ICES4HU
Configuration and Change Management Report

1 Introduction

ICES4HU is such a system that needs Configuration and Change Management because of the amount of people working on this project and the change in requirements over time by stakeholder's demand and feedback.

Configuration and Change Management is a process that involves identifying, organizing, and controlling the changes made to a system or product during its lifecycle. This procedure aids in making sure that all changes are efficiently planned, monitored, and managed, and that any potential risks or repercussions are recognized and reduced. Maintaining a record of the configuration of the system or product as well as all pertinent documents, such as design specifications, test plans, and user manuals, is another aspect of configuration and change management. This makes it possible to guarantee that all interested parties have access to the most recent information and that the system or product can be efficiently maintained and supported throughout time.

Because change occurs frequently in the software development industry, change and configuration management are essential. There are bugs that need to be fixed. As changes are made to the program, a new iteration of the system is created. Therefore, it is possible to think of the majority of systems as a collection of versions that need to be managed and kept up to date.

Thus, Software Configuration and Change Management is a software engineering technique that systematically manages, arranges, and controls changes to documents, codes, and other entities across the Software Development Life Cycle. Enhancing output while making as few errors as feasible is the key goal. SCM is a subset of configuration management, a multidisciplinary field that can identify who made which changes.

2 Purpose

It is essential for team projects where many developers are concurrently working on a software system. Although these developers occasionally collaborate locally, development teams are becoming more distributed. By using a configuration management system, teams can access information about a system that is being developed and can operate independently of one another.

1. **Control changes:** offers an organized approach for handling software changes, including to code, documentation, and other artifacts. Controlling changes reduces the possibility of mistakes and discrepancies that may occur when several individuals are working on the same software.

2. **Ensure consistency:** by keeping track of all changes made to the software, including code changes, documentation updates, and other modifications, helps to assure consistency across all versions of the software.

3. **Improve collaboration:** provides a common area for organizing and storing software artifacts, which encourages team collaboration. This enables team members to follow changes made by other team members and get the most recent software and documentation.

4. **Increase efficiency:** may contribute to increased productivity by automating typical operations like software development and testing. This frees up team members to work on more complex projects like creating and implementing new features.

3 Configuration and Change Management Specifications

Configuration and Change Management is a critical aspect of software development that ensures the proper management, control, and tracking of software artifacts and changes made throughout the software development lifecycle. Software artifacts can be changed in a systematic way with the help of configuration and change management, which lowers the chance of errors, inconsistencies, and conflicts that may result from uncontrolled modifications.

We will go over the specifications and procedures for managing configuration and change in the Instructor and Course Evaluation System for Hacettepe University (ICES4HU) in this Configuration and Change Management Report.

The processes, guidelines, and tools necessary for efficient configuration and change management in the ICES4HU system will be outlined in detail for each specification. The advantages of efficient configuration and change management will also be emphasized in the study, including better control, visibility, and quality verification across the software development lifecycle.

The ICES4HU system will benefit from an organized and monitored method for configuration and change management by following the guidelines provided in this report. This will lower the probability of errors and inconsistencies and ensure that the system satisfies the requirements of Hacettepe University's students, instructors, and university committees.

Configuration Management is the process of identifying, organizing, and controlling changes implemented in software artifacts throughout the lifecycle of software development. Version control, configuration item identification, and upholding the integrity of software artifacts are all part of it. Software artifacts are intended to be consistent, traceable, and accessible through configuration management.

- *Configuration Identification*: The methods for identifying and naming configuration items (CIs) in the ICES4HU system are described in this standard. The naming conventions for CIs and the method for tracking versions should be described in the specification. It should also explain how CIs are added to and removed from

the system, as well as how changes to CIs are monitored over time. An accurate and consistent identification of every configuration item (CI) is made possible by effective configuration identification, enabling efficient and effective administration throughout the whole software development lifecycle.

- *Configuration Control*: The steps and policies for generating, changing, and releasing CIs in the ICES4HU system are described in this standard. The specification should outline how roles and permissions are utilized to regulate access to CIs as well as how modifications to CIs are documented and approved. Effective configuration control minimizes the possibility of mistakes and inconsistencies in the system by ensuring that all modifications are properly allowed and documented.
- *Configuration Status Accounting*: This specification outlines how the ICES4HU system monitors and communicates CIs' statuses over the course of their existence. The standard ought to detail how the current state of CIs are modified and expressed, together with details on their location, version, and modifications. Achieving good status accounting allows for successful and efficient leadership throughout the whole software development lifecycle by providing visibility into the current state of all CIs.
- *Configuration Management Tools*: This specification outlines the tools and software used to handle the configuration and change management process in the ICES4HU system. The functionality and features of these tools, as well as how they help the configuration management process, should be described in the specification. During the software development lifecycle, the system is appropriately managed and regulated thanks to effective configuration management solutions.

