Universidad de Guadalajara

Centro Universitario de los Valles

Software Configuration Management

Project:

System for Event Detection from Text Data

Master: Software Engineering

Third semester

Professor: Omar Ali Zatarain Duran

Efren Plascencia JR.

222977393

Thursday, November 30, 2023

Table of Contents

List of tables

List of figures

Introduction

Software Configuration Management (SCM) is an essential process in software development that involves a series of tasks and responsibilities intended to control and manage changes in a software project. This document focuses on the changes and processes involved in SCM, providing a detailed overview of each of them applied to a project.

The document will address the following key processes:

Configuration identification (CI): this process involves the identification and definition of the elements that constitute the software at a given point in time. This may include source code, design documents, tests, data, among others.

Configuration Control (CC): This process deals with the evaluation, coordination, approval, or disapproval of changes to the software. It is also responsible for the implementation of approved changes.

Configuration Accounting: This process involves recording and reporting the information necessary for the effective management of the software. This includes the status of configuration items and requested changes.

Configuration Audit: This process verifies that changes have been implemented as approved and that all configuration elements are correctly identified and described.

Release management and delivery: This process deals with the preparation and delivery of software to customers, including release planning, release design, and release delivery.

Throughout the document, we describe how each of the process involved in SCM are applied in the project call *System for event detection from text data,* highlighting their importance in effectively managing changes in a software project. The goal is to provide a clear understanding of how changes to the software can be effectively managed.

We hope that this document will be useful guide and valuable resources for those involved in software projects.

Baseline of the project

In the context of Software Configuration Management (SCM), a project baseline is an agreed description of the attributes of a product at a specific point in time, which serves as a basis for defining change.

In this section we show the version 1 of the baseline, this means first attempt the describes the baseline for the system call: System for Event Detection from Text Data.

General description

System with the ability to identify events present in texts written in English, and to determine the possible relationships that exist between the identified events and make a graphic representation of the relationships found.

Where entries to the system will be through Wikipedia links, files in txt format and by writing text in the system.

Key features

The following section shows the key features that are expected to be developed in the system.

Input module

- Users can enter Wikipedia link and the system will fetch the data from the web site.
- Users can upload text files (.txt) where the content will be shown to the user and be able to edit the text in the system.
- User can enter text directly to the system (write and edit).

Preprocessing module

- Creation of a set of tokens which we refer to as a sequence.
- Identification and extraction of stop words in the sequences.
- Identification and extraction of concepts in the sequence.
- Identification and extraction of operators in the sequence.
- Identification of relative and specific dates present in the sequence. (Anchors and intervals)

Processing module

- Identification of relationships between concepts (implicit)
- Identification of relationships between concepts (explicit)

Result module

- Tabular event reporting
- Generation of reports of events graphically (timeline / graph)

Administration module

- Management of users and documents processed for storage.

Requirements

In this section we show the requirements for the system based on the key featured described in the description of the system, we classified by functional and non-functional. Where we will use the acronyms FR to refer to functional requirements and NFR for non-functional requirements.

Functional Requirements

Input module

FR 1: The system will be able to receive text files.

FR 2: The system will be able to show data from the text files to the user.

FR 3: The system will be able to tolerate changes in the data from the data.

FR 4: The system will be able to receive Wikipedia links by the user.

FR 5: The system should be able to fetch the Wikipedia links.

FR 6: The system should be able to get the Wikipedia without HTML and CSS tags (web scrapping)

FR 7: The system will be able to receive data written in the system by the user.

Preprocessing module

FR 1: The system will be able to create a sequence of tokens.

FR 2: The system will be able to identify stop words in the sequence.

FR 3: The system will be able to identify concepts in the sequence.

FR 4: The system will be able to identify operators in the sequence.

FR 5: The system will be able to identify relative and specific dates in the sequence.

Processing module

RF 1: The system will be able to identify relationship between concepts explicit with anchors or intervals.

RF 2: The system will be able to identify relationships between concepts implicit with anchors or intervals.

Result module

FR 1: The system will be able to generate tables representing the concepts and the relationships between concepts.

FR 2: The system will be able to generate tables representing the isolated concepts.

FR 3: The system will be able to generate reports of concepts with a graph.

FR 4: The system will be able to generate reports of concepts with timelines.

Administration module

The following requirements are specified for the web version.

FR1: User must be able to create account by providing their personal information such as full name, email, username, and password.

FR2: User should be able to edit their personal information and contact information.

FR3: User should be able to log in using their registered credentials.

FR4: Users should be able to recover forgotten password through a reset process.

FR5: The system must validate the uniqueness of email to avoid duplicate registration.

FR6: The system must verify the authenticity of the email during the registration process.

FR7: The system should be able to save files and the result of the analysis performed if the user wishes.

No-Functional Requirements

Availability

NFR 1: The system must be available most of the time for the needs of the final user.

Devices:

NFR 1: The program must work consistently across different operating systems (Windows or Linux).

NFR 2: The program must work consistently across different internet browsers.

