change Requests," reflecting a commitment to continuous improvement.

An update to the company's policy document added clarity to roles and responsibilities within the team. It also introduced an evaluation of project outcomes against the initial budget, providing valuable insights. The document presented here encapsulates the dynamic narrative of the project, providing a concise yet comprehensive overview of collective efforts and the evolving nature of the software development.

In this final presentation, the ultimate versions of the Master Document and Company Policies are presented. These documents signify the culmination of collective efforts and the realization of a dynamic software development journey.

Software Configuration Management

According to [1], Software Configuration Management (SCM) is one of the disciplines which grew in response to the many failures of the software industry throughout the 1970s. SCM is a process that focuses on managing changes to Software Configuration Items (SCI's) in all of their representations. also mentions that this paper will focus on the discipline of SCM by first placing it in its proper context with respect to the rest of the software development process, as well as to the goals of that process. further explains that SCM involves a process of managing changes to SCI's.

Configuration Identification

Configuration Identification is the process of identifying and labeling the constituent components of a software system. The most elementary entity in the software configuration identification labeling mechanism is the software configuration item (SCI). Viewed from an SCM perspective, a software line appears as a set of SCI's. The SCI's within a baseline are related to one another via a tree-like hierarchy. As the software system evolves through its life cycle, the number of SCI's and their interrelationships change.

The following document was established as a baseline for the development of the project and is an example of Configuration Identification. [2]

Baseline

Project: Development of an Algorithmic Trading Program

Date: September 5, 2023

Project Vision:

To develop an algorithmic trading program that allows users to design, test, and execute automated investment strategies in financial markets, with the goal of improving the efficiency and profitability of financial operations.

Project Objectives:

1. Design and develop a functional and scalable algorithmic trading program.
2. Provide users with an intuitive platform for creating and programming custom trading strategies.
3. Integrate an API from at least one trading platform for automated order execution.
4. Ensure data and financial transaction security.
5. Implement risk management and investment protection measures.
6. Provide real-time monitoring and analysis tools.
7. Offer comprehensive documentation and user support.

Project Scope:

The project scope includes the design, development, testing, and deployment of the algorithmic trading program. This encompasses user interfaces, trading algorithm programming, integration of trading platform API, security and risk management measures, and documentation and support provision.

Functional Requirements:

Data Capture and Analysis:
The program must have the ability to get real-time data from at least one financial market, including asset prices, trade volumes, and other relevant indicators. It should analyze this data to identify patterns, trends, and investment opportunities.

Algorithmic Model Development:
Multiple algorithmic models need to be designed and implemented that decide when to buy, sell, or hold financial assets. These models should be adaptable to fit the market better.

Automatic Execution of Trades:
The program should execute buy and sell orders automatically through a trading platform using APIs. It must follow predefined investment strategies.

Intuitive User Interface:
A user interface must be provided, enabling users to monitor the program performance, view executed trades, and access detailed reports.

Non-Functional Requirements:

Low Speed and Latency:
The system must make fast decisions and execute trades within fractions of a second to capitalize on real-time opportunities.

Security and Data Protection:
The program must operate securely, safeguarding financial and personal data involved in trades.

Adaptability to Different Markets:
The program must work effectively in at least one financial asset market, adapting to its specific characteristics.

- Stock markets (Like Apple or Tesla shares for example)
- Forex market (fiat money from different countries)
- Cryptocurrency markets (such as Bitcoin or Ethereum)

With the possibility of expanding the markets and platforms where the program operates in the future.

Reliability and Stability:
The system should be reliable and stable, operating without interruptions and avoiding critical failures.

Understandable Documentation:
Detailed documentation should be provided explaining the program's functionality, implemented models, and how to interpret generated reports.

Key Deliverables:

1. Functional algorithmic trading program.
2. Intuitive user interface.
3. Detailed documentation.
4. Test and analysis reports.
5. Technical user support.

Major Milestones:

1. Completion of program design and specifications.
2. Completion of program development.
3. Completion of unit and integration testing.
4. Security and performance testing conducted and approved.
5. Initial program launch.
6. Ongoing technical support and monitoring.

Risks and Mitigations:

- **Risk:** Volatility in financial markets.
    - **Mitigation:** Implementation of risk management strategies and loss limits.

- **Risk:** Data security breaches.
    - **Mitigation:** Implementation of robust security measures and regular security reviews.

- **Risk:** Changes in financial regulations.
    - **Mitigation:** Staying updated with regulations and adapting the program as needed.

Key Resources:

- Software development team.
- Trading and financial market specialists.
- Access to trading platform APIs.
- Server infrastructure and cloud technology.

Assumptions:

- Necessary permissions and licenses will be obtained for accessing financial markets and executing trades.
- Market data will be available in real-time and reliably through APIs.

Software Architecture:

1. **Microservices Architecture:** The software could adopt a microservices architecture to modularize core functions and facilitate scalability. Each microservice could handle a specific task, such as data capture, analysis, order execution, and monitoring.

2. **Database:** A reliable and scalable database would be used to store market data, transaction histories, user profiles, and strategy configurations.

Key Components:

1. **User Interface (UI):** The software could feature an intuitive user interface that allows users to configure and monitor strategies, view performance reports, and make adjustments as needed.

2. **Data Capture Module:** This module would be responsible for fetching real-time data from multiple market sources, such as stock exchanges and data providers. It would use APIs to access this data.

3. **Analysis Engine:** The analysis engine would process and analyze market data to identify relevant patterns and signals for trading strategies.

