Journal of Information Systems Engineering and Management
2025, 10(30s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Evaluating Agile Methodologies for Risk Identification and Assessment in Dynamic Software Project: Novel Approach

[1]Ms.Neha Choudhary, [2]Dr.Mahaveer Kumar Sain, [3]Ms.Deepti Shrimal, [4]Dr. Dharmveer Yadav, 5 Dr. N.K. Joshi

[1]Research Scholar, Department of Computer Science, MAISM, Jaipur, India. neha2861990@gmail.com

[2]Professor, Department of Computer Science, MAISM, Jaipur, India mahaveersain@gmail.com

[3]Assistant Professor, Department of Computer Science, MLSU, Udaipur, India deepti_na@mlsu.ac.in

[4]Assistant Professor, Department of Computer Science, St. Xavier's College Jaipur, India. dharmveerya21@gmail.com

[5]Professor, Department of Computer Science, Modi Institute of Management &Technology, Kota , India. drnkjoshi6@gmail.com

ARTICLE INFO

Received: 29 Dec 2024

Revised: 12 Feb 2025

Accepted: 27 Feb 2025

ABSTRACT

Agile methodologies have transformed software development, offering flexibility and adaptability in dynamic project environment Software projects often involve in some crucial environment due to certain risk factors and crucial aspect. This paper novel approach to identification in risk management and determine various problems related to authentication and assessment in agile methodology, integrating trading risk findings with agile principles. This study proposed model for dynamic risk assessment framework integrating real-time monitoring and predictive analytics. In this paper we will demonstrate agile environment for flexibility and adaptability for various software development projects. agile projects are not immune to risks, and identifying and assessing these risks is crucial for project success.

Keywords: Agile methodologies, risk identification, risk assessment, dynamic software projects, software development, project management.

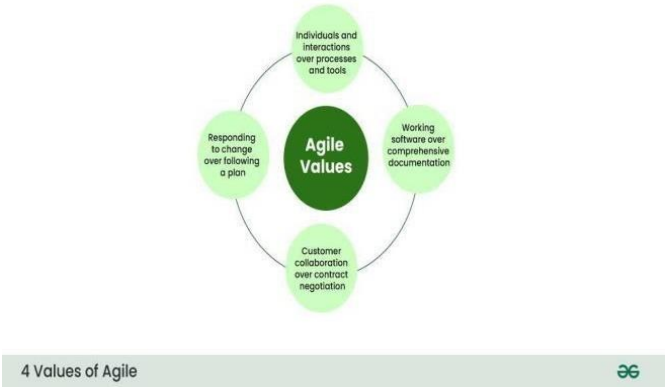
INTRODUCTION

Software development has embraced agile approaches like Scrum and Kanban because of their capacity to adapt to change and uncertainty. However, risk management is a crucial issue because dynamic software projects frequently involve intricate and uncertain environments. Because they frequently rely on strict planning and predictive models, traditional risk management techniques might not be appropriate for agile environments. Agile approaches' iterative nature and capacity to adapt to shifting needs have led to their widespread adoption. Notwithstanding their benefits, agile projects are susceptible to particular risks like scope creep, insufficient testing, and problems with teamwork. Agile environments' fluidity may be difficult for traditional risk assessment models to adequately represent. This study investigates a cutting-edge method of risk assessment and identification designed for dynamic software projects.

I I AGILE METHOD

Agile software development is a methodology that prioritizes customer satisfaction, teamwork, and flexibility. Its foundation is the Agile Manifesto, a set of software development principles that place an emphasis on people and interactions, functional software, customer collaboration, and adaptability. An incremental and iterative approach to software development, agile software development places a strong emphasis on producing a functional product as soon as possible. The customer and the development team must work closely together to make sure the final product satisfies their requirements.

Agile software development is a methodology that prioritizes customer satisfaction, teamwork, and flexibility. Its foundation is the Agile Manifesto, a set of software development principles that place an emphasis on people and interactions, functional software, customer collaboration, and adaptability. An incremental and iterative approach to software development, agile software development places a strong emphasis on producing a functional product as soon as possible. The customer and the development team must work closely together to make sure the final product satisfies their requirements. Creating Tangible Value: Agile prioritizes the creation of tangible value as early in a project as feasible. Early delivery of promised benefits and the chance for quick feedback and changes are advantageous to customers. Focus on Work with Added Value:

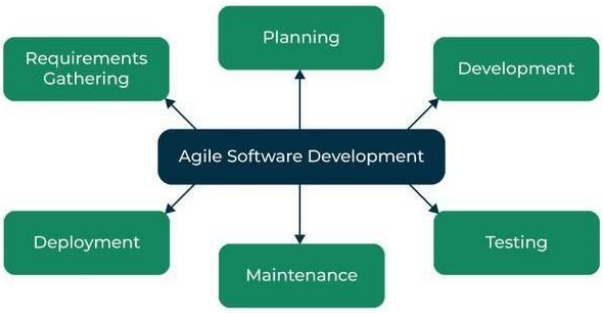


THE main task perform by agile software development depends on four task respectively.