Design

NFR 1: Responsive user interface for use on desktop and web applications.

NFR 2: The system must allow informative messages about errors.

Maintainability

NFR 1: The code source should be well structured in some pattern of software development to facilitate future update and changes.

NFR 2: Documentation on the code and architecture to facilitate maintenance by future developers.

Performance

NFR 1: Loding time should be adequate to avoid user frustration.

NFR 2: Processing time for texts needs to be the more efficient possible.

Security and Privacy

NFR 1: The system must follow regulations and privacy standards.

NFR 2: The system access must have some security mechanisms.

Usability

NFR 1: User interface must be intuitive and easy to navigate for the final user.

Requirements Analysis

In this section, we analysis the requirements and we present a use case diagram for the general representation of main actors that intervene in the system with the intention demonstrating the purpose for the system and the understanding for the development team and the stakeholders.

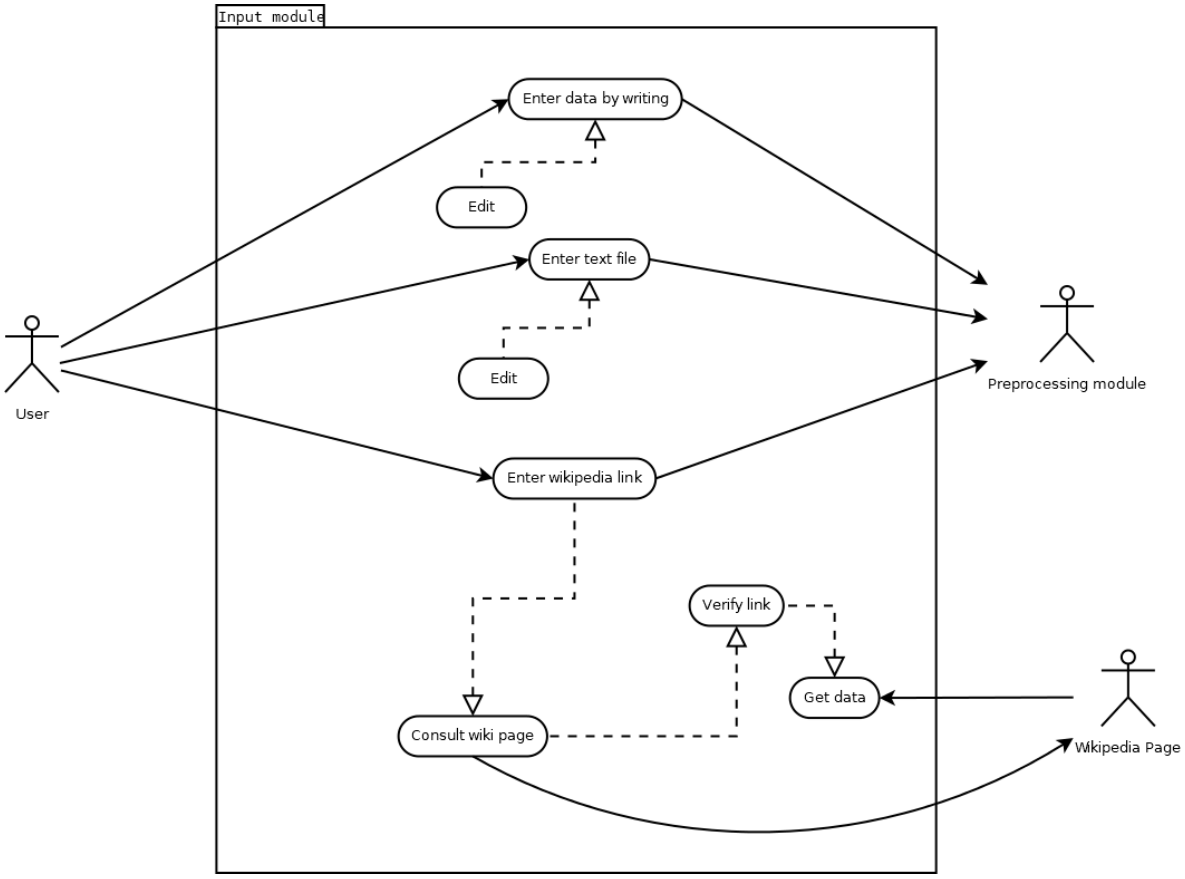

Figure 1 Use case diagram input module.