4. **Order Execution Module:** This module would connect to trading platform APIs to automatically execute buy and sell orders based on defined strategies.

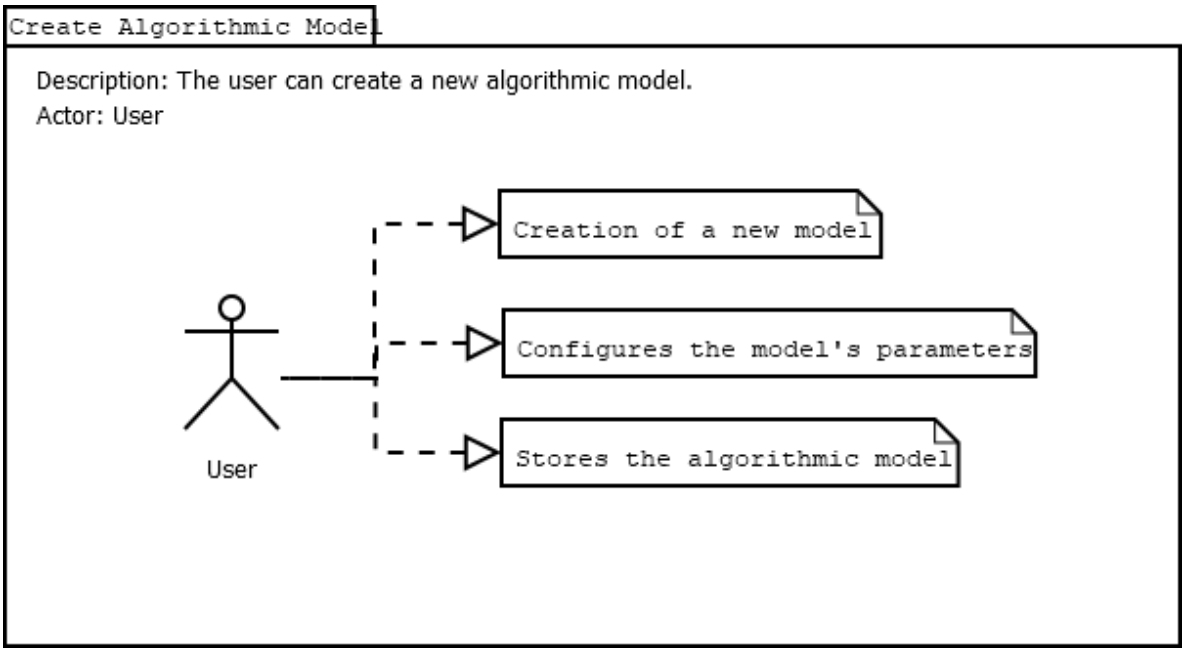
5. **Risk Management:** A critical component would be the risk management module, which evaluates the risk associated with each trade and applies measures like loss limits and stop-loss to protect the investment.

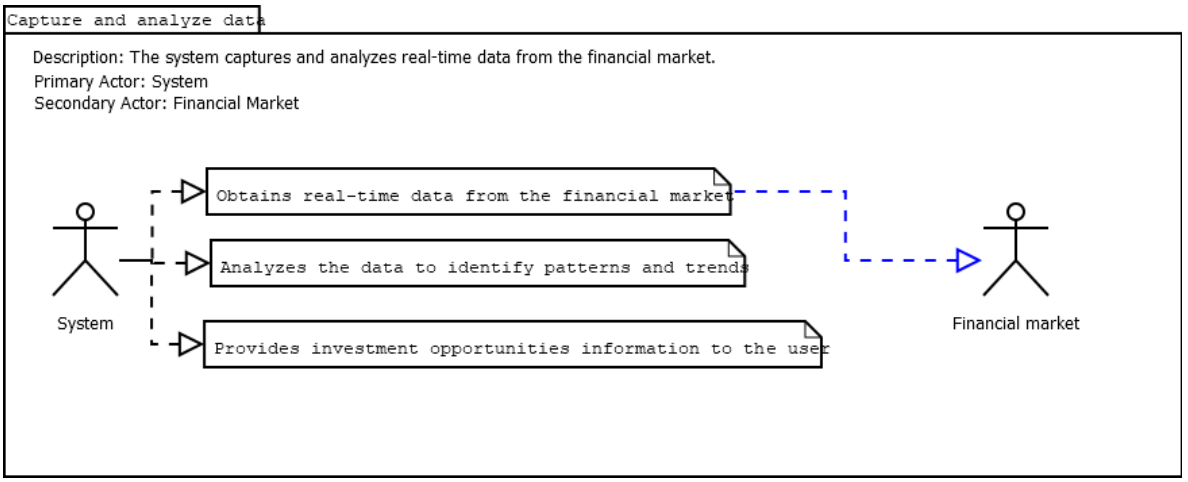
Security and Protection:
Robust security protocols would be implemented to safeguard financial and personal data. This includes user authentication, data encryption, and measures to prevent unauthorized access.

Scalability:
The design would be planned to make the software scalable, capable of handling increased data volume and transactions as it grows.

Documentation:
Comprehensive documentation would be provided, describing the architecture, functionality, and guidelines for strategy development.

