

Improve the Scrum Process Model for the Structure and Behavior of the IT Organization using the Viable System Model

Muhammad Hamid ^{1,*} , Muhammad Saleem ²  and Saadia Malik ³ 

¹ Department of Computer Science, Government College Women University Sialkot, Pakistan; mhamid@gcwus.edu.pk.

² Department of Industrial Engineering, Faculty of Engineering, Rabigh King Abdul-Aziz University, Jeddah, Saudi Arabia; msaleim1@kau.edu.sa.

³ Department of Information Systems, Faculty of Computing and Information Technology-Rabigh King Abdul-Aziz University, Jeddah, Saudi Arabia; Smalik1@kau.edu.sa.

* Correspondence: mhamid.itprofessional@gmail.com.

Abstract: The Scrum process model is widely adopted in the IT industry to improve project management and software development. However, there is an increasing need to enhance the structure and behavior of IT organizations within this framework. This research addresses this by proposing the integration of the Viable System Model (VSM) into the Scrum process. The VSM is a proven framework for analyzing and improving the viability of complex systems. Integrating VSM into Scrum provides a holistic view of an organization, considering its internal and external interactions, power dynamics, and overall viability. This integration aims to enhance organizational effectiveness and improve project outcomes. The research investigates how VSM can optimize Scrum in IT organizations by identifying and addressing challenges such as communication gaps, decision-making inefficiencies, and power struggles. By leveraging VSM insights, organizations can better align roles, responsibilities, and processes, leading to increased productivity, improved collaboration, and stronger project performance. The proposed approach is tested using actual software project data, and results show reduced delays, improved quality, and higher stakeholder satisfaction. In conclusion, this study advances the Scrum model by incorporating VSM to strengthen organizational structure and behavior, resulting in better project outcomes.

Keywords: Agile Methodology; Fuzzy Logic System; Scrum Process Model; Viable System Model.

1. Introduction

The origins of agile methodologies are closely linked to the concepts of iterative and incremental development [1]. Traditional development methodologies are surpassed by breaking down the plan, execution and revision phases into smaller iterative cycles, which promote practical and measurable outcomes at the end of each cycle. One of the significant challenges in modern project management is the growing complexity and dynamism of changing conditions. Project requirements are becoming increasingly complex, causing substantial problems for project managers in meeting these requirements in software development. We use process models such as the agile model Scrum to deal with this complexity. Scrum is an agile methodology used to manage development projects through an iterative and incremental approach. The name Scrum comes from a technique used in the game of English Rugby, where the ball is passed between two team members on the field. The same rugby technique is used in the scrum methodology to promote interactions among team members and divide the work into sprints. While Scrum is easy to understand, it may not be easy to implement in large projects where complex problems and distributed development teams can lead to losing control. Managing organizational politics, power dynamics and technostress in large projects or IT organizations can also be challenging [2-4]. We often focus on resolving one problem at a time during the development cycle to address these challenges. However, this approach can lead to higher costs, missed project deadlines and declining product quality. It's important to remember that there is a science, cybernetics, that can help us manage complexity.

Cybernetics is a natural science that forms the foundation of many well-functioning processes. It helps guide a project in the right direction from start to finish and shows how to navigate complexity without fighting against it [5, 6]. Cybernetics is an integrative multidisciplinary meta-science encompassing various theories, including the theories of information and communication and the theory of regulation and control. Without the principles of cybernetics, many things would not work, including aircraft, computers, large cities and organisms. As one of the most fundamental and powerful sciences, cybernetics incorporates essential mechanisms for dealing with complexity, such as self-control, self-regulation and self-organization. In our increasingly complex world, cybernetics provides the invariant laws of functioning that apply to biological, technical, physical, social and economic systems. However, many people are not aware of this. The question arises as to whether the Scrum process meets the basic requirements of cybernetics to guide the entire development cycle effectively. This question is not limited to specific business cases but is fundamental. Cybernetics defines the basic laws that must be met; otherwise, the development process could become uncontrollable. We have begun our research activities with the agile process model Scrum, which is well-known and widely used in business sectors where requirements are volatile. The cybernetic aspects of Scrum have been examined and they have helped control the development process. The main question of this thought paper is how to improve the performance of the Scrum process model for managing large projects, dealing with and maintaining the reliability of software development in IT organizations through the use of cybernetics and a viable system model to control the complexity of existing Scrum process and controlling all dependent system integration among five systems, which is supervised by a meta-organizational unit.

Section 2 describes related work in detail. Section 3 describes the methodology in detail, while Section 4 covers the prototype demonstration and evaluation. Finally, Section 5 provides conclusions and outlines future work.

2. Related Work

In this section, we will briefly introduce the existing Scrum model and explain the need for a viable system model and the importance of organization structure and behavior.

2.1. Scrum Process Model

The Scrum process model is a framework for managing complex projects, particularly in software development. It was developed using Agile methodology, emphasizing flexibility, collaboration and customer satisfaction. Scrum provides a structured approach to managing projects by breaking them down into smaller, manageable pieces and regularly reviewing progress. Scrum is based on roles, events and artifacts that work together to ensure the project is completed on time and within budget. The leading roles within a Scrum project are the Product Owner, Scrum Master and Development Team. The Product Owner is responsible for defining the project's goals and ensuring that the project meets the customer's needs. The Scrum Master is responsible for facilitating the process, removing obstacles and ensuring the team follows the Scrum methodology. The Development Team is responsible for completing the work and delivering the product.

