Journal of Advanced Research in HR and Organizational Management
Volume 10, Print Issue 1 - 2023, Pg. No. 18-24
Peer Reviewed Journal

Review Article

Risk Management Framework Review For Large Scale Scrum

Vivan Singh

Student, Department of management, School of Management Sciences, Khushipur, Varanasi, Uttar Pradesh.

INFO

E-mail Id:
vivansingh54@gmail.com
Orcid Id:
https://orcid.org/0009-0000-7736-5073

How to cite this article:
Singh V. Risk Management Framework Review For Large Scale Scrum. *J Adv Res HR Organ Mgmt* 2023; 10(1): 18-24.

Date of Submission: 2023-01-27
Date of Acceptance: 2023-02-30

ABSTRACT

Project risk management is an essential component. The main goal of risk management is to identify hazards early on in the project and take the appropriate actions to reduce them. In the realm of software development, it is critical and vital to effectively prioritise risk that arises. One of the key objectives of using agile development methodologies is to reduce risk, which results in more effective and successful information systems. Agile approaches to distributed software development are being adopted by organisations to generate high quality work in a shorter amount of time. Many software-based projects have recently transitioned to dispersed agile development projects. Due to the fact that these projects are spread out over a large-scale area, cost savings and proximity to the market are their main benefits. The creation of large- scale scrum projects has special risks for risk management, particularly the obstacles posed by team collaboration when there is no set method for communication or cooperation. A key component for the success of big scrum teams is team collaboration and the sharing of crucial information. Therefore, in a large- scale scrum setting, it is crucial to develop a dynamic strategy that promotes team communication and collaboration. Therefore, we will examine a paradigm for risk management in large-scale scrum employing outer meta data requests in this study. This framework aims to coordinate requests from several teams. As a result, it combats hazards and threats to project completion as well as the lack of team collaboration. Additionally, it makes the sharing of team information, experience, and abilities easier. Two distinct case studies were examined using the suggested methodology. The proposed framework's viability was shown through its implementation and evaluation.

Keywords: Software Product, Metadata, Risk Management Software Development Life Cycle, Agile, Security Risk Assessment, Distributed Agile Development, Large Scale Scrum;

Introduction

Software development is one of the biggest industries on the planet. There are many software projects in development, ranging in size, expense, and complexity. The Software Development Life Cycle (SDLC) is

imposed on the process of developing a software product by a software organisation as a framework or approach.[1]

The steps involved in the standardised software development life cycle (SDLC) include issue attribution, project viability analysis, requirement specification, design, development, testing, deployment, and maintenance.

One of the largest industries on the planet is software development. Many software projects of various sizes, costs, and complexity are now under development. A software organisation imposes the Software Development Life Cycle (SDLC) as a framework or approach on the process of generating a software product.[2] Issue attribution, project viability analysis, requirement specification, design, development, testing, deployment, and maintenance are all steps in the standardised software development life cycle (SDLC).[3] Understanding where to integrate risk management at a fundamental level is crucial before understanding how to expand Agile and risk management. Risk management must be a part of every project from the beginning, during development, and at the end. Let's take a closer look at the particular stages where Agile risk management is crucial.[4]

Identification of Risk

When the project's stakeholders have assembled to decide on a lengthy list of requirements, it is time to assess the risks associated with the project.[5]

Risk Planning

Following the identification of known hazards during the initial planning session, stakeholders establish an action plan (or a contingency plan) to decrease the risks that have been recognised. After that, they create a project risk register that is updated as new risks are identified.[6]

To Monitor the Risk Involved

Risk is tracked at every stage of the procedure. This refers to keeping an eye on the risks that were previously identified and updating the Project Risk Register with how they are being handled (and, ideally, closed) as well as any new risks that appear along the way.[7]

To Review the Risks

The Project Risk Register is updated after the project's conclusion to indicate how risks were handled throughout the project. This can be investigated and evaluated to determine lessons learnt and the best practises moving ahead.[8]

Risk Management on Iterative level

It's imperative to figure out how to apply the same attitudes and behaviours to iterative or Scrum processes (depending on which Agile framework you use in your organisation) after project-level risk management procedures are in place. Here are a few examples of phases where risks should be taken into account and which, despite taking place on a smaller scale, parallel events that take place at the project level.[9]

Sprint planning: It is carried out during the planning session at the start of an iteration.