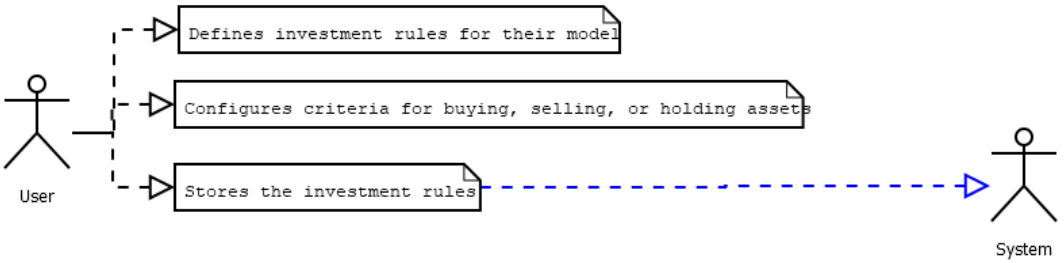
Testing:
Thorough testing, including unit, integration, security, and performance testing, would be conducted to ensure the software operates reliably.


Create Algorithmic Model
Description: The user can create a new algorithmic model.
Actor: User
Creation of a new model
Configures the model's parameters
Stores the algorithmic model
User


Capture and analyze data
Description: The system captures and analyzes real-time data from the financial market.
Primary Actor: System
Secondary Actor: Financial Market
Obtains real-time data from the financial market
Analyzes the data to identify patterns and trends
Provides investment opportunities information to the user
System
Financial market

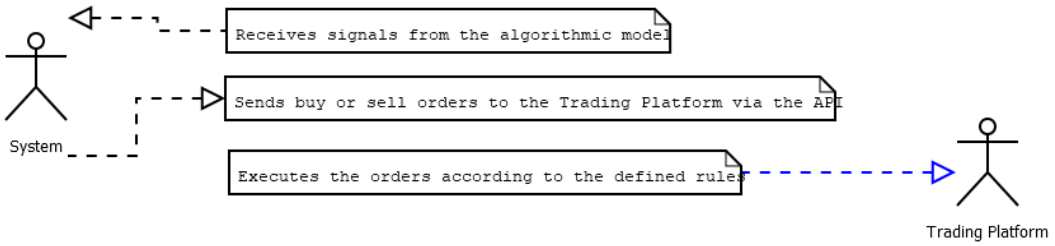
Program investment rules

Description: The user can program investment rules for their algorithmic model.
Actor: User
Secondary Actor: System



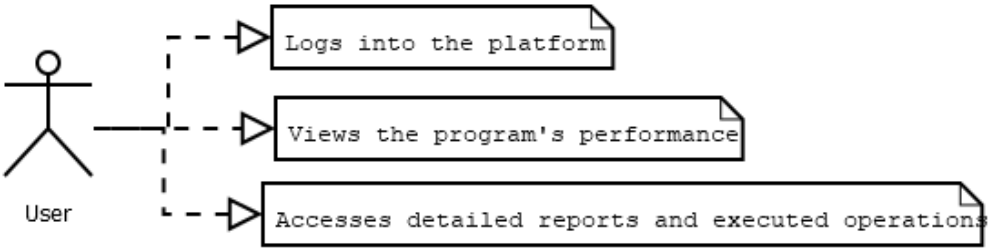
Execute trades automatically

Description: The system executes buy and sell orders automatically using an API.
Primary Actor: System
Secondary Actor: Trading Platform



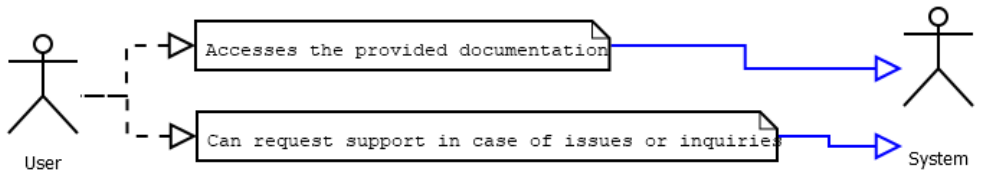
User interface

Description: The user can access an intuitive interface to monitor the program's performance.
Actor: User



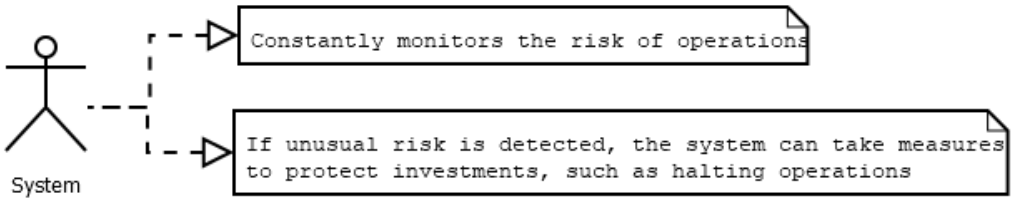
Provide user support

Description: The system offers documentation and support to the user.
Primary Actor: System
Secondary Actor: User



Risk management and investment protection

Description: The system implements risk management measures to protect investments.
Actor: System



Configuration Control

Configuration Control is the process of managing changes to software configuration items (SCI's) in all of their representations. This process involves three basic ingredients: documentation, an organizational body for formally evaluating and approving proposed changes, and a mechanism for implementing and disseminating changes. Configuration Control is a key aspect of Software Configuration Management (SCM), which is a discipline that focuses on managing changes to SCI's in all of their representations.

The following document contains the established policies, the committee board members, and the mechanisms to implement the changes, an example of Configuration Control. [3]

Policies

Decision-Making Policy for MacroSoft Company

Date: 27/11/2023

1. Introduction

This policy sets the guidelines and procedures for decision-making at **MacroSoft & Co**, a software development company. Our board of directors, consisting of the CEO, CFO, HR Director, Project Manager, and a Senior Programmer, is responsible for making key decisions for the success of the company.