The main events within a Scrum project are Sprint Planning, Daily Scrums, Sprint Review and Sprint Retrospective. Sprint Planning is a meeting where the team plans the work for the upcoming sprint. The Daily Scrum is a short meeting held every day to review progress and plan for the day ahead. The Sprint Review is a meeting where the team reviews the work completed during the sprint and receives feedback from the customer. The Sprint Retrospective is a meeting where the team reflects on the sprint and identifies areas for improvement. The primary artifacts within a Scrum project are the Product Backlog, Sprint Backlog and Increment. The Product Backlog is a list of all the work that needs to be done for the project. The Sprint Backlog lists the work done during the upcoming sprint. The increment is the work completed during the sprint and added to the product.

The scrum process model is a flexible and iterative approach to software development that allows teams to respond to changing requirements and adapt to new information. It's designed to be transparent, inspectable and adaptable, which helps the team to deliver high-quality software that meets the customer's needs. The scrum process is widely used in software development, product development, IT services, education and manufacturing. The Scrum methodology for software development is represented in Figure 1.

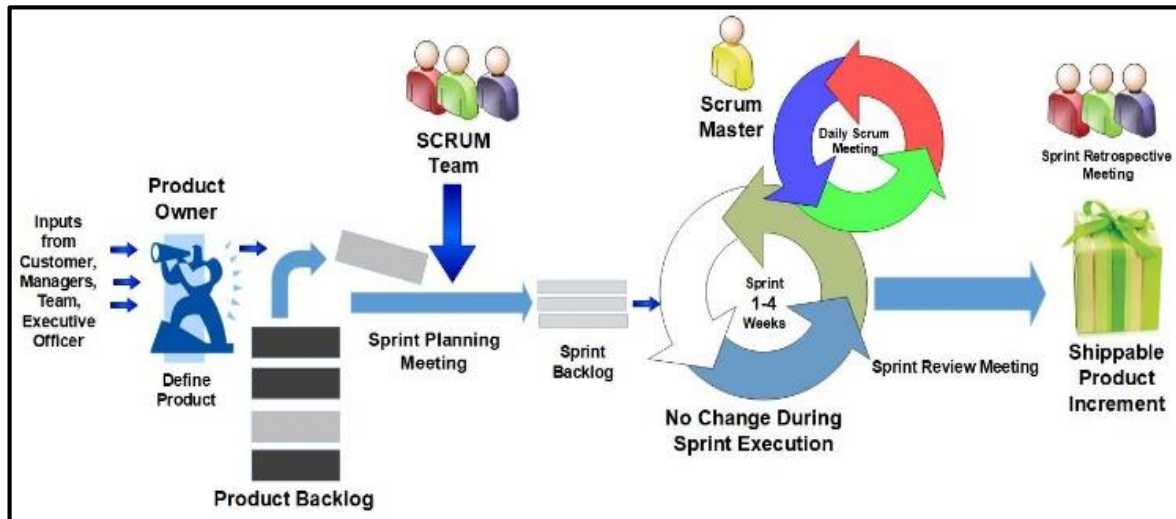


Figure 1. Scrum development process [3].

A significant problem in Scrum is handling and controlling distributed teams working on large projects. In a changing environment, seemingly minor changes can have substantial unintended impacts on the system, requiring software developers to have mechanisms in place to understand the impact of changes on the system. As software evolves, maintaining it becomes more complex and more expensive. The software maintenance process can only be optimized with precise and unambiguous information about the potential ripple effects of a change on an existing extensive system. Furthermore, Scrum software methodology emphasizes an iterative and incremental high pace of software delivery, which can lead to inadequate time spent on software reliability concerns. Due to the short development cycles, Scrum teams focus on delivery through increments. However, this can lead to system failure issues during the operational phase if proper analysis and testing concerning reliability concerns are not performed in the Scrum process model [7].

2.2. Viable System Model

The Viable System Model (VSM) [8] is a framework for understanding and managing complex systems. Developed by British cyberneticist Stafford Beer, it is based on the idea that any viable system, whether an organization, a society, or an ecosystem, must have specific characteristics to function effectively and adapt to change. These characteristics include a clear hierarchy of decision-making, effective communication and coordination and the ability to learn and adapt to changing conditions.

The VSM is based on the idea that a viable system comprises five main components: the environment, the system itself, the management function, the operating function and the meta-management function. The environment represents the external factors that the system must adapt to, such as market conditions, regulations and technology. The system is the central unit of analysis, responsible for achieving its goals and objectives. The management function makes strategic decisions and allocates resources to the operating function, which is accountable for the system's day-to-day operations. Finally, the meta-management function is responsible for monitoring and controlling the

system's overall performance and ensuring that it is aligned with the goals and objectives of the organization.

The VSM is a powerful tool for understanding and managing complex systems, as it provides a clear framework for identifying the key components of a system and their interactions. It can be used to analyze any system, such as a business, a government, or a social organization. The VSM can be applied to different levels of organization, from the individual to the team and the entire organization. It has been used in various IT, healthcare and manufacturing industries. In short, VSM is a powerful tool for understanding and managing complex systems, providing a clear framework for identifying the key components of a system and their interactions. It can be applied to different levels of organization, from the individual to the team and the entire organization; it is widely used in various industries to improve performance and the ability to adapt to change. Table 1. gives a brief introduction to these five systems. All five systems are continuously interacting.

Table 1. Brief Overview of VSM Mechanisms.