Daily scrum: i.e., as iterations are being executed, progress is being checked, and project impediments are being discussed

Sprint review - It is the ideal time to examine risks while debating the current iteration that could have an impact on the backlog.

Retrospectives in the sprint: It's based on what risks arose during the previous iteration, how did you handle them, and what could be handled more effectively going forward.[10] Organisations need to implement effective knowledge management practises. Analysis of prior studies has shown that there are some problems with collaboration between team members working in different locations that prevent information from being shared. As a result, issues like outdated documentation and knowledge vaporisation exist. Large-scale Scrum projects enable Scrum team members to operate from a range of remote locations, maximising the advantages of the Scrum methodology. LeSS has a subtype called DAD.[11] The use of the Scrum methodology in large-scale scrum projects is viewed as a significant source of evolving risks and raises a number of challenges. These difficulties were related to the daily Scrum meetings, which focused on collaboration, engagement, and customer communication.[12]

The Major issues Come From

- physical distances that make communication difficult,
- ineffective team coordination,
- disagreement s that occur because of needs between the several product owners and the development team.
- cultural differences also play a part that hinders the effective communication.[13]

As a result of these issues, LeSS now has a number of hazards. The potential outcomes are divided into categories, and each group has a number of risk factors (RF).

The RFs listed below significantly influence LeSS:

Communication: No set procedure exists for the teams to collaborate and communicate.[14]

1. **Coordination and cooperation:** LeSS's circumstances may make it difficult to follow team cooperation guidelines suggestions and comments that come from software development projects that are iterative and collaboratively gathered by the teams can lead to software that is not aligned.[15]
2. One of the most crucial responsibilities in project management is risk management. Project management activities include assessing risk to classify it into low, moderate, and high levels, responding to risk to take appropriate action to reduce its effect, and monitoring risk to update the risk strategy.[16] Several of these actions are put into practise through the risk management. In order to achieve project objectives as efficiently as feasible, IT project management activities include managing the structure of the various project phases. Consequently, it is essential to integrate all of these management paradigms into a single model, such as the agile framework approach.[17,18]
3. **SDLC :** Planning, analysis, product design, further implementation, and testing are some of the aspects that must be completed within the software development process. Agile principles highlighted an individual's involvement in each stage of the SDLC, which makes it difficult for LeSS teams to evolve.[19,20]

The following concise list of concerns might be used to convey the suggested framework's goal:

Establish a formal centralization-based coordinating stregy: Each and every scrum master on the sender's side receives requests from each team and shares them with the scrum masters on the receiver

side. When they receive requests, they collaborate with their own teams to generate responses, which they subsequently turn in to the Scrum master of the sender.[21] Each and every request point in this context consists of a request shared with the other side, and every request statement that is generated could be classed into a certain attribute regarding the risk factors. As a result, it has the ability to account for every risk factor related to a certain request.

Risk assessment: To analyse the request points, each team assigns a reward value to each request point.

Each team receiving approved responses to each request's key components is the fundamental risk reduction technique.

Through central management, risk is tracked and controlled, allowing Scrum masters to keep track of new risks and how they are being addressed. These responses are noted in the outer request characteristics of the meta-data for all large-scale scrum teams.

Because of this, the Scrum masters are able to manage these risks and ensure that newly discovered hazards are covered.

Information exchange and experience swapping: The team's increased learning process is aided by the creation of a knowledge repository using the results of the actual coordinating process.[22]

Literature Review

Risk management based on distributed agile development has been the subject of numerous research. To identify the risk issues connected to distributed agile development projects, some of them also propose a variety of innovative agile risk management methodologies. New methods for risk management in distributed agile development-based projects are established by other studies.Researchers have focused on developing frameworks for widespread use in order to address the problems related to knowledge sharing in the context of remote teams. It is mentioned in[23,24] that dangers associated with distributed agile development have been identified, categorised, and given numerical rankings.