2. Committee Board

The committee board at **MacroSoft & Co** is composed of the following members:

- ➔ CEO (Chief Executive Officer)
- ➔ CFO (Chief Financial Officer)
- ➔ HR Director (Human Resources)
- ➔ Project Manager
- ➔ Senior Programmer

The committee board meets regularly to discuss and make strategic and operational decisions in the name of the company.

3. Roles and responsibilities

CEO (Chief Executive Officer):
The CEO is the main leader of the company and is in charge of making big decisions. Their tasks include:

- Setting the vision and business strategy.
- Representing the company to partners, clients, and other people.
- Making sure that policies and goals are put into action.
- Working with other board members to make the company successful.

CFO (Chief Financial Officer):
The CFO is in charge of managing the company's money. Their tasks include:

- Watching over financial planning and accounting.
- Checking if projects and decisions make financial sense.
- Managing budgets and controlling costs.
- Regularly reporting the financial status to the board.

Human Resources Director:
The Human Resources Director takes care of managing the people in the company and making the organization better. Their responsibilities include:

- Coordinating hiring and selection processes.
- Developing and managing training programs.
- Managing relationships at work and solving problems.
- Making sure that human resources policies are followed.

Project Manager:
The Project Manager leads the execution of projects and makes sure they are successful. Their tasks include:

- Making detailed project plans.
- Coordinating teams and assigning tasks.
- Keeping an eye on project progress and managing risks.
- Communicating with the board and project teams.

Senior Programmer:
The Senior Programmer is an expert in software development and leads technical initiatives. Their responsibilities include:

- Designing software architectures.
- Mentoring junior programmers.
- Solving complex technical problems.
- Contributing to high-quality code development.

Junior Programmer:
The Junior Programmer is a software developer in training. Their tasks include:

- Writing and maintaining code following team guidelines.
- Participating in testing and debugging.
- Collaborating with the team to achieve goals.
- Learning and improving technical skills continuously.

4. Importance Scale

Decisions are categorized based on their importance into one of the following levels:

1. **Low:** Decisions with limited impact (less than 1%) on company operations that can be made by appropriate hierarchical levels without board approval.

2. **Medium:** Decisions with a moderate impact (1%-9.999%) on company operations that require review and approval by the board before implementation.

3. **Critical:** Decisions with significant impact (10%-24.999%)on the company and its financial outcomes, requiring unanimous board approval before implementation.

4. **Urgent:** Decisions that must be made immediately, possibly involving emergency situations or decisions with very high (more than 25%) impact on the company. In such cases, the CEO is allowed to make urgent decisions and subsequently communicate them to the board for review.

5. Decision-Making Process

The decision-making process at **MacroSoft & Co** follows these steps:

- Any board member or authorized employee may propose a decision.
- The project manager will be responsible for evaluating and categorizing all changes based on the importance scale.
- "**Low**" and "**Medium**" importance decisions may be made by the appropriate hierarchical level after an internal review.
- "**Critical**" important decisions require unanimous board approval.
- "**Urgent**" decisions can be made by the CEO in emergency situations and are reported to the board afterward.
- If there is a mandatory law that requires changes in requirements, these changes will be above the scale of importance and company policies, which will only apply after complying with all regulations.

6. Evaluation of Initial Budget

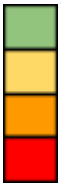
6.1 Budget Measurement Scale:

We will use a scale to measure how close the initial budget estimates are to the actual outcomes in three aspects: time, money, and human resources. The scale will use percentages over time and money to show the difference between initial estimates and final results. To be more accurate regarding human resources, percentages will not be used.

➢ **Time:**

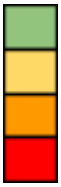
±5%: Excellent accuracy.
±10%: Good accuracy.
±15%: Acceptable accuracy.
>15%: Requires detailed analysis of the reasons for the difference.

➢ **Money:**



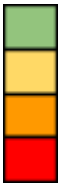
±5%: Excellent financial accuracy.
±10%: Good financial accuracy.
±15%: Acceptable financial accuracy.
>15%: Requires detailed analysis of the reasons for the difference.

➢ **Effort:**



±5%: Excellent accuracy.
±10%: Good accuracy.
±15%: Acceptable accuracy.
>15%: Requires detailed analysis of the reasons for the difference.

➢ **Human Resources[1]:**



0 staff required: Excellent accuracy in estimating human resources.
±1 staff required: Good accuracy in estimating human resources.
±2 staff required: Acceptable accuracy in estimating human resources.
±3 or more staff required: Requires detailed analysis of the reasons for the difference.

[1] It refers to the staff members that were required at the end of the project compared to the members that were originally stipulated.

6.2 Budget Evaluation Table:

The following is a budget evaluation table that will be used in the evaluation process to standardize the results.

	Estimate	Final Result	Difference	Color Indicator	Accuracy
Time			%	⬜	Overestimate / Underestimate
Money			%	⬜	Overestimate / Underestimate
HR			±	⬜	Overestimate / Underestimate
Effort			%	⬜	Overestimate / Underestimate

6.3 Evaluation Process:

- ➔ At the project's completion, we will conduct a detailed review of the initial estimates and the actual outcomes in terms of time, money, effort, and human resources using the budget evaluation table.
- ➔ Each aspect will be assessed using the corresponding measurement scale.
- ➔ We will document the reasons for any significant differences between the estimates and the actual results.
- ➔ The evaluation will be used to improve future estimates and planning processes.