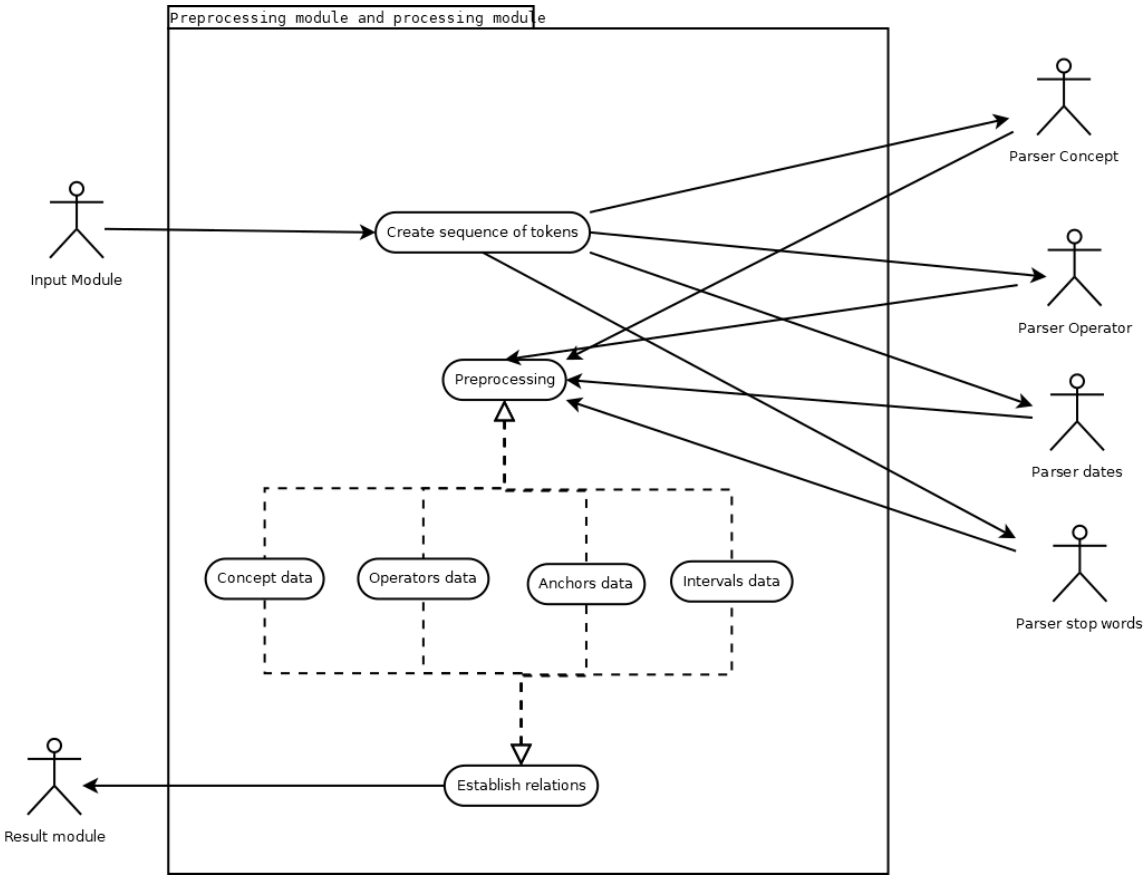

Figure 2 Use case diagram preprocessing and processing module.

Design System

In this section, we describe the flow of the system figure 3 where it seeks to show the different stages involved in the system.