System 1	It is the collection of all the operational components of the system, such as departments, teams, and individuals. It is responsible for carrying out the day-to-day activities of the organization.
System 2	It is responsible for maintaining the stability and integrity of the system and resolving conflicts among the operational components. It acts as a coordination and control mechanism, making sure that the system is working towards its goals and objectives.
System 3	It is responsible for optimizing the performance of the system and generating collaboration among the operational components. It acts as a decision-making and planning unit, determining the best course of action for the system to achieve its goals and objectives.
System 4	It is responsible for future planning and strategy, and for adapting the system to a changing environment. It acts as a monitoring and control unit, making sure that the system is aligned with its goals and objectives, and that it is able to adapt to changes in the environment.
System 5	It is responsible for policy-making, visioning, and high-level strategy. It acts as a guidance and control unit, making sure that the system is aligned with its goals and objectives, and that it is able to adapt to changes in the environment. It interacts with System 3 and System 4 to ensure that the system is working towards its goals and objectives, and that it is able to adapt to changes in the environment.

The VSM system comprises five subsystems: operation, coordination, control, planning and identity. It can also be applied and mapped to different scenarios, such as project management and organizational modeling.

2.3. IT Organization Structure and Behavior

The structure and behavior of IT organizations can significantly impact their overall performance and effectiveness [9]. IT organizations typically have a hierarchical structure, with different levels of management and other teams responsible for various aspects of IT operations.

One typical structure for IT organizations is the traditional hierarchical structure, which is based on a top-down approach to decision-making and management. This structure is characterized by a transparent chain of command and a centralized decision-making process. It is generally considered efficient and effective for stable organizations with well-defined processes.

Another standard structure for IT organizations is the matrix, which combines the traditional hierarchical structure with a project-based structure. This structure organizes teams based on specific projects or functions rather than traditional departmental lines. It is generally considered more flexible and adaptable than the conventional hierarchical structure and is often used in organizations undergoing rapid growth or change.

In addition to structure, IT organizations' behavior can significantly impact performance and effectiveness. For example, a culture of innovation and experimentation can lead to the development of

new technologies and processes. In contrast, a culture of bureaucracy and inflexibility can lead to stagnation and inefficiency.

Furthermore, the level of collaboration and communication within an IT organization can also significantly impact performance and effectiveness. A culture of open communication and collaboration can lead to better decision-making. In contrast, a culture of silos and lack of communication can lead to a lack of coordination and inefficiency.

In short, the structure and behavior of IT organizations can significantly impact their overall performance and effectiveness. Different structures, such as hierarchical and matrix structures, can be chosen depending on the organization's needs and goals. In addition, the IT organization's culture, behavior and communication are crucial to its effectiveness.

2.3.1. Organization Power and Politics

Power and politics play a significant role in the functioning of IT organizations [10]. Power refers to the ability of individuals or groups to influence the actions and decisions of others. In contrast, politics is using power to achieve specific goals or objectives. In IT organizations, power and politics can manifest in various ways, including allocating resources, selecting projects and developing policies and procedures.

Various individuals and groups can hold Power in IT organizations, including senior management, technical experts and project managers. Senior management, for example, has power through their control over the organization's budget and resources, while technical experts hold power through their knowledge and expertise in specific technology areas. Project managers hold power through their ability to control and direct the resources and personnel of a particular project.

Politics in IT organizations can take many forms, including forming coalitions, manipulating information and using formal and informal networks. Coalitions can form around specific projects or initiatives, with individuals and groups working together to achieve common goals. Information manipulation can influence decisions and gain support for particular projects or initiatives. Formal and informal networks can access resources and information and build support for specific projects or initiatives.

Power and politics can have both positive and negative effects on IT organizations. On the one hand, they can lead to allocating resources to essential projects, developing effective policies and procedures and selecting the best projects and initiatives. On the other hand, they can lead to allocating resources to less important projects, developing ineffective policies and procedures and selecting less-than-optimal projects and initiatives.

In short, Power and politics play a significant role in the functioning of IT organizations. Different individuals and groups within the organization, such as senior management, technical experts and project managers, hold power. Politics can take many forms and can be used to achieve specific goals or objectives. However, power and politics can positively and negatively affect IT organizations. It is essential for IT organizations to have a clear understanding of the dynamics of power and politics and to develop strategies for managing them effectively.

2.3.2. Technostress in Distributed Scrum Teams

Technostress is a term used to describe the adverse psychological effects of technology on individuals [11]. Distributed Scrum teams, where team members work remotely or in different locations, are particularly vulnerable to technostress due to their increased reliance on technology for communication and collaboration.

Distributed Scrum teams often use various technologies like video conferencing, instant messaging, and project management tools to communicate and collaborate effectively. While these technologies can increase productivity and effectiveness, they can also lead to feelings of isolation, overload and burnout.

One aspect of technostress for distributed Scrum teams is the lack of face-to-face interaction. Distributed team members may feel isolated and disconnected from the team, which can lead to decreased motivation and engagement. Additionally, the constant availability and pressure to respond to messages and updates from team members can lead to feelings of overload and burnout.

Another aspect of technostress for distributed Scrum teams is the reliance on technology for communication and collaboration. Distributed team members may feel pressure to use technology to communicate with other team members, even when it is not the most appropriate or effective way. Additionally, the constant need to be connected to technology can lead to feelings of disconnection from the team and work.

To mitigate the effects of technostress on distributed Scrum teams, it is essential to establish clear guidelines and expectations for communication and collaboration and to promote face-to-face interactions when possible. Additionally, it is critical to provide team members with the necessary training and support to use the technologies they are required to use effectively. Regular check-ins and open communication channels can also help address issues and support team members struggling with technostress.

In short, technostress is a term used to describe the adverse psychological effects of technology on individuals. Distributed Scrum teams are particularly vulnerable to technostress due to the increased reliance on technology for communication and collaboration.

To mitigate the effects of technostress on distributed Scrum teams, it is essential to establish clear guidelines and expectations for communication and collaboration, to provide team members with necessary training and support and to promote face-to-face interactions when possible. Regular check-ins and open communication channels can also help address issues and support team members struggling with technostress.