7. Review and Updates

This decision-making policy will be reviewed annually by the board of directors to ensure alignment with changing company objectives and needs.

Approved by:

_____	_____	_____
[**CEO's Signature**]	[**CFO's Signature**]	[**HR Director's Signature**]
[**CEO's Name**]	[**CFO's Name**]	[**HR Director's Name**]

_____	_____
[**Project Manager's Signature**]	[**Senior Programmer's Signature**]
[**Project Manager's Name**]	[**Senior Programmer's Name**]

Date of Approval: 27/11/2023

Auditing

Software auditing is a process that provides traceability between a software life cycle product and the requirements for that product. It also provides a basis for the establishment of the audited life cycle product as a new baseline. The availability of management information, such as a list of all the software structures changed to solve a particular Software Trouble Report or the details on the latest Software document, provides a means for the control function to effectively operate without burdening the development team with cumbersome procedures and administrative paperwork. The purposes of software audit are to provide visibility and traceability of the software life cycle products and to establish control over that process.

The following document establishes a context for the baseline, and 4 changes are requested that alter the visibility and traceability of the software life cycle so they have to be reconsidered Configuration Items. [4]

CONTEXT:

HUMAN RESOURCES

For this project, we are considering hiring the following people:

- ➔ UX/UI Designer (who will be in charge of designing the program and the user interface)
- ➔ Database designer (in charge of the data capture module)
- ➔ Data analyst (in charge of the analysis engine module)
- ➔ Programmer with experience in APIs (in charge of the order execution module)
- ➔ 1 Programmer and 1 finance expert (in charge of the risk management module)
- ➔ 1 Project Manager (in charge of documentation, dealing with the client and in charge of the project in general)

In total 6 employees and the project manager.

MONEY

- The 6 employees will receive the same salary of $1,000 USD monthly.
- The project manager will earn $2,000 USD monthly.
- $500 USD will be used to pay rent, electricity and other expenses.
- A percentage of the final project cost (20%) will be added as a risk budget. This percentage will be included in the final budget, and will serve as a margin for possible mistakes and incidentals that may come up throughout the execution of the project and that are not attributed to changes in specifications by the client.

TIME

The estimated time at the beginning of the project is 6 months.

BUDGET

By having 6 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $8,500 USD per month.

6 x 1,000 = 6,000 + 2,000 = 8000 + 500 = 8,500

The estimated time is 6 months, so $51,000 USD will be required just to cover salaries and expenses.

6 x 8,500 = 51,000

Adding 20% of the risk budget ($12,750 USD), the final total cost will be $63,750 USD.

$\frac{51,000}{80\%}$ ⧅ $\frac{X}{100\%}$ = 63,750

Changes

CR1	The client requests new functionality for trading in the real market and real-time assessment of the trader's decision. It includes an analysis with the timing of 1 minute and provides reasoning on if the trader's decision was good or wrong
CR 2	The client requests to include three new trading algorithms; the first is based on quantum computing, the second in elliptic curves, and the third on convolutional neural networks.
CR3	The client requests the implementation of the most successful investments and the worst investments per day, per week, and month.
CR4	Due to government regulations, the system should send a monthly encrypted signed report with all transactions performed during the day, week, and month.

CR1

The client requests new functionality for trading in the real market and real-time assessing of the trader's decision. It includes an analysis with the timing of 1 minute and provides reasoning on if the trader's decision was good or wrong.

Configuration Identification:

- Code +Add a new function
- Design + Add a new screen with 1 minute analysis
- Documentation + Add the new specifications

Affected modules:

Four of the five modules will be directly affected by client changes.

➔ **User Interface (UI):** The software could feature an intuitive user interface that allows users to configure and monitor strategies, view performance reports (real-time analysis will need to be implemented for 1-minute operations), and make adjustments as needed.

➔ **Data Capture Module:** This module would be responsible for obtaining real-time data from multiple market sources, such as stock exchanges and data providers. It would use APIs to access this data.

➔ **Analysis Engine:** The analysis engine would process and analyze market data to identify relevant patterns and signals for trading strategies. (updates will need to be implemented in periods of 1 minute)

➔ **Order Execution Module:** This module would connect to trading platform APIs to automatically execute buy and sell orders based on defined strategies. (A new type of market order will be implemented that allows the trader to operate manually)

➔ **Risk Management:** A critical component would be the risk management module, which evaluates the risk associated with each trade and applies measures like loss limits and stop-loss to protect the investment. (A new function will be implemented that evaluates if the trader's decision was good or wrong)

Risk assessment:

These changes will imply an increase in time, and consequently in money. Therefore the budget must be updated.

- **Effort.** Time and budget increased **16.66%**.
- **Time.** It will be necessary to increase **one month** of work to the time that was contemplated.
- **Money & Budget.** By having 6 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $8,500 USD per month.

  6 x 1,000 = 6,000 + 2,000 = 8000 + 500 = 8,500

  The new estimated time is 7 months, so **$59,500 USD will be required just to cover salaries and expenses**.

  7 x 8,500 = 59,500

  Adding 20% of the risk budget ($14,875 USD), **the final total cost will be $74,375 USD**.

  $\frac{59,500}{80\%}$ ⇲ $\frac{X}{100\%}$ = 74,375

- **Human resources.** For this particular client request it won't be necessary to hire more qualified workers.

CR2

The client requests to include three new trading algorithms; the first is based on quantum computing, the second in elliptic curves, and the third on convolutional neural networks.