For defining risks associated with distributed agile development, the following three phases were advised in: risk definition and categorization, expert validation of the risk definition stage, and risk prioritisation based on importance.

[25] offers a strategy for identifying risks that is centred on three goals: accelerating projects while retaining quality and affordability.

The risk management for agile approaches tool was used in [26] to divide the list of 128 risk management practises into 48 subcomponents, and subsequently into five additional components.Large-scale agile frameworks indicate that businesses face a number of challenges.[27] Some of the older studies relied on gathering scanty data and getting suggestions for risk management methods from risk management experts or from the analysis of earlier data surveys. There have been some concerns raised concerning the accuracy of the data due to the fact that the majority of the information was acquired by individual human observations. Due to the absence of dynamic risk management, some of these studies suffered. Therefore, updating the data and regulating it have been done by hand as is normal for risk management methods. In contrast to past studies, this research recommends a comprehensive fix for the problems associated with distributed agile development. Review of a risk management paradigm for large- scale agile projects for risk metadata outer requests is presented in this study. There are several issues with the past investigations. Each of the prior studies focused on identifying a solution to a specific problem that distributed teams encounter. But a comprehensive risk-reduction strategy must be developed in order to lessen the hazards that distributed teams encounter. This is especially true given the diversity of distributed agile development-related challenges' beginning causes and sources.

Communication, collaboration and coordination, project management, and the suggested framework are incorporated into the large-scale scrum organisation in [1] in order to handle the four risk factors in large-scale scrum development. The proposed approach enables the big size scrum remote team to share expertise while also enhancing team experience.The suggested structure helps the Scrum master manage outside demands among the large size scrum distributed team and is implemented on two projects for a firm.[28]

Large Scale Scrum Process

In addition to integrating several processes, this methodology contains a framework for managing large software-based products. The Scrum management methodology is used to manage these models in the software development process. The stages of the Scrum life cycle are depicted in Fig. 1. Each stage of the sprint period—which is normally between two and four weeks—is referred to as a "sprint." It hinges on five brief ceremonies that each last a few minutes. If their ceremony isn't finished, they risk missing their chance to conclude a project.

Scrum, as a management framework, provides the capacity to continuously monitor the outcome and identify any threats.

Scrum teams are versatile groups with members who are proficient in all facets of development.

Without a leader, Scrum teams can also self-organize and choose the optimal strategy for completing their work.

The Scrum methodology is suitable for small to medium sized teams and projects. Scrum is rapidly being employed in large-scale projects, particularly those with many sites, coupled with numerous distributed and remote teams. The main flaw with Scrum is its daily ceremony meetings, which are difficult to conduct in large-scale projects and necessitate in-person meetings.[29]

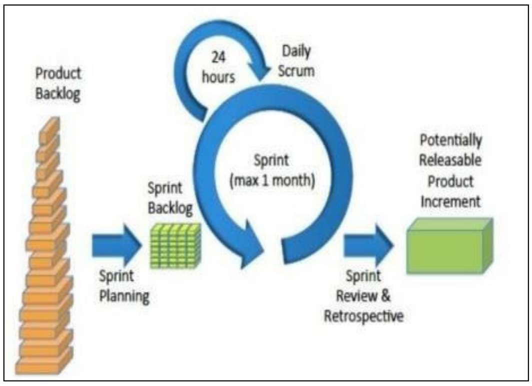

Figure 1.Stage of the Sprint Period

The product owner on the customer's end creates the product backlog, which is subsequently divided into groups of distinct, essential functions as part of project planning. A group of user stories are created for sprint planning using these things provided by the product owner. Each user story describes who uses it, what it does, and how much it means to the user.