3. Proposed Methodology

The literature analysis presented in a previous section highlights the problems organizations encounter using the Scrum process and the advantages the VSM framework offers. The analysis suggests that integrating the Scrum process and VSM could improve the performance of distributed Scrum teams that handle large projects. This integration would also enhance the reliability of IT organizations by reducing time to market, minimizing defects and lowering costs. It also introduces the concept of organizational power and politics within project functionality processes, which can impact the organization's structure and behavior. In the IT industry, the proposed integration of Scrum and VSM is viewed as a new approach to meeting customer needs and developing new software platforms. Additionally, it addresses the issue of technostress, which arises from the need to enhance software development constantly and the balancing of the overload of new technology with software and customer needs within the IT organization or software development company. The proposed extended Scrum model using VSM is illustrated in Figure 2.

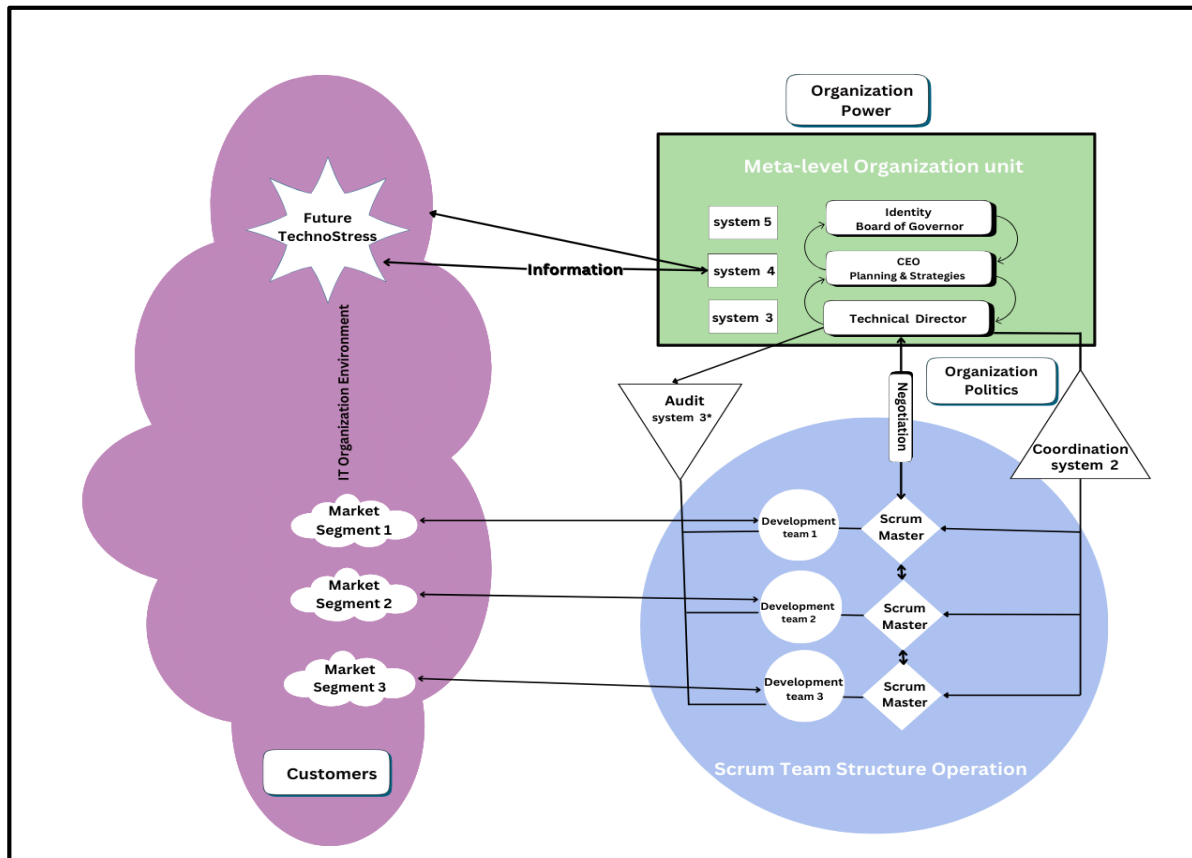


Figure 2. Proposed framework for scrum process model using VSM.

3.1. IT Organization Environment

In the IT environment element, the VSM-based framework offers a way to analyze communication problems within an IT organization, optimize the use of resources, reduce wastage, and improve the ability to adapt to changes in future technostress and IT market environments. This is achieved by measuring daily regulation at various levels of recursion, such as at every level from the scrum team structure and operations unit to the management unit, and by emphasizing the system in focus. The VSM framework focuses on the analysis of the IT organization's internal and external communication, resource utilization, and ability to adapt to changes in the market environment. By measuring daily regulation at different levels of recursion, the framework can identify areas for improvement and provide suggestions for addressing issues related to communication, resource utilization, and adaptability. This approach emphasizes the importance of the system in focus, which is the IT organization, and how it interacts with its internal and external environment. Overall, the VSM-based framework provides a comprehensive approach to analyzing and improving the performance of an IT organization.

3.2. Scrum Team Structure Operation

This independent systems element emphasizes the operation of the scrum team structure. It has two further sub-operations: a development team and one documentation team. Scrum masters manage these sub-operations, which handle the daily stand-up meetings and sprint reviews. This model aims to control the complexity of the existing scrum process and improve the reliability of large projects by providing documentation in parallel processing after the sub-operation unit feedback within this element. This element handles the primary activity of System 1 operations functionality.