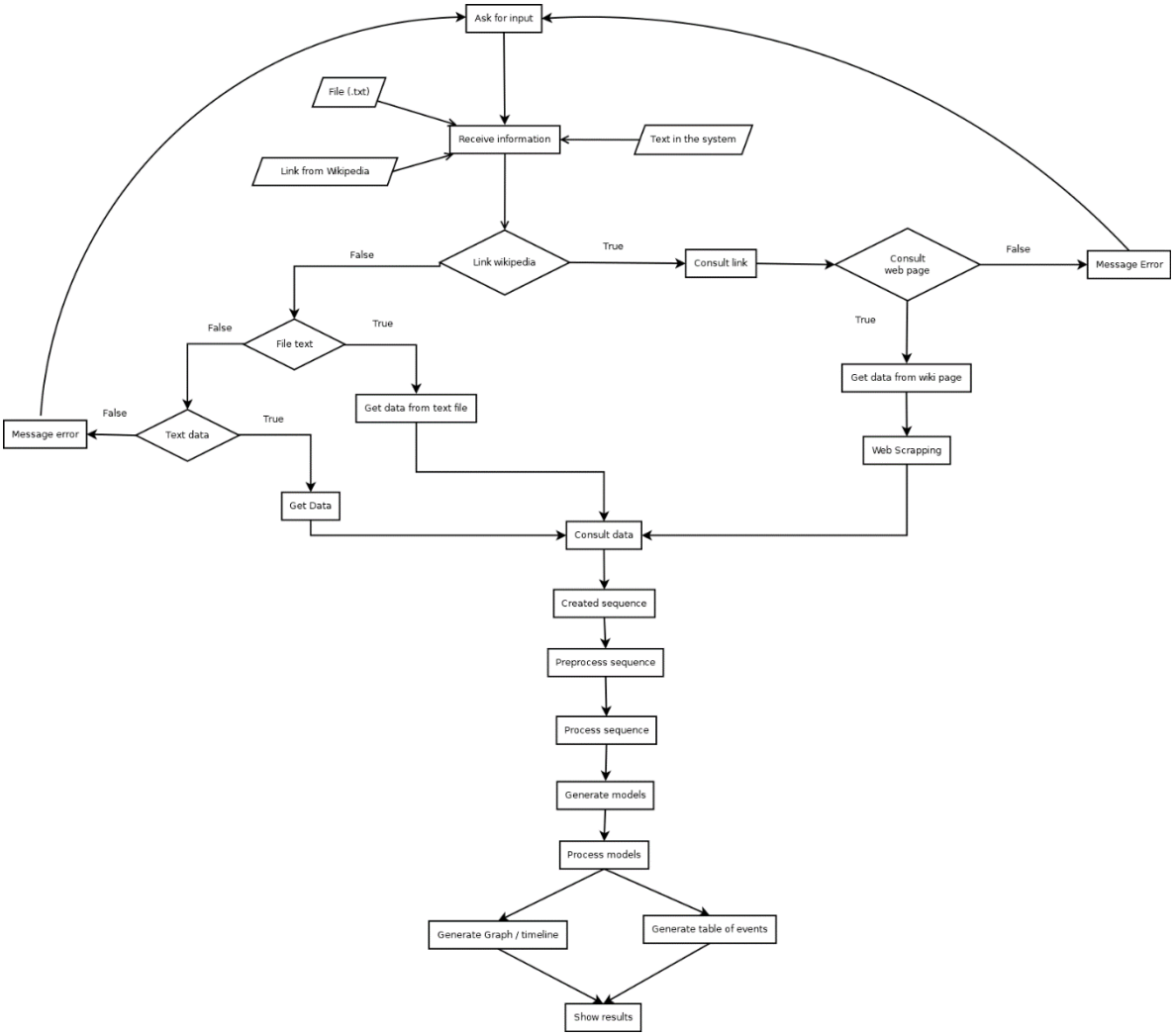

Figure 3 Flowchart of the system.

Committee board

In this section we describe the software development committee board that is an essential for the SCM framework. This board comprises a group of key stakeholders, including project managers, lead developers, and quality assurance leads, among others, who are responsible for overseeing and guiding the software development process.

The board's primary role is to ensure that the software development process aligns with the organization's objectives and standards. They are involved in critical decision-making process, such as approving changes to the software, managing risks, and resolving conflicts that may arise during the development process.

General description committee board functions within the project

For this project, we consider the following roles:

- **Project Manager**: Person in charge of the project direction and management.
- **Finance representative / advisor**: Person in charge of financial recommendation.
- **Chief programmer**: Senior programmer of the team.
- **Stakeholder**: Person with the interest in the project.
- **Legal adviser**: Person who provides legal advice.

We are considering the following team for the development of the system.

6 developers (2 seniors, 2 semi – seniors, 2 junior)

According to the baseline and the description of the initial team, we will describe the aspects of time and budget for development of the system:

Time: Is estimated that it will take one year to create the system and put it into operation.

Budget: Considering that there is an estimated time of one year for the development of the system, the following aspects to consider are broken down to generate the initial budget based on the first version of the baseline.

Monthly payments:

Personnel

- Project Manager: $7,275.00
- Finance advisor: $6,659.00
- Legal adviser: $8,554.00
- Senior developer: $10,076.00
- Semi senior developer: $8,664.00
- Junior developer: $5,650.00

Office and Services

- Office: $3,148
- Services (internet, electricity, water): $309.00

Total for month: $50,335.00

Personnel: $46,878.00

Office and Services: $3,457.00

Total for the entire project: $604,020.00

Personnel per year: $46,876.00 * 12 = $562,536.00

Office and Services per year: $3,457.00 * 12 = $41,484.00

Based on this initial budget, we will consider for the decisions making for the future change requests.

Change management policy.

In this section we describe the change management policy that is a critical component of SCM. It outlines the procedures for making changes and managing the effects of these changes in a software project. The policy is designed to handle changes in a controlled manner, ensuring that no unnecessary updates are made, while necessary changes are not overlooked. It involves identifying, documenting, approving, or rejecting changes to the project, with the aim of controlling costs, reducing unnecessary risks, and providing a basis for project evaluation and accountability.

The change management policy includes guidelines on how to submit change requests, how these requests are approved and implemented, and how changes are tracked and reported. It also defines the roles and responsibilities of each member of the software development team, ensuring that everyone understands their part in the change process.

General Criteria

For making the decision to accept or dismiss the change request in the software project we consider:

- The project manager must be clear about the objectives and motivation of the project.
- Compilation of initial stakeholder requirements: It involves:
  1. Understanding and documenting what the stakeholders expect from the software from the beginning of the project.
  2. Changes that may be generated during development must be document for future analysis.
- There are laws or regulations in the country where the system will be implemented.
- The financial impact of the change is acceptable (15 or 25 % of the entire project) considering more than one change request. If only consider one change request will be 5 to 10 % of the entire project.
- The risks estimated without starting development, what level of risk do they have?

