▪ Mohammed Fachry Dwi Handoko
▪ 5025201159

Multilevel Ontology Framework for Improving Requirements Change Management in Global Software Development

The Multilevel Ontology Framework for Improving Requirements Change Management in Global Software Development is a comprehensive approach aimed at enhancing the management of requirements changes in software development projects conducted across different geographical locations. In global software development, teams are often distributed across multiple locations, making coordination and communication challenging. In short, it aims to enhance the efficiency, quality, and overall success of software projects conducted in a distributed environment.

In software engineering, requirement change refers to the modifications or additions made to the initial set of requirements defined for a software project. These changes may arise due to various factors such as evolving business needs, technological advancements, user feedback, market dynamics, or external factors like regulatory compliance. Requirements changes are a frequent occurrence during the software development lifecycle, and effectively managing these changes is crucial to ensure project success.

The framework incorporates ontologies at multiple levels, namely the business level, process level, and technical level. This multilevel approach ensures that requirements changes are captured and analyzed from various perspectives, facilitating a comprehensive understanding of their impact.

The business level focuses on capturing high-level business goals, objectives, and requirements. It provides a shared understanding of the project's overall scope and aligns the different stakeholders towards a common vision. By considering the business context, the framework ensures that requirements changes are evaluated in terms of their impact on strategic objectives.

The process level ontology encompasses the software development process itself. It defines a standardized set of processes, methodologies, and best practices to be followed during the development lifecycle. This level ensures consistency and clarity in managing requirements changes across global teams, facilitating effective coordination and communication.

The technical level ontology concentrates on the technical aspects of the software system, including architecture, design patterns, and implementation details. It captures the technical requirements and constraints of the system. By considering the technical dimension, the framework enables an evaluation of how requirements changes affect the system's design, implementation, and performance.

The Multilevel Ontology Framework offers several benefits for requirements change management in global software development. By capturing and representing business goals and technical constraints, the framework ensures better alignment between stakeholders, minimizing misunderstandings and conflicts during requirements changes.

The use of ontologies and collaborative tools promotes effective communication among distributed teams, enabling them to share insights, raise concerns, and discuss the impact of changes in a structured and organized manner. The automated reasoning capabilities and the multilevel nature of the framework enable efficient impact analysis, allowing teams to assess the consequences of requirements changes more comprehensively and accurately. By facilitating a thorough understanding of requirements changes, the framework helps mitigate risks associated with incorrect or incomplete changes. It also reduces the costs and delays caused by rework or reevaluation of changes due to insufficient analysis.

Through better coordination, traceability, and decision-making, the framework contributes to improved project success rates in global software development. It enhances the ability to deliver software that meets stakeholders' expectations and fulfills the business objectives.

The use of automated reasoning and inference techniques enhances the efficiency and accuracy of impact analysis, reducing manual effort and minimizing the risk of overlooking potential effects of changes. This automated analysis provides a reliable basis for decision-making regarding requirements changes.

Collaborative tools and platforms integrated into the framework foster effective communication and coordination among globally distributed teams. These tools provide real-time access to ontologies, enabling stakeholders to collaborate, discuss, and reach consensus on requirements changes. By facilitating transparent and well-informed discussions, the framework reduces miscommunication and ensures that all team members are aligned.

A Framework for Emotion-Oriented Requirements Change Handling in Agile Software Engineering

Agile software engineering is an iterative and collaborative approach to software development that emphasizes adaptability and responsiveness to change. It involves breaking down a project into smaller increments called sprints, where requirements can be identified, developed, and tested in short cycles.

Encouraging team members to be aware of their own emotions and the emotions of others involved in the project. This can be achieved through open and honest communication, active listening, and empathy. Actively involving stakeholders in the requirements change process. This includes understanding their emotional needs and concerns, facilitating discussions to gather feedback, and addressing any emotional barriers that may arise. Evaluating the potential emotional impact of requirements changes on different stakeholders. This involves considering factors such as their expectations, preferences, and the perceived value of the changes

Maintaining clear and transparent communication channels to manage emotions related to requirements changes. This includes providing regular updates, explaining the rationale behind changes, and ensuring that stakeholders are well-informed throughout the process. Encouraging collaboration and participation from all relevant stakeholders when making decisions about requirements changes. This can help mitigate negative emotions and increase stakeholder buy-in. Utilizing agile rituals and techniques, such as daily stand-up meetings, retrospectives, and user story mapping, to facilitate emotional awareness and effectively manage requirements changes. These practices can foster a supportive and collaborative environment.

Developing emotional intelligence within the team can enhance collaboration, empathy, and effective communication. Team members with higher emotional intelligence are better equipped to handle the emotional impact of requirements changes, respond to conflicts constructively, and maintain positive working relationships. Collecting user feedback during the development process allows software teams to understand how users are experiencing the product. This feedback can include emotional responses, such as frustration, delight, or confusion.

Integrating emotions into requirements change handling involves actively seeking and incorporating user emotions and feedback to inform decision-making and prioritize changes that have a positive emotional impact. Agile ceremonies, such as daily stand-ups, retrospectives, and sprint reviews, provide opportunities to address emotions and emotional well-being within the team. These rituals can create a supportive environment for discussing emotions related to requirements changes, sharing concerns or challenges, and collectively finding solutions. Emphasizing emotional well-being in the agile process can contribute to team morale, motivation, and productivity.

Requirements changes can sometimes lead to conflicts among stakeholders, as different perspectives and interests come into play. Effective conflict resolution and negotiation techniques can help manage emotional tensions and facilitate productive discussions. Techniques like active listening, empathy, and compromise can foster a shared understanding and agreement on requirements changes. It's important to note that while emotions are an essential aspect of the human experience, they may not always be explicitly addressed in software engineering frameworks or methodologies. The integration of emotions into requirements change handling in agile software engineering is still an emerging fieldfield.