The emphasis on the operation of the scrum team structure in this independent systems element is critical for the project's success. The development and documentation teams are separate sub-operations managed by Scrum masters. These Scrum masters oversee the daily stand-up meetings and sprint reviews, which are essential for keeping the project on track and identifying any issues that need to be addressed.

The model aims to control the complexity of the existing scrum process and improve the reliability of large projects by providing documentation in parallel processing after the sub-operation unit feedback. This ensures that all the necessary information is available at the right time, which helps to minimize defects, reduce time to market and lower costs. The primary activity of System 1 operations functionality, which handles this documentation, is crucial for the project's success and overall performance.

Overall, this independent systems element of the proposed framework provides a comprehensive approach to managing the scrum team structure, ensuring the reliability of large projects and controlling the complexity of the existing scrum process.

3.3. Meta Organization Management Unit

This unit adapts the IT organization to its external and internal future technostress environments. Different individuals manage three central systems: the technical director handles system three operations, the CEO is responsible for planning and strategies, system four operations are handled and the identity of the board of governors handles system five operations. The overall management unit controls all systems and operations within this framework.

3.4. Proposed Framework Subsystems

This system is considered viable if the above three elements are balanced. The overviews of these five sub-systems are given below: System 1 performs the operation task handled by the teams and overseen by the scrum master. System 1 integrates with System 2 through the politics and conflicts among all teams. These results are forwarded to System 3, the management system's second element and deal with the technical director's functionality. System 3 finds the optimized solution among the three teams' operations and forwards it to the planning and strategy of the IT organization, which is dealt with by System 4. System 4 deals with the outside environment through future techno-stress, meaning how many applications run on the market and which technology is currently used for competition among different organizations. Finally, System 5 has the power and policy to process and approve all these integrated elements. All these system functionalities are discussed below based on the five functions: operations, coordination, control, planning and policy.

3.4.1. System 1 (Operation)

System 1 operations are the core of the recursive VSM framework. It performs the primary activity of three teams, which are handled and overseen by scrum masters. System 1 integrates with System 2 by managing organizational politics among all operations and conflicts. It also integrates with the external environment; each team interacts with different market segments of the outside environment through stakeholders' meetings.

In more detail, the scrum master's role is to facilitate the work of the three teams, ensuring that they are working collaboratively and efficiently towards completing the primary activity. System 1 integrates with System 2 by clearly understanding the organizational politics that may impact the operations and addressing conflicts. Additionally, the system 1 teams interact with the external environment through stakeholders' meetings, which allow them to gain insight into the needs and expectations of different market segments and to ensure that their work aligns with these needs and expectations.

3.4.2. System 2 (Coordination)

The essential role of System 2 is to coordinate the operations unit's problems with the System 3 technical standards and the direction of information flows. This information mechanism involves organizational standards, protocols, project timeline scheduling and all scrum team operation programs, among other things. IT automation systems can be designed and implemented with system 2 to reduce direct communication and human intervention. This can help to manage and control organizational politics. The mutual adjustment and coordination between the primary activities of the operations unit and management units' primary activities is effective in redesigning value-adding activities.

In more detail, system 2 acts as a bridge between the operations unit and system 3, responsible for technical standards and direction of information flows. This coordination is essential to ensure that the organization's standards and protocols are met and that the project timeline is followed. System 2 also plays a role in managing the scrum teams' operation programs. IT automation systems can be integrated into System 2 to help automate specific processes, which can reduce the need for direct communication and human intervention. This can improve efficiency and reduce the potential for human error. The mutual adjustment and coordination between the primary activities of the operations unit and management units' primary activities is necessary for redesigning the value-adding activities to meet the organization's goals and objectives.

3.4.3. System 3 (Control)

In system 3, the control unit manages the coordination activities from system 2 and negotiates with the operation unit's sub-management for responses. The essential role of system 3 is to provide synergy among resource management, monitoring performance measures through intervention rules and regulations. The technical director of the scrum project management performs this role. Another system part of system 3 is system 3*, which has the essential role of creating a channel for auditing and monitoring between the development teams and documentation team, as well as the technical director of system 3. The basic aim of this channel is to be aware of internal environment activities and produce and build new strategic planning at a higher level. This information is delivered to System 4 and a cycle of improvement in planning and implementation is built between System 3 and System 4.

In more detail, system 3's control unit is responsible for managing the coordination activities from system 2 and negotiating with the operation unit's sub-management for responses. The technical director of the scrum project management plays a key role in system 3 by ensuring that resources are managed effectively and that performance measures are monitored through intervention rules and regulations. Additionally, system 3* is also a part of system 3. Its primary role is to create a channel for auditing and monitoring between the development and documentation teams and the technical director of system 3. The purpose of this channel is to keep the technical director informed of the internal environment activities, which will help to build new strategic planning at a higher level. This information is then delivered to system 4, which enables a continuous cycle of improvement of planning and implementation between system 3 and system 4.

3.4.4. System 4 (Planning)

In system 4, planning and strategies take place at a higher level and are focused on processing and interacting with two-way communication between the organization's management and the external environment. The external environment includes market segments and the future technostress environment, which are demands and new opportunities for the IT organization's growth. This information is transmitted to the upper-level board of governors for decision-making and creation of new policies. It is also transmitted to the technical director for further implementation.

The everyday activities of system 4 include forecasting future technostress for IT software growth, market and operation research, current trends of development, and public relations worldwide.