Previous analysis

This particular change request actually includes 3 different requests, so it is necessary to separate them to evaluate each one in particular:

CR2.1

The client requests to include a new trading algorithm based on quantum computing.

Particular analysis: Quantum computing
Quantum computing is a computing paradigm different from classical computing. It is based on the use of qubits, a special combination of ones and zeros. The bits in classical computing can be 1 or 0, but only one state at a time, while the qubit (quantum bit) can have both states simultaneously. This gives rise to new logic gates that make new algorithms possible.

Physical support
The problem of which physical medium would be ideal for quantum computing has not yet been resolved. A series of conditions have been defined that it must fulfill, known as Di Vincenzo's list, and there are currently several candidates.
Google engineers are still working on a quantum processor.

***For these reasons, this particular change request is impossible to make, at least for the moment.**

CR2.2

The client requests to include a new trading algorithm based on elliptic curves.

Particular analysis: Elliptic curves
In 1985, mathematicians Neal Koblitz and Victor Miller independently presented a proposal to use elliptic curves on finite fields in the development of encryption schemes. To understand what elliptic curve cryptography (ECC) is, it is necessary to first review the concept of what elliptic curves are as such.

In cryptography, elliptic curves based on real numbers are not used, as this causes rounding errors in computers. For this reason, elliptic curves defined on finite bodies are used, which can be represented by the following equation:

$$y^2 = x^3 + 10x + 2$$

This elliptic curve cryptography is characterized by having a finite number of points, whose coordinates will only be integers. This characteristic is of utmost importance for this encryption algorithm, since it allows calculations to be made efficiently and without rounding errors.

There are two types of finite fields that can be used in elliptic curve cryptography:

- **Prime bodies**, which have a prime number of elements.
- **Binary bodies**, which have a number of elements that is a prime power of 2.

The way in which another point on the curve is obtained by this method is by adding a point (x, y) many times with itself. So that:

$$Q = P + \ldots + P = n * P$$

Encryption algorithms are usually based on mathematical problems whose solution has not yet been found. This ensures that the feature is not reversible at the hands of a malicious third party.
In the case of RSA encryption, its security is based on the mathematical problem of factoring integers, which means that it is too difficult to factor a very large integer into prime numbers. It is said that this cryptographic system will be able to break with the technology of quantum computers. Therefore, more alternatives have emerged to anticipate this factor.

Elliptic curve cryptography is based on the elliptic logarithm problem, also known as the discrete logarithm problem on elliptic curves. This problem could be defined as follows:

Given a point on the curve Q, obtained by multiplying an integer n and a point P:

$$Q = n * P$$

The problem is that it is very difficult to find n from two known numbers P and Q. To date, a solution has not yet been found, which means that this algorithm continues to be cryptographically secure.

Configuration Identification:

- Code +Add new encryption feature (back end)
- Design (interface or front end will not change)
- Documentation + Add the new specifications for encryption

Affected modules:

One of the five modules will be directly affected by client changes.

- ➔ **User Interface (UI)**
- ➔ **Data Capture Module:** It will be necessary to implement encryption measures on sensitive data in databases.
- ➔ **Analysis Engine**
- ➔ **Order Execution Module**
- ➔ **Risk Management**

Risk assessment:

These changes will imply an increase in time, and consequently in money. Therefore the budget must be updated.

- **Effort.** Time increased **25%** and budget increased **39.7%**.
- **Time.** It will be necessary to increase **one month and a half** of work to the original time.
- **Human resources.** For this particular client request it will be necessary to hire **one** qualified employee expert in cryptography.
- **Money & Budget.** By having 7 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $9,500 USD per month.

  7 x 1,000 = 7,000 + 2,000 = 9000 + 500 = 9,500

  The new estimated time is 7.5 months (30 weeks), so **$71,250 USD will be required just to cover salaries and expenses**.

  7.5 x 9,500 = 71,250

  Adding 20% of the risk budget ($17,812.5 USD), **the final total cost will be $89,062.5 USD**.

  $\frac{71{,}250}{80\%} \nearrow \quad \frac{X}{100\%} = 89{,}062.5$

CR2.3

The client requests to include a new trading algorithm based on convolutional neural networks.

Particular analysis: Convolutional neural networks

A Convolutional Neural Network (CNN) is a type of artificial neural network designed for working with data that comes in a grid-like structure, such as images. CNNs are commonly used in computer vision tasks, pattern recognition, and image processing.

In CNNs, there are specific layers that automatically learn features and patterns from the input data, especially images. Some important parts of a CNN include:

1. **Convolutional Layer:** This layer applies filters (kernels) to parts of the input image to detect features like edges, corners, and textures. Each filter creates a feature map that highlights a specific feature.
2. **Pooling Layer:** Also known as the subsampling layer, it reduces the resolution of feature maps, which reduces the number of parameters and computations in the network. Subsampling helps to retain important features while reducing data redundancy.
3. **Fully Connected Layer:** After the convolutional and pooling layers, the extracted features are flattened and connected to fully connected layers that perform tasks like classification or regression.
4. **Activation Functions:** Functions like ReLU (Rectified Linear Unit) are used in the convolutional layers to introduce non-linearity into the network, improving its ability to learn.

CNNs have proven to be very effective in various applications, including image classification, object detection, image segmentation, and more. Their success is largely due to their ability to learn hierarchical features, allowing them to recognize complex patterns in visual data.

CNNs are widely used in fields such as computer vision, medical image processing, autonomous vehicles, image classification on social media, and many others.