Change Management, on the other hand, is the process for handling changes to the procedure for developing software itself, including changes to project scope, time frames, and resources. To reduce conflicts and errors, it entails the effective administration of change requests, appraisal, approval, and implementation. To make sure that modifications to the software development process are effectively managed and communicated to stakeholders, change management was created.

- *Change Management:* The processes for handling changes to CIs in the ICES4HU system are described in this specification. The specification should outline the procedures for submitting, reviewing, and approving change requests as well as for implementing and verifying modifications. Effective change management minimizes the possibility of mistakes and inconsistencies in the system by ensuring that all changes are fully authorized, recorded, and executed.
- *Configuration Management Plan:* This specification outlines the overall management of configurations plan for the ICES4HU system. The specification has to outline the team's roles and responsibilities, the methods for managing and controlling CIs, and the procedures for tracking and documenting CI modifications. The danger of mistakes and inconsistencies in the system is decreased by the configuration management plan, which makes sure that the system is appropriately managed and regulated throughout the software development lifecycle.

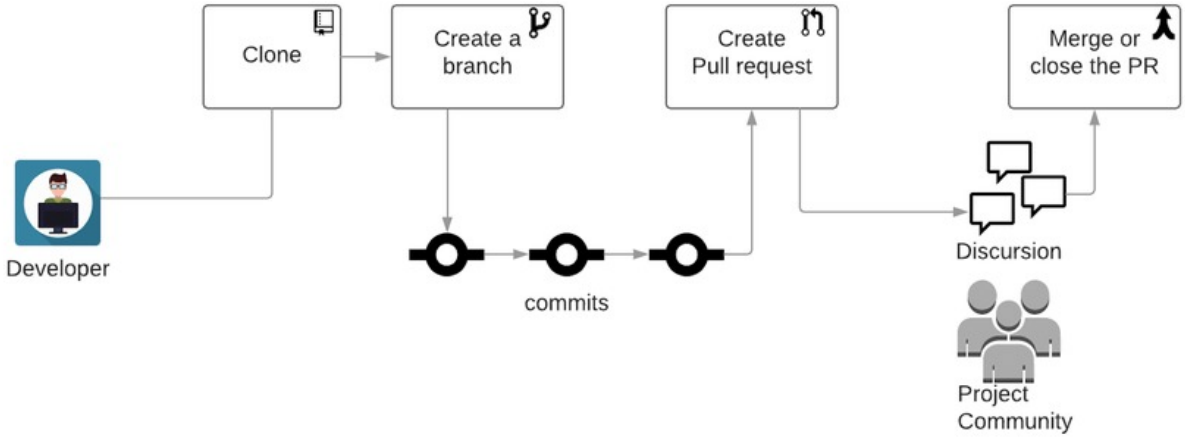
4 Key Considerations

There are a number of important factors that must be taken into account when creating the

ICES4HU system to guarantee that Configuration and Change Management are correctly implemented. The size and complexity of the system are two of the most important variables. Change management and control will be more difficult the bigger and more complicated the system is. To make sure that modifications are thoroughly tested and verified before implementation, it is crucial to develop an in-depth knowledge of the system's architecture and dependencies.

The number of parties involved in the development process is a further important consideration. ICES4HU is a multi-stakeholder system at Hacettepe University that includes students, teachers, department chairmen, faculty managers, and university committees. It is essential for all stakeholders to work together and communicate effectively to ensure that all demands and expectations are met. To accomplish this, it is necessary to establish transparent channels for feedback and input.

An effective version control solution for configuration and change management is GitHub. It gives developers a central location to store and manage code and documentation, enabling them to work concurrently on the same project. To keep track of changes, preserve version histories, and make sure that all stakeholders are working with the most recent system version, GitHub can be used as a configuration and change management platform. It is also simpler to handle the Configuration and Change Management process thanks to GitHub's developer collaboration and communication features. To make sure that all parties understand how to utilize GitHub efficiently and adhere to the established processes, it is crucial to develop explicit policies and procedures.



In addition to these considerations, it's necessary to think about how system changes might affect people. Unintended consequences of system modifications may affect the stability or efficiency of the system. Therefore, it's crucial to have alternate strategies established in case of unforeseen problems. This includes putting backup systems and practices in place to reduce disruptions and guarantee business continuity.