In more detail, system 4 is responsible for planning and strategies at a higher level. The planning process in system 4 involves interacting with two-way communication between the organization's management and the external environment. The external environment includes the market segments and the future technostress environment, which are the demands and new opportunities for the IT organization's growth. This information is then transmitted to the upper-level board of governors for decision-making and creation of new policies. It is also transmitted to the technical director for further implementation. The everyday activities in system 4 include forecasting future technostress for IT software growth, market and operation research, identifying current development trends, and managing public relations worldwide.

3.4.5. System 5 (Policy)

This last system is responsible for preparing and designing IT organization policies and providing direction, values and goals for the Meta organization unit. The primary role of this system is to make decisions within and between the planning strategy system 4 and the control functions of system 5. This communication is supervised by the board of governors, which includes the overall division of planning and strategy sections and representatives from the outside environment, such as state authorities. They supervise the synergy of all systems' communication and ensure its viability within and outside the competitive environment.

In more detail, system 5 is responsible for preparing and designing IT organization policies that provide direction, values, and goals for the Meta organization unit. The primary role of System 5 is to make decisions within and between the planning strategy of System 4 and the control functions of System 5. This communication is supervised by the board of governors, which is composed of representatives from the overall division of planning and strategy sections, as well as representatives from the outside environment, such as state authorities. The board of governors plays a critical role in supervising the synergy of all systems' communication and ensuring that the IT organization is viable within and outside the competitive environment. They are responsible for ensuring that all the systems' communication is aligned with the overall goals and objectives of the organization and that the organization can operate successfully in the external environment.

4. Prototype Demonstration and Evaluation

The demonstration and evaluation phase consists of four activities. The first activity describes the fuzzy inference system for predicting software project outcomes. The second activity describes the input and output of the knowledge base fuzzy system. The third activity presents the rules-based approach and the fourth discusses the experimental results generated using MATLAB.

4.1. Fuzzy Inference System for Prediction of Software Project Outcomes

The proposed system employs mathematical functions, referred to as membership functions, which provide statistical representations of input and output variables. These membership functions can be found in the MATLAB toolbox. Figure 3 shows what goes into and comes out of the Fuzzy inference system.

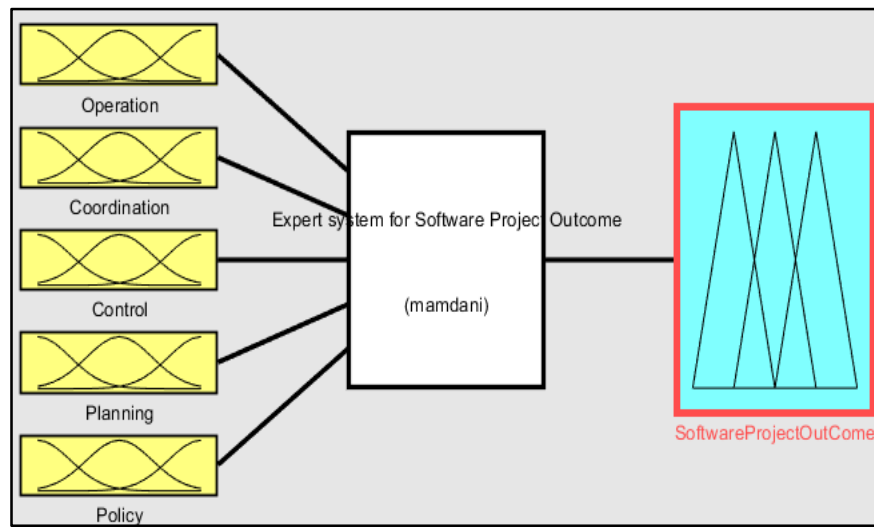


Figure 3. Input and output of the system.

4.2. Knowledge Base Input and Output Fuzzy System

The proposed system has five input variables and one output variable: operation, coordination, control, planning and policies. Tables 2 and 3 describe the input and output variables in detail.

Table 2. Ranges for Fuzzy Inference System for Software Project Outcomes.

Fuzzy Input Variable	Linguistic Term	Parameters of Fuzzy set
Operation	{“Small,” “Moderate,” “High”}	[0,0.50,1]
Coordination	{“Low,” “Medium,” “High”}	[0,0.50,1]
Control	{“Low,” “Medium,” “High”}	[0,0.50,1]
Planning	{“Low,” “Medium,” “High”}	[0,0.50,1]
Policy	{“Normal,” “Moderate,” “Dangerous”}	[0,0.50,1]

Table 3. Linguistic Representation of Output Variable for Software Project Outcomes.

Fuzzy Output Variable	Linguistic Term	Parameters of Fuzzy set
Software Project Outcome	{“Failed”, “Challenged”, “Succeeded”}	[0,50,100]

4.3. Rules-based

Fuzzy rule-based systems are adaptable and extensible, allowing them to make decisions based on various input and output shapes and different types of fuzzy logic. Fuzzy inference systems for predicting software project outcomes utilize fuzzy rules to represent a broad range of knowledge in other contexts. In this investigation, the rules were generated by Mamdani's fuzzy inference rule-making system, implemented in MATLAB. The rules diagram, illustrated in Figure 4, is graphical.