Configuration Identification:

- Code +Add new module
- Design + Add a new screen of the convolutional neural networks module with its new functionalities
- Documentation + Add the new specifications for code and design (backend and frontend)

Affected modules:

All of the five modules will be directly affected by client changes. In addition, it will be necessary to create a new module:

- ➔ **User Interface (UI):** A new screen will be required with all the new functionalities.
- ➔ **Data Capture Module:** A new table will be needed to be created in the database with relations to the new module.
- ➔ **Analysis Engine:** The original analysis will be complemented by the convolutional neural network.
- ➔ **CNN Module:** Will be necessary to create this new module.
- ➔ **Order Execution Module:** This module will be affected by the new module in automatic orders.
- ➔ **Risk Management:** This module is directly related to the analysis engine module, and now it will be necessary to manage risk using the new module as well.

Risk assessment:

These changes will imply an increase in time, and consequently in money. Therefore the budget must be updated.

- **Effort.** Time increased **50%** and budget increased **67.64%**.
- **Time.** It will be necessary to increase **three months** of work to the time that was contemplated in the initial context.
- **Human resources.** For this client request it will be necessary to hire **one** qualified employee expert in Convolutional Neural Network .
- **Money & Budget.** By having 7 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $9,500 USD per month.

  7 x 1,000 = 7,000 + 2,000 = 9000 + 500 = 9,500

  The new estimated time is 9 months, so **$85,500 USD will be required just to cover salaries and expenses.**

  9 x 9,500 = 85,500

  Adding 20% of the risk budget ($21,375 USD), **the final total cost will be $106,875 USD**.

  $\frac{85,500}{80\%} \nearrow \quad \frac{X}{100\%} = 106,875$

CR3

The client requests the implementation of the most successful investments and the worst investments per day, per week, and month.

Configuration Identification:

- Code +Add a new function with the best and worst results per day, week and month
- Design + Add a new screen with the new requirement
- Documentation + Add the new specifications on backend and frontend

Affected modules:
Four of the five modules will be directly affected by client changes.

- ➔ **User Interface (UI):** A new screen or button will be added with the best and worst results allowing the user to choose the timing.
- ➔ **Data Capture Module:** A new relationship will be necessary between the database tables, the execution orders module and the analysis engine module.
- ➔ **Analysis Engine:** It will be necessary to analyze each executed order and compare it with the other orders to be able to locate the best and the worst in different time units.
- ➔ **Order Execution Module:** A new relationship will be necessary with the data capture module.
- ➔ **Risk Management**

Risk assessment:
These changes will imply an increase in time, and consequently in money. Therefore the budget must be updated.

- **Effort.** Time and budget increased **16.66%**.
- **Time.** It will be necessary to increase one month of work to the time that was contemplated.
- **Human resources.** For this client request it won't be necessary to hire more qualified workers.
- **Money & Budget.** By having 6 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $8,500 USD per month.

  6 x 1,000 = 6,000 + 2,000 = 8000 + 500 = 8,500

  The new estimated time is 7 months, so $59,500 USD will be required just to cover salaries and expenses.

  7 x 8,500 = 59,500

  Adding 20% of the risk budget ($14,875 USD), the final total cost will be $74,375 USD.

  $\frac{59,500}{80\%}$ ⇗ $\frac{X}{100\%}$ = 74,375

CR4

Due to government regulations, the system should send a monthly encrypted signed report with all transactions performed during the day, week, and month.

Configuration Identification:

- Code +Add new encrypted monthly report functionality
- Design +Add a new button or screen with the new functionality
- Documentation + Add the specifications with the new government regulations

Affected modules:

Two of the five modules will be directly affected by client changes.

- ➔ **User Interface (UI):** A new button or screen with the new functionality on the UI will be necessary.
- ➔ **Data Capture Module:** It will be necessary to implement encryption measures on sensitive data in databases and all the orders performed will be recorded and encrypted in the monthly report.
- ➔ **Analysis Engine**
- ➔ **Order Execution Module**
- ➔ **Risk Management**

Risk assessment:

These changes will imply an increase in time, and consequently in money. Therefore the budget must be updated.

- **Effort.** Time increased **33.33%** and budget increased **49.01%**.
- **Time.** It will be necessary to increase **two months** of work to the time that was contemplated.
- **Human resources.** For this client request it will be necessary to hire **one** qualified employee expert in cryptography.
- **Money & Budget.** By having 7 employees with a monthly salary of $1,000 USD and a Project Manager with a monthly salary of $2,000 USD and $500 USD for expenses, gives us a total of $9,500 USD per month.

  7 x 1,000 = 7,000 + 2,000 = 9000 + 500 = 9,500

  The new estimated time is 8 months, so **$76,000 USD will be required just to cover salaries and expenses**.

  8 x 9,500 = 76,000

  Adding 20% of the risk budget ($19,000 USD), **the final total cost will be $95,000 USD**.