The LeSS process management has allowed for the resolution of this LeSS limitation. Figure 2 depicts the LeSS workflow, and the following is a summary of how a LeSS project develops:

1. In order to fulfil the user interface and user acceptance tests, as well as the job estimating time, one or more user stories from the scheduled sprint are assigned to each feature team in a specific area. They then build a work plan and a sprint backlog.[30]

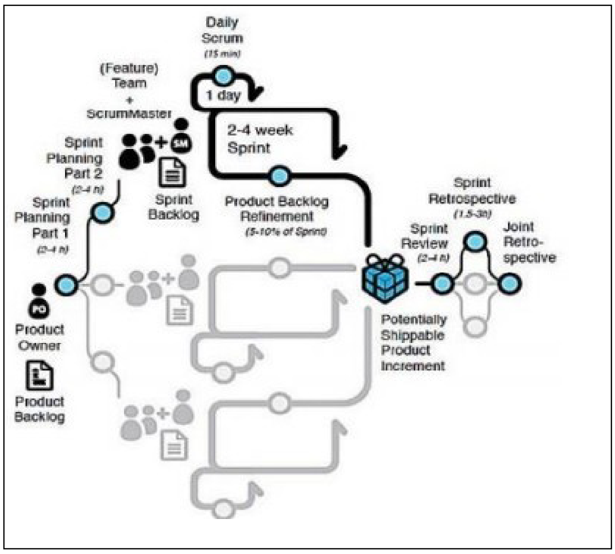

Figure 2.Depicts the Less Porkflow

2. Each day after the start of the sprint development, the feature teams at each location communicate with the scrum master. Making sure the team produces value, supporting in the development of a self-organizing team, and removing roadblocks are the responsibilities of the scrum master.
3. After the sprint development is finished, the Scrum master builds a demo to go over the created function with the development team. At each location, the product owner oversees the integration and thorough testing of sprint user stories.

The Scrum masters from each site then incorporate the results of their separate retrospective meetings during the subsequent retrospective meeting.

Metadata Management

This management helps the project manager complete all duties associated to the project by employing data attributes. These data attributes aid in the coordination and sharing of data through inquiries and conversations among the project locations. The information that has been gathered for these traits is helpful in the decision-making process.

In order to improve team comprehension, the results of this stored coordination are also communicated. Any organisation can choose which data qualities are pertinent to them while managing or coordinating a team.

LeSS serves as a kind of DAD project for this study's use of metadata management. The graphic shows metadata management as the process that takes place between the two LeSS sides. The LeSS team has been able to coordinate and control the management process thanks to the exchange of request points among the dispersed team members.

Two different kinds of metadata attributes required are as follows:

1. Request date, point number, risk factor request description, point reward ratio, and sprint number are some examples of request metadata attributes.
2. Reply metadata attributes: The reply status, reward ratio, reply point total, reply description, reply date, reply content acceptance, reply period status, and reply point are some of these features. Each Scrum master at a site has a responsibility to send or receive the traded request points. He also gives the metadata to their featuredteam (figure 4).

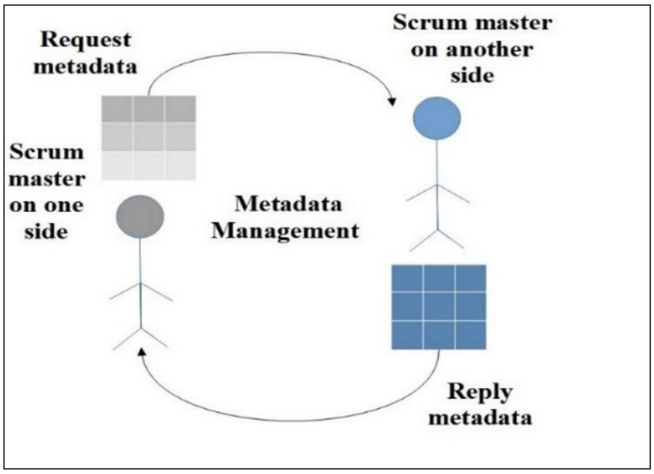

Figure 3.Kinds of Metadata Attributes

Large Scale Scrum's Metadata Ouer Request Risk Management Framwork

The suggested framework for Large Scale Scrum is depicted in this figure 4 by its structure. It was used to work on multiple projects in three different locations. At each location, there are two positions: the feature team and the Scrum master.

While locations A is for the sender's side, locations B and C are for the receiving or receptor side. The two basic models that make up the framework are the model of collaboration and coordination and the model of information exchange.

Collaboration and Coordination Model

The respective model is responsible for collaboration and coordination between the sender's and receiver's sides by exchanging requests and responses for the shared responsibilities.