199. If (Operation is Moderate) and (Coordination is Medium) and (Control is Medium) and (Planning is Medium) and (Policy is Normal) then (SoftwareProjectOutcome is 36)

200. If (Operation is Moderate) and (Coordination is Medium) and (Control is Medium) and (Planning is High) and (Policy is Moderate) then (SoftwareProjectOutcome is 36)

201. If (Operation is Moderate) and (Coordination is Medium) and (Control is Medium) and (Planning is High) and (Policy is Dangerous) then (SoftwareProjectOutcome is 36)

202. If (Operation is Moderate) and (Coordination is Medium) and (Control is High) and (Planning is High) and (Policy is Moderate) then (SoftwareProjectOutcome is 36)

203. If (Operation is Moderate) and (Coordination is Medium) and (Control is High) and (Planning is High) and (Policy is Dangerous) then (SoftwareProjectOutcome is 36)

204. If (Operation is Moderate) and (Coordination is Medium) and (Control is High) and (Planning is High) and (Policy is Normal) then (SoftwareProjectOutcome is 36)

205. If (Operation is Moderate) and (Coordination is High) and (Control is Medium) and (Planning is High) and (Policy is Moderate) then (SoftwareProjectOutcome is 36)

206. If (Operation is Moderate) and (Coordination is High) and (Control is Medium) and (Planning is High) and (Policy is Dangerous) then (SoftwareProjectOutcome is 36)

207. If (Operation is Moderate) and (Coordination is High) and (Control is Medium) and (Planning is High) and (Policy is Normal) then (SoftwareProjectOutcome is 36)

208. If (Operation is Large) and (Coordination is High) and (Control is High) and (Planning is High) and (Policy is Dangerous) then (SoftwareProjectOutcome is 36)

If and and and and

Operation is Coordination is Control is Planning is Policy is

Small Low Low Low Normal

Moderate Medium Medium Medium Moderate

Large High High High Dangerous

none none none none none

☐ not ☐ not ☐ not ☐ not ☐ not

Figure 4. Input and output rules for software project output predication.

4.4. Experimental Results

Figure 5 describes the values of operation (0.109), coordination (0.0824), control (0.205), planning (0.178) and policy (0.0984). The software project outcomes value of 36 shows that the software project has failed.

Figure 6 describes the values of operation (0.109), coordination (0.0824), control (0.205), planning (0.178) and policy (0.0984). Since the software project outcomes value is 71, the software project has succeeded.



Figure 5. Lookup diagram for failed Software projects.

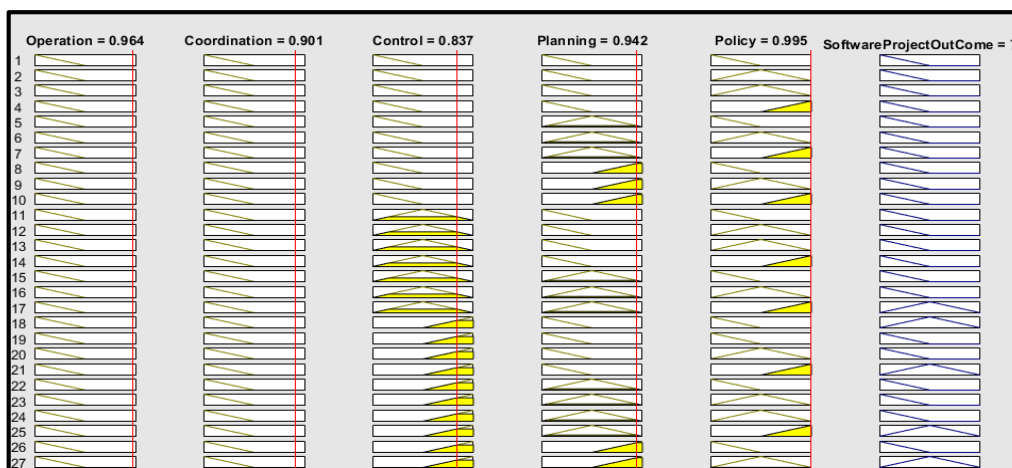
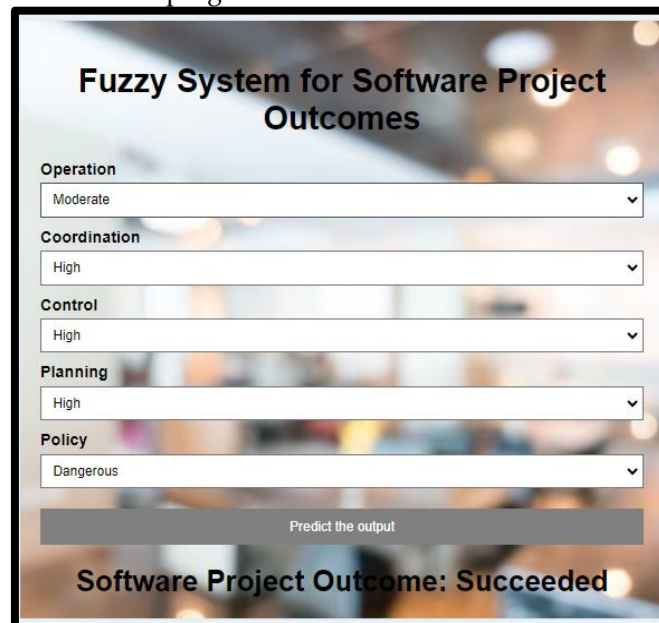


Figure 6. Lookup diagram for succeeded Software projects.