  High = 15 to 25 % of the project budget
  Medium = 10 to 15 % of the project budget
  Low = 5 to 10 % of the project budget
- Is there adequate equipment for the requested change?
- There are trained personnel for the requested change.
- In terms of time, an agreement will be reached with the clients for the delivery date, it is considered as a rule that the maximum time allowed will be 30 percent of the complete project.
- In terms of human resources, we must consider the budget in terms of percent we can assigned to make the decision to hire more staff; if it consumes more than you have, you will choose to discard the change request or not hiring more people.

Change request.

In this section we talk about the change request that is a formal proposal for an alteration to be made in a software project and is a fundamental aspect of SCM. It is typically raised by any stakeholder involved in the project, such as a team member, a project manager, or even a client, when they identify a need for a change in the product.

The change request includes detailed information about the proposed change, including the reason, the benefits it will bring, any potential risks, and the resources required to implement it.

In this section we show the desired modification to the software systema and can range from a simple request for a minor adjustment to a significant architecture alteration. Once a change request is submitted, it undergoes a thorough review process by the change control board or the Software Development Committee Board.

This board evaluates the impact of the proposed change on the project's scope, cost, schedule, and quality. If approved, the change is then implemented, tested, and verified before being incorporated into the software product.

First change request

The client wants a module that detect geographic relations within the text and generate a spatial-temporal graph that describes the following directions: north, south, east, west, above, below, left, right inside, and outside.

Configuration Identification

Using the concept of configuration identification, we identify the following concepts applicable to the change request:

Code

Functionalities to be implemented:

1. **Geographic Relation Detection:** The system should be able to parse the text and identify geographic relationships. This includes directions such as north, south, east, west, above, below, left, right, inside, and outside.
2. **Spatial-Temporal Graph Generation:** Post detection (point 1), the system should generate a spatial-temporal graph that visually represents these geographic relationships.

This new functionality will enhance the system's ability to understand and visualize geographic relations in each text, providing a more interactive and intuitive user experience.

Design

Implementation of this feature may depend on the existing system architecture and may require additional libraries or tools for text parsing and graph generation.

Documentation

All modules to be developed must be documents complying with the quality characteristics as required by ISO 25000.

Risk assessment

Budget

The additional budget for this change request is:

$63,330.00

Human Resource

We have the team already only needs to update details of the change.

Time

The additional time required for this change is:

Two months

Second change request

The client wants a module that differentiates semantics according to the context: “I look forward to hearing from you” and “We moved forward towards the tower.” One speaks about the future and the other describes a direction.

Configuration Identification

Using the concept of configuration identification, we identify the following concepts applicable to the change request:

Code

Functionality to be implemented:

1. **Contextual Semantics Differentiation:** The module should be able to parse the text and identify the semantics based on the context. It should be able to differentiate between similar phrases used in different contexts.

This new functionality will enhance the system’s ability to understand and interpret text more accurately, providing a more nuanced and context-aware user experience.

Design

Implementation of this feature may depend on the existing system architecture and may require additional libraries or tools for text parsing and semantics analysis.

Documentation

All modules to be developed must be documents complying with the quality characteristics as required by ISO 25000.

Risk assessment

Budget

The additional budget for this change request is:

$47,497.50

Human Resource

We have the team already only needs to update details of the change.

Time

The additional time required for this change is:

One and half month.

Third change request

The client requires a module that perceives the phonetics from a person and translates it into text.

Configuration Identification

Using the concept of configuration identification, we identify the following concepts applicable to the change request:

Code

Functionalities to be implemented:

1. **Phonetic Perception:** The module should be able to perceive and interpret the phonetics from a person's speech.
2. **Speech-to- text translation:** the perceived phonetics should be translated into text.

This new functionality will enhance the system's ability to understand and transcribe spoken language, providing a more accessible and user-friendly experience.

Design

Implementation of this feature may depend on the existing system architecture and may require additional libraries or tools for speech recognition and speech-to-text.

Documentation

All modules to be developed must be documents complying with the quality characteristics as required by ISO 25000.

Risk assessment

Budget

The additional budget for this change request is:

$220,000.00

Considering only the new team members.

Human Resource

We need to hire new team members:

2 software engineers:

1 Sound engineer:

Time

The timeline for this change request like this can range from a few months for a simple implementation to over a year for more accurate and precise recognition. We are considering eight months.

Fourth change requests

The client desires a new Screen for interacting with the system. It requires web design for people with disabilities.

Configuration Identification

Using the concept of configuration identification, we identify the following concepts applicable to the change request:

Code

Functionalities to be implemented:

1. **Accessible Web Design:** The new screen should follow the principles of accessible web design. This includes features like text-to-speech, high contrast mode, keyboard navigation, and more.
2. **User Interaction:** The screen should provide an intuitive and user-friendly interface for interacting with the system.

This new functionality will enhance the system's usability and accessibility, making it more inclusive and user-friendly for people with disabilities.

Design

Implementation of this feature may depend on the existing system architecture and may require additional libraries or tools for accessible web design.

Documentation

All modules to be developed must be documents complying with the quality characteristics as required by ISO 25000.

Risk assessment

Budget

The additional budget for this change request is:

$21,500.00

Human Resource

We need to hire new team members:

2 Web designers:

1 Graphic designer:

Time

The additional time required for this change is:

One month.