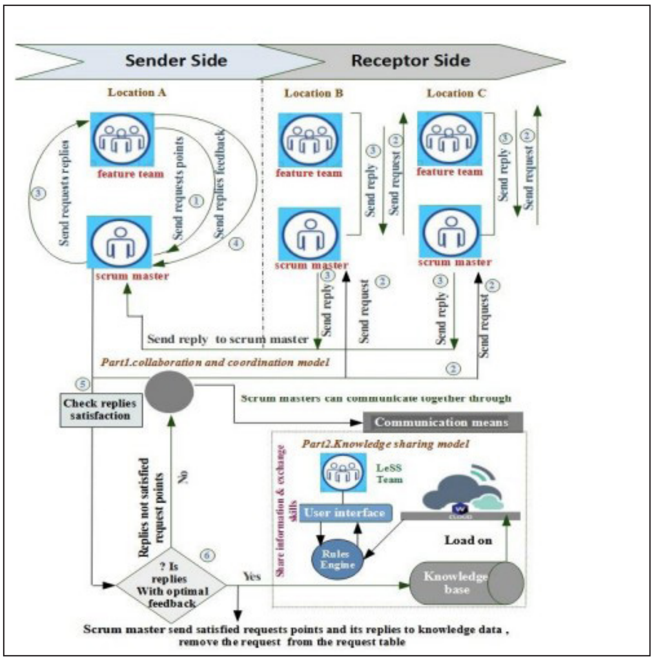
Sender Side
Receptor Side
Location A
Location B
Location C
feature team
scrum master
Send requests replies
Send requests points
Send replies feedback
Send reply
Send request
Send reply to scrum master
Part1.collaboration and coordination model
Check replies satisfaction
Scrum masters can communicate together through
Communication means
Part2.Knowledge sharing model
LeSS Team
User interface
Rules Engine
Load on
Knowledge base
Share information & exchange skills
Replies not satisfied request points
No
? Is replies With optimal feedback
Yes
Scrum master send satisfied requests points and its replies to knowledge data , remove the request from the request table
1
2
3
4
5
6

Figure 4.Basic Models That Make up the Framework Are the model of Collaboration and Coordination and the Model of Information Exchange.

Knowledge Sharing Model

The main objective of the aforementioned paradigm is to encourage communication and skill-sharing among the members of the remote team. This might be accomplished by gathering data from the requests and responses to the data that the large-scale scrum teams communicate. This information is impacted by the coordinating process. The circumstance for developing the knowledge repository is shown by one of the flow diagram's steps. If a request reply wins the top prize, the scrum master will send the complete request and its replies to the knowledge sharing repository for the large scrum team to use.

The findings show that among the allowed answers, dangers are covered with 161 points out of a possible 168. 31% of these are risks connected to the SDLC, 30% are hazards connected to collaboration and coordination, and 26% are risks connected to project management. Last but not least, 13% represents the risk of outcomes as a result of team misunderstanding. The suggested methodology successfully addresses 95% of the requests put forth by the teams.

The obtained results show that two elements, including (i) the count of reply points per request in proportion to the number of request points, the number of reply points per request rises.

(ii) The number of episodes per request – As The learning process moves along more quickly and reliability is increased with fewer episodes per request.

The Large-scale scrum teams experience would quickly enhance.

Conclusion

A big Scale Scrum project manages a variety of challenges and is a type of distributed agile development project. In-depth suggestions for formal coordinating strategies based on centralization are provided in this study. The suggested framework has been used by the large-scale scrum organisation. As a result, it successfully lowers the risk factors related to LeSS, and highly centralised coordination has been formed. Because of the excellent coordination that results in knowledge and information sharing among team members, the large size scrum distributed team succeeds in increasing team experience and adopts an agile mentality. This study's methodology for reducing DAD risks in large-scale scrum is described, along with how this framework might aid a distributed team in developing skills.

The group is successful in achieving openness while operating under scattered agile settings.