  $\frac{76{,}000}{80\%}$ ⇲ $\frac{X}{100\%}$ = 95,000

Change Request	Budget	Time	HR	Effort	Risk Expertise	Law Mandatory	First Approval	Ranking	Final Decision
CR1	16.66%	16.66%	0	16.66%	Medium	No	Approved	4	**Rejected**
CR2.1	-	-	-	-	-	No	Rejected	6	**Rejected**
CR2.2	40%	25%	1	32%	Low	No	Approved	2	**Approved**
CR2.3	67.64%	50%	1	58.82%	High	No	Rejected	5	**Rejected**
CR3	16.66%	16.66%	0	16.66%	Low	No	Approved	3	**Approved**
CR4	49.01%	33.33%	1	41.17%	Low	Yes	Approved	1	**Approved**

CR1	The client requests new functionality for trading in the real market and real-time assessment of the trader's decision. It includes an analysis with the timing of 1 minute and provides reasoning on if the trader's decision was good or wrong
CR 2.1	The client requests to include a new trading algorithm based on quantum computing.
CR 2.2	The client requests to include a new trading algorithm based on elliptic curves.
CR 2.3	The client requests to include a new trading algorithm based on convolutional neural networks.
CR3	The client requests the implementation of the most successful investments and the worst investments per day, per week, and month.
CR4	Due to government regulations, the system should send a monthly encrypted signed report with all transactions performed during the day, week, and month.

Decision Criteria

Because CR4 is mandatory by government regulations, although the cost is high in the budget, time and effort, it is something that needs to be implemented as soon as possible.

Therefore, if we do a detailed analysis of the new requirements, we can realize that CR4 will need an encryption algorithm, so the most efficient decision would be to use an elliptic curve algorithm (instead of RSA or more common algorithms) to at the same time perform CR2.2 without spending more resources, so just by making that decision, the cost of CR2.2 could be zero.

At the same time, doing the CR4, which is mandatory, we will have detailed information on all transactions by day, week and month, so doing the CR3 by locating the best and the worst investment in these time periods will be much simpler and less expensive than what appears in the table, because the table only considered making that requirement without taking into account any other.

Conclusions

I could say that if resources are used intelligently, being as efficient as possible, we could carry out 2 more requirements (CR2.2 and CR3) almost without spending resources.

Status Accounting

Software Configuration Status Accounting is a process that involves the administrative tracking and reporting of all software items formally identified and controlled. It also involves the maintenance of records to support software configuration auditing. Data is collected and organized for input to a computer and reports giving the status of entities are generated. Status accounting is generally supported in part by automated processes.

Below is a change history that documents how the project evolved and records all the changes that each document has had.

History of changes

Deadline: September 1st, 2023

At the beginning of the semester, a presentation was held on a software development project that we chose ourselves on the topic we wanted with the condition that it should not be too big or too small. For this project I chose the project titled: Development of an Algorithmic Trading AI with machine learning, which consisted of the development of an AI that would learn to invest in different markets through different platforms through the use of APIs to obtain benefits based on the machine. learning and algorithmic trading. [5]

Deadline: September 7th, 2023

Subsequently, due to the professor's recommendations, some changes were made to the original project, the idea of using AI and machine learning was discarded, the requirement that the software be multiplatform was also discarded, so there would only be one configurable algorithmic trading software. [6]

Deadline: September 15th, 2023

Some use case diagrams were made to complement the baseline, which were attached to the GitHub repository and were added to the baseline. [2]

Deadline: September 22nd, 2023

Later, the first “Change Request” was made, for which it was necessary to establish a context (the people who would be working on the project, the money they would be paid for their services provided, the frequency of payment and a budget based on the specifications of our project). This work was presented in PowerPoint format but 2 documents were made that contain the same information only in different presentations, one in document format, and another in presentation format. [7][8]

Deadline: October 6th, 2023

Later, the first document was made with the company's policies, where the members of the Committee Board were defined, a scale of importance for decision-making and the decision-making process within the company. [9]

Deadline: October 20th, 2023

Subsequently, the initial policy document was corrected with the suggestions given in class. In addition, 3 new specific “Change Requests” were made for each project, for which the following documents were delivered [4][10].

Deadline: November 17th, 2023

On this date, an update was made to the company's policy document, where roles and responsibilities were added, and an evaluation of the results with the initial budget, as well as a measurement scale for the budget, and the process of assessment. [11]

Deadline: November 24th, 2023

On this date, the draft of the master document (this document) was presented. [12]

Deadline: December 1st, 2023

Finally, the final and definitive version of the master document and company policies were presented, where the suggested changes were made in the class taught on November 17th, 2023. [3]

References

[1] E. H. Bersoff, 'Elements of Software Configuration Management', IEEE Transactions on Software Engineering, vol. SE-10, pp. 79–87, 1984.
[2] J. Zepeda. "Baseline 1.1". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Baseline%201.1.pdf (accessed Nov. 26, 2023).
[3] J. Zepeda. "Policies 2.1". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Policies%202.1.pdf (accessed Nov. 26, 2023).
[4] J. Zepeda. "CR4". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/CR4.pdf (accessed Nov. 26, 2023).
[5] J. Zepeda. "Project SCM". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Project%20SCM.pdf (accessed Nov. 26, 2023).
[6] J. Zepeda. "Baseline 1.0". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Baseline%201.0.pdf (accessed Nov. 27, 2023).
[7] J. Zepeda. "Changes". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Changes.pdf (accessed Nov. 27, 2023).
[8] J. Zepeda. "Client Change Request". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Client%20Change%20Request.pdf (accessed Nov. 27, 2023).
[9] J. Zepeda. "Policies 1.0". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Policies%201.0.pdf (accessed Nov. 27, 2023).
[10] J. Zepeda. "Policies 1.1". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Policies%201.1.pdf (accessed Nov. 28, 2023).
[11] J. Zepeda. "Policies 2.0". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Policies%202.0.pdf (accessed Nov. 28, 2023).
[12] J. Zepeda. "Master Document (Draft)". GitHub, 2023. [Online]. https://github.com/JosuIsa94/SCM_Class/blob/main/Master%20Document%20(Draft).pdf (accessed Nov. 28, 2023).