SWOT

This part of the current section talks about the change analysis made for each of the four-change request. Where it is intended to demonstrate through swot tables the analyzes carried out on each of the requested changes.

Table 1 SWOT Table for First Change Request

SWOT: Change Request # 1	
Strength	**Weaknesses**
Advanced Functionality: The ability to detect geographic relations within the text and generate a spatial-temporal graph is an advanced functionality that can significantly improve the system's capabilities. User Experience: It can enhance user experience by providing more context-aware and visually appealing responses.	Technical Complexity: Developing a module that accurately detects geographic relations and generates a spatial-temporal graph can be technically challenging. Data Availability: The accuracy and effectiveness of the functionality heavily depend on the availability and quality of the geographic data.
Opportunities	**Threats**
Innovation: This could lead to the development of new features or products based on advanced geographic data analysis. Market Differentiation: By prioritizing geographic relations detection, the client could differentiate themselves in the market.	Competition: There are already established players in the market offering similar functionalities. User Expectations: High user expectations for accurate geographic relations detection could lead to dissatisfaction if not met.

Table 2 SWOT Table for Second Change Request

SWOT: Change Request # 2	
Strength	**Weaknesses**
Contextual Understanding: A module that differentiates semantics according to the context can significantly improve the system's understanding and interpretation of user inputs. User Experience: It can enhance user experience by providing more accurate and context-aware responses.	Technical Complexity: Developing a module that accurately understands and differentiates semantics based on context can be technically challenging. Ambiguity: Language is inherently ambiguous, and context can be subjective, which might lead to inaccuracies in interpretation.
Opportunities	**Threats**
Innovation: This could lead to the development of new features or products based on advanced natural language understanding. Market Differentiation: By prioritizing contextual understanding, the client could differentiate themselves in the market.	Competition: There are already established players in the market offering similar functionalities. User Expectations: High user expectations for accurate contextual understanding could lead to dissatisfaction if not met.

Table 3 SWOT Table for Thrid Change Request

SWOT: Change Request # 3	
Strength	**Weaknesses**
User Interaction: A module that translates phonetics into text can significantly improve user interaction and experience. Versatility: Such a module can be useful in various scenarios, such as transcription services, voice assistants, etc.	Technical Complexity: Developing a module that accurately perceives phonetics and translates it into text can be technically challenging. Resource Intensive: It may require significant computational resources, which could impact the performance of the overall system.
Opportunities	**Threats**
Market Demand: With the increasing use of voice technology, there is a growing market demand for such modules. Innovation: This could lead to the development of new features or products based on voice recognition technology.	Competition: There are already established players in the market offering similar functionalities. Accuracy: Ensuring high accuracy in different languages and accents can be challenging.

Table 4 SWOT Table for Fourth Change Request

SWOT: Change Request # 4	
Strength	**Weaknesses**
Accessibility: Designing for people with disabilities can make the system more accessible and user-friendly. Competitive Advantage: Few systems prioritize accessibility, so this could provide a competitive edge. User Base Expansion: Accessibility could attract a wider user base, including those with disabilities.	Specialized Knowledge Required: The team may lack the necessary skills or expertise to develop an accessible web design. Increased Development Time: Ensuring accessibility could extend the development timeline.
Opportunities	**Threats**
Market Differentiation: By prioritizing accessibility, the client could differentiate themselves in the market. Regulatory Compliance: The client could potentially meet more regulatory standards by ensuring accessibility.	Increased Costs: The cost of developing the new screen and ensuring it is accessible could exceed the initial budget. Project Delays: The project could take longer than expected due to the need for extensive testing and adjustments to ensure accessibility.

Decision making

In this section we describe what change request where accept based on the policy we describe previously in this document.

Table 5 Decision Making Guideline

Decision Making Guideline	
Budget	25 % (Considering more than one change request)
Time	30 % (Considering more than one change request)
Human Resources	Check with the budget.
Work Effort	25 %

Table 6 Budget Analysis of Change Request

Table for budget analysis of requested changes						
Change Request	**Number of People**	**New Personnel**	**Months Work**	**Salary for Change request**	**Total**	**Percentage Of main budget**
CR1	0	-	2	$63,330.00	$63,330.00	10.48%
CR2	0	-	1.5	$47,497.50	$47,497.50	7.86%
CR3	3	Software Engineer Sound Engineer	8	$152,000.00 $68,000.00	$220,000.00	36.42%
CR4	3	Web Designer Graphic Designer	1	$15,000.00 $6,500.00	$21,500.00	3.56%

Note: If a - is found in the table, it must be understood that it will be the initial team of the project.