References

1. Rehab Adel , Hany Harb , Ayman Elshenawy Faculty of Engineering, Al-Azhar University, Cairo- Egypt,Faculty of Engineering and Technology, Egyptian, Chinese University, Cairo- Egypt, " A Risk Management Framework for Large Scale Scrum using Metadata Outer Request Management Methodology"
2. Mahnaz Afshari, Taghi Javdani Gandomani "A novel risk management model in the Scrum and extreme programming hybrid methodology" DOI: http://doi.org/10.11591/ijece.v12i3.pp2911-2921
3. Suprika Vasudeva Srivastava, Urvashi Rathod b, "A risk management framework for distributed agile projects", Information and Software Technology Vol 85, pp. 1-15, 2017.
4. Mohammad Shameem, Rakesh Ranjan Kumar, Chiranjeev Kumar, Bibhas Chandra, Arif Ali Khan, "Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process", J Softw Evol Proc.30: e1979,16, 2018.
5. Shrivastava, Suprika, Rathod, Urvashi, "A Goal-driven Risk management approach for Distributed Agile Development Projects", Australasian Journal of Information Systems, Vol 23, pp. 4-29,2019, Doi 10.3127/ajis. v23i0.1843.
6. Breno Gontijo Tavares, Mark Keil, Carlos Eduardo Sanches da Silva & Adler Diniz de Souza (2020),"A Risk-Management Tool for Agile Software Development", Journal of Computer Information Systems, Vol.61, ssue 4, Doi: 10.1080/08874417.2020.1839813
7. Sinha, Richa, Shameem, Mohammad, Kumar, Chiranjeev, "SWOT: Strength, Weaknesses, Opportunities, and Threats for Scaling Agile Methods in Global Software Development", Innovations in Software Engineering Conference, Jabalpur, India, pp.27–29, 2020, Doi: 10.1145/3385032.3385037
8. Ayesha Khalid, Shariq Aziz Butt, Tauseef Jamal, Saikat Gochhait, "Agile Scrum Issues at Large-Scale Distributed Projects: Scrum Project Development At Large", International Journal of Software Innovation, Vol. 8, No. 2, pp. 85–94,2020., Doi: 10.4018/IJSI.2020040106.
9. Vol. 8, No. 2, pp. 85–94,2020., Doi: 10.4018/IJSI.2020040106. Gabriela Castro Flores, Ana M. Moreno, Lawrence Peters, "Agile and Software Project Management Antipatterns: Clarifying the Partnership" in IEEE Software, vol. 38, no. 05, pp. 39-47, 2021, Doi: 10.1109/MS.2020.3001030.
10. Hui Yi Chiang, Bertrand M. T. Lin, "A Decision Model for Human Resource Allocation in Project Management of Software Development," in IEEE Access, vol. 8, pp. 38073-38081, 2020, Doi: 10.1109/ACCESS.2020.2975829.
11. Tavares, Breno, Silva, Carlos, Diniz de Souza, Adler, "Risk management analysis in Scrum software projects", International Transactions in Operational Research, pp.1–22,2017, Doi: 10.1111/itor.12401.
12. MARIA ILARIA LUNESU , ROBERTO TONELLI , (Member, IEEE), LODOVICA MARCHESI , (Member, IEEE), AND MICHELE MARCHESI , (Senior Member IEEE) "Assessing the Risk of Software Development in Agile Methodologies Using Simulation"
13. Gopalkrishna Waja, Jill Shah, Pankti Nanavati IT Department, KJSCE "AGILE SOFTWARE DEVELOPMENT" International Journal of Engineering Applied Sciences and Technology, 2021
14. Yehia Ibrahim Alzoubi, Asif Qumer Gill, "An Empirical Investigation of Geographically Distributed Agile Development: The Agile Enterprise Architecture is a Communication Enabler," in IEEE Access, vol. 8, pp. 80269-80289, 2020, doi: 10.1109/ACCESS.2020.2990389. Apoorva Srivastava, Sukriti Bhardwaj, Shipra Saraswat, "SCRUM model for agile methodology," 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 864- 869, Doi: 10.1109/CCAA.2017.8229928.
15. Muhammad Hammad, Irum Inayat, "Integrating Risk Management in Scrum Framework," 2018 International Conference on Frontiers of Information Technology