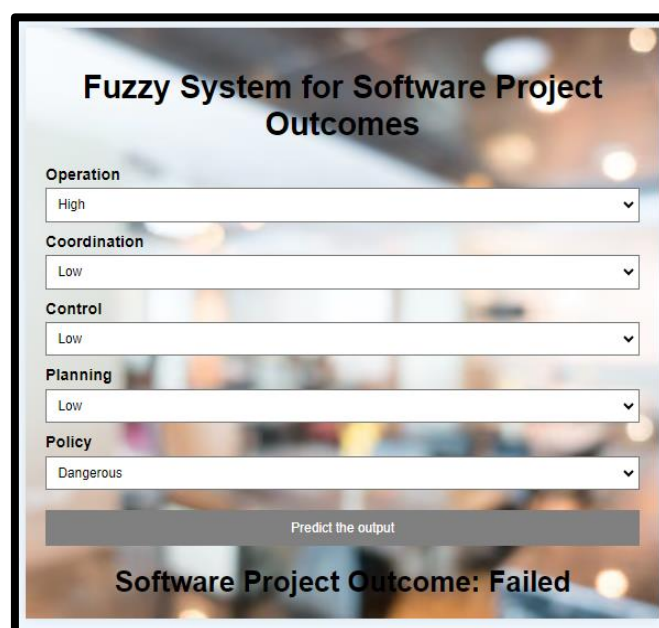
4.5. Fuzzy Inference System for Software Project Outcomes

The fuzzy inference system (FIS) for software project outcomes utilizes various project-related factors as input from the user. These factors include project Operation, Coordination, Control, Planning and Policy. By analyzing these inputs, the system can predict the likelihood of the project's success, which can be categorized into three levels: "Failed," "Challenged," and "Succeeded." The proposed system can be visualized as shown in Figure 7-8, which demonstrates how the inputs from the user are processed and used to determine the project's potential outcome. The system is designed to provide a quick and accurate assessment of the project's success probability, which can inform project planning decisions or track progress over time.



The screenshot displays a web-based interface titled "Fuzzy System for Software Project Outcomes". It features five input fields, each with a label and a dropdown menu: "Operation" (Moderate), "Coordination" (High), "Control" (High), "Planning" (High), and "Policy" (Dangerous). Below these fields is a button labeled "Predict the output". At the bottom of the interface, the result is displayed as "Software Project Outcome: Succeeded".

Figure 7. Fuzzy Inference System for Software Project Outcomes (Succeeded).



The screenshot displays the same web-based interface as Figure 7, but with different input values: "Operation" (High), "Coordination" (Low), "Control" (Low), "Planning" (Low), and "Policy" (Dangerous). The "Predict the output" button is present. At the bottom, the result is displayed as "Software Project Outcome: Failed".

Figure 8. Fuzzy Inference System for Software Project Outcomes (Failed).

5. Conclusion and Future Work

This paper proposed improving the Scrum process model for IT organization structure and behavior using the VSM. The paper presents a cybernetic analysis of the Scrum process through the VSM approach, which provides decision analysis and modeling of the extended Scrum process to address issues related to distributed development teams handling large projects. The VSM is a flexible and robust tool that can be used to design new IT organizations and diagnose the weaknesses of existing Scrum process models. This paper employed the VSM logic to improve the Scrum process. Systems 1 through 5 of the VSM were correlated with specific functions and elements. The performance of the Scrum development team increases through the parallel execution of tasks and early control of organizational politics in the operational phase. Proper utilization of organizational power maximizes the reliability of large projects. It reduces the chances of errors, thus decreasing the project's overall cost but increasing the time of the first few releases. In later releases, the overall time to market decreases. The proposed framework ensures the integrated design of these mechanisms, thereby securing the Scrum process viability and effectiveness within and outside its external environment. The author also believes that the extended Scrum process is mainly about making things more visible and measurable. It makes requirements more visible, progress more measurable and impediments more visible, as it is a management framework.

Declarations

Ethics Approval and Consent to Participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for Publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of Data and Materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Competing Interests

The authors declare no competing interests in the research.

Funding

This research was not supported by any funding agency or institute.

Author Contribution

All authors contributed equally to this research.

References

- [1] F. K. Chan and J. Y. Thong, Acceptance of agile methodologies: A critical review and conceptual framework, Decision support systems. 46(4), 803-814(2009).
- [2] S. Munawar, R. Yousaf, and M. Hamid. Extended Scrum process model using software reliability engineering concerns Journal of Information Communication Technologies and Robotic Applications, 1-10(2018).
- [3] M. Hamid, F. Zeshan, A. Ahmad, F. Ahmad, M. A. Hamza, Z. A. Khan, et al. An intelligent recommender and decision support system (IRDSS) for effective management of software projects, IEEE Access. 8, 140752-140766(2020).
- [4] F. Hamid, A. Ahmad, and E. Aimeur. Factors contributing in failures of software projects," International Journal of Computer Science and Network Security, 19, 62-77(2019).

- [5] K. Krippendorff. The cybernetics of design and the design of cybernetics, *Design Cybernetics: Navigating the New*. 119-136(2019).
- [6] P. Ravi Shankar and M. G. Narayana. Developing agile teams for project execution: A cybernetic approach, *Global journal of flexible systems management*. 14, 33-43(2013).
- [7] M. Hamid, F. Zeshan, A. Ahmad, S. Munawar, E. Aimeur, S. Ahmed, et al. An intelligent decision support system for effective handling of IT projects. *Journal of Intelligent & Fuzzy Systems*. 38, 2635-2647(2020).
- [8] R. Espejo and R. Harnden. The viable system model. *Systems practice*. 3, 219-221(1990).
- [9] J. Żywiołek, E. R. Tucmeanu, A. I. Tucmeanu, N. Isac, and Z. Yousaf. Nexus of transformational leadership, employee adaptiveness, knowledge sharing, and employee creativity. *Sustainability*. 14, 11607(2022).
- [10] M. Lalancette and V. Raynauld. The power of political image: Justin Trudeau, Instagram, and celebrity politics. *American behavioral scientist*. 63, 888-924(2019).
- [11] N. Dragano and T. Lunau. Technostress at work and mental health: concepts and research results. *Current opinion in psychiatry*. 33, 407-413(2020).

Received: 14 Sep 2024, **Revised:** 15 Jan 2025,

Accepted: 15 May 2025, **Available online:** 18 May 2025.



© 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.