Equation applied to determine the percentage of the budget:

(Cost of salary for change request/ initial budget) * 100

Table 7 Time Analysis of Change Request

Table for time analysis of requested changes			
Change Request	**Original Time (Months)**	**Additional Time (Months)**	**Percentage**
CR1	12	2	16.67%
CR2	12	1.5	12.5%
CR3	12	8	66.67%
CR4	12	1	8.33%

Equation applied to determine the percentage of the time:

(Original time/ additional time) * 100

Table 8 Analysis of aspects for acceptance of changes

Table with the analysis concentrate									
Change Request	**Budget**	**Time**	**Human Resource**	**Effort**	**Potential Risk**	**Law Mandatory**	**First Approval**	**Ranking**	**Final Decision**
CR1	Accepted	Accepted	Accepted	13.58	Medium	No	Accepted	3	Accepted
CR2	Accepted	Accepted	Accepted	10.18	Medium	No	Accepted	2	Accepted
CR3	Rejected	Rejected	Rejected	51.55	High	No	Rejected	4	Rejected
CR4	Accepted	Accepted	Accepted	5.95	Low	No	Accepted	1	Accepted

Justification for accepting change requests.

CR1 and **CR2**: They are considered medium risk and considering the analysis carried out in the SWOT allows us to consider them as good changes and if changes 1 and 2 are considered at the same time the impact on the initial budget could be reduced.

CR3: It is considered a very dangerous change for the project given the analyzes carried out, since it was practically considered as a subsystem within the system. Which could lead to many errors and thus a failed system.

CR4: It is considered low risk which benefits the project in terms of the requested changes and will improve the user experience. Even so, the change should be taken carefully from what is mentioned in the SWOT.

Changes in Baselines

In the following sections, we will delve into the modifications made to the established baseline. These changes, whether they are minor tweaks or major overhauls, play a crucial role in enhancing the performance, efficiency, and overall functionality of the system. Each section will provide a detailed analysis of the specific change implemented, and their impact on the system's performance. By understanding these changes, we can gain insight into the evolution of the system and the continuous efforts for improvement.

The changes made in the different baselines are made in accordance with the order of acceptance of the requested changes. Therefore, the first change will be CR 4, then CR 2 and finally CR 1. In order not to repeat all the functionalities mentioned above, only the changes in the functional and non-functional requirements that will be added from the requested changes will be listed in each new baseline.

Baseline version 1.1

For the CR4 we considered the next break down in terms of functional and non-functional requirements for this new screen:

Functional Requirements

User Authentication: The screen should provide a mechanism for user authentication before allowing interaction with the system.

Accessibility Features: The screen should be designed to be accessible for people with disabilities. This could include features such as:

- text-to-speech
- high-contrast themes
- keyboard navigation

Data Interaction: The screen should allow users to view, add, edit, and delete data in the system.

Error handling: The screen should include help text-to-speech with clear error messages and guidance to users when something goes wrong.

Non-Functional Requirements

Performance: The screen should load quickly and respond to user input without delay.

Usability: The screen should be intuitive and easy to use, even for users who are not tech-savy.

Accessibility: The screen should be accessible to people with disabilities. This includes complying with standards like the web content accessibility guidelines (WCAG).

Security: The screen should protect user data and prevent unauthorized access to the system.

Scalability: The screen should function correctly even as the number of users or amount of data in the system increases.

Baseline version 1.2

For the CR2 we considered the next break down in terms of functional and non-functional requirements for this semantic differentiation module:

Functional Requirements

Context Understanding: The module should be able to understand the context of a sentence to differentiate the semantics of words. For example, it should understand that “forward” in “I look forward to hearing from you” refers to anticipation (future), while in “We moved forward towards the tower” it refers to a direction.

Semantic differentiation: The module should be able to differentiate between different meanings of the same word base on context. It should correctly identify whether “forward” is being used to refer to time or direction.

Input Handling: The module should accept text input and return the differentiated semantics as output.

Non-functional Requirements

Accuracy: The module should accurately differentiate the semantics of words in a variety of contexts.

Efficiency: The module should process input and return output quickly, even for large amounts of text.

Scalability: The module should maintain its performance as the amount of data it needs to process increases.

Usability: It should be easy to use and integrate with the rest of the system.

Reliability: The module should be able to operate continuously without failure.

Error Handling: The module should provide clear error messages if it fails to differentiate the semantics of a sentence.

Baseline version 1.3

For the CR1 we considered the next break down in terms of functional and non-functional requirements for this geographic relations detection module:

Functional Requirements

Geographic Relation Detection: The module should be able to detect geographic relationships within the text. This includes recognizing and understanding directions such as north, south, east, west, above, below, left, right, inside, and outside.

Spatial-Temporal Graph Generation: The module should generate a spatial-temporal graph based on the detected geographic relations. This graph should accurately represent the relations and directions identified in the text.

Input Handling: The module should accept text input and return the generated graph as output.

Non-functional Requirements

Accuracy: The module should accurately detect geographic relations and generate the corresponding spatial-temporal graph.

Efficiency: The module should process input and return output quickly, even for large amounts of text.

Scalability: The module should maintain its performance as the amount of data it needs to process increases.

Usability: It should be easy to use an integrate with the rest of the system.

Reliability: The module should be able to operate continuously without failure.

Error Handling: The module should provide clear error messages if it falls to detect geographic relations or generate the graph.