(FIT), 2018, pp. 158-163, Doi: 10.1109/FIT.2018.00035

16. Shrivastava, Suprika, Rathod, Urvashi, "A Goal-driven Risk management approach for Distributed Agile Development Projects", Australasian Journal of Information Systems, Vol 23, pp. 4-29,2019, Doi 10.3127/ajis. v23i0.1843.
17. Gu¨nther Ruhe, Claes Wohlin,"Software Project Management in a Changing World", ISBN 978-3-642- 55034-8 ISBN 978-3-642-55035-5 (eBook)- Doi:10.1007/978-3-642-55035-5.
18. Saskia Bick, Kai Spohrer, Rashina Hoda, Alexander Scheerer, Armin Heinz, "Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings," in IEEE Transactions on Software Engineering, vol. 44, no. 10, pp. 932-950, 1 Oct. 2018, Doi: 10.1109/TSE.2017.2730870.
19. Hamzane Ibrahim, Belangour Abdessamad,"Project Management Metamodel ConstructionRegarding IT Departments", (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 10, No. 10, 2019.
20. F S Rahayu, T Indrawan, S Kamarudin, "Risk Mitigation Strategies in Implementing Scrum Framework for Internet-Based IT Companies in Indonesia", Journal of Computer Information Systems", Indonesian Journal of Information Systems (IJIS)), Vol. 3, No. 1, ISSN 2623-0119, E- ISSN 2623-2308, August 2020.
21. Breno Gontijo Tavares, Carlos Eduardo Sanches da Silva, and Adler Diniz de Souza, "Practices to Improve Risk Management in Agile Projects", International Journal of Software Engineering, and Knowledge Engineering, Vol. 29, No. 3, pp. 381–399,2019, Doi: 10.1142/S0218194019500165. [22] Rizwan Qureshi, Mohammed Bashiri, Ahmad A AL Zahrani, "Novel Framework to Improve Communication and Coordination among Distributed Agile Teams", I.J. Information Engineering and Electronic Business, Vol.4,
22. No. 3, pp. 16-24, 2018, Doi: 10.5815/ijieeb.2018.04.03
23. T. E. Abioye, O. T. Arogundade, S. Misra, A. T. Akinwale, and O. J. Adeniran, ''Toward ontology-based risk management framework for software projects: An empirical study,'' J. Softw., Evol. Process, vol. 32, no. 12, Dec. 2020, Art. no. e2269.
24. M. Asif and J. Ahmed, ''A novel case base reasoning and frequent pattern based decision support system for mitigating software risk factors,'' IEEE Access, vol. 8, pp. 102278–102291, 2020.
25. D. Hilčíková, M. Dohnanská, D. Lajčin, and G. Gabrhelová, ''Agile project management as one of the critical success factors in project risk management in machinery industry in Slovakia,'' Int. J. Econ. Financ. Issues, vol. 1, no. 1, pp. 37–54, 2020.
26. V. S. V. M. R. Mittal, ''Risk analysis in software cost estimation: A simulation-based approach,'' Turkish J. Comput. Math. Educ. (TURCOMAT), vol. 12, no. 6, pp. 2176–2183, Apr. 2021.
27. B. G. Tavares, C. E. S. da Silva, and A. D. de Souza, ''Risk management analysis in scrum software projects,'' Int. Trans. Oper. Res., vol. 26, no. 5, pp. 1884–1905, Sep. 2019
28. C. A. Thieme, A. Mosleh, I. B. Utne, and J. Hegde, ''Incorporating software failure in risk analysis—- Part 2: Risk modeling process and case study,'' Rel. Eng. Syst. Saf., vol. 198, Jun. 2020, Art. no. 106804.
29. B. Zhao, J. Cao, S. Jiang, and Q. Qi, ''An agent based simulation system for open source software development,'' in Proc. IEEE World Congr. Services (SERVICES), Oct. 2020, pp. 164–170.
30. T. Alencar, M. Cortés, N. Veras, and L. Magno, ''A proactive approach to support risk management in software projects using multi-agent systems,'' in Proc. 20th Int. Conf. Enterprise Inf. Syst., 2018, pp. 415–424.