Status Accounting

In this section we talk about status accounting of SCM. It involves the process of recording and reporting the necessary information on the status of the development process. This includes tracking all elements of the software configuration, documenting their state during the different stages of the software development lifecycle, and maintaining a history of the changes that have been made. It provides visibility into the progress of the project, allowing stakeholders to make informed decisions based on the current state of the software.

Status Accounting helps in understanding the composition of different versions and builds of the software. It also aids in tracing the changes made to each component over time. This information Is vital for managing the software effectively, diagnosing issues, and planning future actions.

The criterial to evaluate in the aspect of documentation is shown in the following table.

Table 9 Documented Process Criteria

Documented Process Criteria	
Check Mark	**Task description**
	Record the current approved configuration documentation and configuration identifiers associated with each configuration item (CI)
	Record and report implementation status of authorized changes.
	Provide the traceability of all changes from the original released configuration documentation of each configuration item (CI) affected.
	Record all document and software that has been delivered to the stakeholder or the main repository.

The criterial to evaluate in the aspect of code is shown in the following table.

Table 10 Code Process Criteria

Code Process Criteria	
Check Mark	**Task description**
	Each commit has a unique identifier, a timestamp, the person who made the change, and a message describing the change.
	All function develop has a backup.

The criterial to evaluate in the aspect of time is shown in the following table.

Table 11 Criteria for Time

Criteria table for time	
Evaluated in days or month (depends on the project or the change request)	**Aspect to consider:**
	Amount of time for the approve a change.
	Amount of time for the conception of a feature to its implementation.
	Amount of time that developers spend working on the project.
	Amount of time from the start to the completion of the task.

The following table shows the guide to evaluate aspects of interest based on configuration items.

Table 12 Guideline for Status Accounting

Guideline for status accounting		
Configuration Item (CI) Identification		
Score:	Score (0 – 25) Criteria: 25: The system can perfectly identify and track all Cis 15: Can do so for most Cis 0: It fails to identify most CiI.	Description Check if the system can accurately identify and track all CIs.
Status Recording		
Score	Score (0 – 25) 25: The system can perfectly record the status of all Cis 15: Can do so for most Cis. 0: It fails to identify most Cis.	Description: Evaluate the system's ability to always record the status of CIs.
Status Reporting		
Score	Score (0 – 25) 25: The system can perfectly record the status of all Cis 15: Can do so for most Cis. 0: It fails to identify most Cis.	Description: Asses the system's ability to generate accurate and timely reports on the status of Cis.
Audit Trail		
Score	Score (0 – 25) 25: The system can perfectly record the status of all Cis 15: Can do so for most Cis. 0: It fails to identify most Cis.	Description: Check whether the system maintains a complete and accurate audit trail of all changes to CIs.
Total Score:	Evaluated by:	

Auditing

In the upcoming section, we will be focusing on the critical aspect of Software Configuration Management (SCM). SCM Auditing is a systematic process of evaluating and ensuring the integrity and effectiveness of software configuration management activities. It involves the examination of the SCM processes, tools, and personnel to verify that they comply with the defined standards and best practices. This section will provide are guidelines.

Table 13 Guideline for Auditing

Guideline for Auditing		
Configuration Identification		
Score:	Score (0 – 25) Criteria: 25: The system can perfectly identify and document all Cis. 15: Can do so for most Cis. 0: It fails to identify most CIs.	Description Check if the system can identify and document the characteristics of a configuration item (CI).
Configuration Control		
Score	Score (0 – 25) 25: The system can perfectly control all changes. 15: Can do so for most Cis. 0: It fails to identify most Cis.	Description: Evaluate the system's ability to evaluate, approve or disapprove, and implement changes to CIs.
Configuration Status Accounting		
Score	Score (0 – 25) 25: The system can perfectly record and report all CI information 15: Can do so for most Cis. 0: It fails to identify most Cis.	Description: Assess the system's ability to record and report CI information and status.
Configuration Audits		
Score	Score (0 – 25) 25: The system can perfectly audit all Cis. 15: can audit most CIs 0: it fails to audit most CIs.	Description: Check if the system can audit the CIs to verify conformance to specifications and physical presence.
Total Score:	Evaluated by:	

Reference

[1] E. H. Bersoff, "Elements of Software Configuration Management," in IEEE Transactions on Software Engineering, vol. SE-10, no. 1, pp. 79-87, Jan. 1984, doi: 10.1109/TSE.1984.5010202.

[2] ANSI/EIA-649-1998, “National Consensus Standard for Configuration Management,” ANSI/EIA, 1998.

[3] EIA-649-A, “National Consensus Standard for Configuration Management,” EIA, 2004.

[4] SAE EIA-649-C, “Global Consensus Configuration Management Standard,” SAE, 2019.

[5] ISO 10007, “Quality management systems – Guidelines for configuration management,” ISO, latest revision year.

[6] Plascencia JR., E. GitHub. Retrieved November 20, 2023, from https://github.com/Efr3nJR/SCM_Class2023B