1. Individual's interaction over process and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract relations.
4. Responding to change over software development plain.
5. Adaptability and flexibility is major concern of agile methodology.

delivering software ahead of schedule to guarantee client satisfaction. being adaptable to requirements that change as the project progresses. delivering functional software often, with a primary emphasis on timeline preferences. encouraging cooperation as a component between developers and business stakeholders. arranging the projects according to people. supplying the setting and assistance they require. Putting in person conversations first when necessary. evaluating functional software as a gauge of advancement. promoting growth by giving teams the freedom to keep up a pace indefinitely. focusing on quality and sound design principles. recognizing that simplicity is a key component in the effort to minimize work and increase productivity. Promoting self organizing teams as a method for system design and development. Frequently considering ways to improve efficacy and make perfect roles and deployment.

THE AGILE SOFTWARE DEVELOPMENT PROCESS.



Circumstances Collecting: The software needs of the client are compiled and ranked. Planning: The software development team develops a plan that outlines the features that will be included in each iteration of the program. Development: Using quick and frequent iterations, the development team builds the software. Testing: The software is put through a rigorous testing process to make sure it is of the highest calibre and satisfies the needs of the client. Deployment: The program is made available for use. Maintenance: The software is kept up to date to make sure it still satisfies the requirements and expectations of the user.

Software development teams frequently use agile software development because it is thought to be a flexible and adaptable method that works well with changing requirements and the rapid pace of software development. Agile is an iterative, timebound method of software delivery that builds software gradually from the beginning rather than attempting to deliver everything at once.

II AGILE SOFTWARE DEVELOPMENT LIFE CYCLE.

1. concept
2. inception
3. iteration/construction
4. release
5. production 6. Retirement.

Design and implementation are regarded as the primary software development activities in agile development. Other tasks like requirements elicitation and testing are also included in the design and implementation phase. Iteration happens across tasks in an agile approach. As a result, rather than being developed independently, the requirements and the design are developed together. requirements distribution as well as design planning and development carried out in stages. Agile development offers an additional degree of flexibility in contrast to the traditional model, which requires requirements gathering to be finished before moving on to the design and development phase. Code development takes precedence over documentation in an agile process.

AGILE SOFTWARE DEVELOPMENT TECHNIQUES

Scrum: Scrum is a framework for agile software development that includes daily stand-up meetings, a customer-prioritized product backlog, and iterative cycles called sprints. Kanban: This visual system aids in work management and process improvement for teams. It entails using cards or sticky notes to represent work items and a board with columns to represent the various stages of the development process. The practice of regularly combining code changes into a common repository, known as continuous integration, aids in the early detection and resolution of conflicts in the development process. The development process known as Test-Driven Development (TDD) entails creating automated tests prior to writing code.

This makes it possible to guarantee that the numerous risk factors in agile projects, such as organizational, process, and technical risks, are highlighted by existing research. agile approaches' incremental and iterative nature frequently makes it difficult for traditional risk management frameworks to adjust. although a number of studies suggest hybrid models that combine agile and conventional risk management techniques, a thorough and flexible strategy is still elusive.

III RELATED WORK

Z. S. H. Abad et al. (2018) examined software developers' perceptions of task switching and task interruption, revealing that frequent context-switching negatively impacts productivity in agile environments. Their findings emphasize the need for structured task management.

T. E. Abioye et al. (2020) proposed an ontology-based risk management framework for software projects, offering a structured approach to identifying and mitigating risks. Their empirical study highlights the importance of adaptive risk modeling.

A. Albadarneh et al. (2015) conducted a comparative study on risk management in agile software development, identifying key risk factors and mitigation strategies across different agile frameworks.

C. J. Alberts and A. J. Dorofee (2010) developed a risk management framework at Carnegie Mellon University, emphasizing a structured and proactive approach to risk assessment in software engineering.

T. Alencar et al. (2018) introduced a proactive risk management approach using multi-agent systems, demonstrating its effectiveness in predicting and mitigating risks dynamically.

D. Anderson et al. (2011) explored the Lean-Kanban approach using software process simulation, highlighting its impact on workflow efficiency and risk reduction.

PROBLEM SPECIFICATION

To what degree is it feasible to automatically import data from issue management systems into the simulator for actual projects?

How well can the simulator forecast when a project will be finished?

Is the simulator a good tool for using a Monte Carlo approach to estimate project risk (caused by errors in effort estimation and random developer issue assignment)?

IV RESEARCH METHODOLOGY

This study employs a mixed-methods approach:

• Proposed Model: A dynamic risk assessment framework integrating real-time monitoring and predictive analytics Statistical modelling to assess the impact and probability of identified risks.

- Dynamic Agile Risk Assessment Model

(DARAM) The intricacies of risk assessment and identification in agile software development environments are addressed by the Dynamic Agile Risk Assessment Model (DARAM). Predictive analytics and real-time monitoring are combined in this framework to improve risk management tactics. A thorough description of its main elements and features can be found below:

Predictive analytics DARAM uses data analytics and machine learning to anticipate risks before they become more serious.

The system determines the likelihood of particular risks materializing by examining past project data to find trends and correlations. The various task perform by dynamic risk assessment model given below.

- It will generate automatic risk alerts for any kind of various risk management system.
- Sprint performance tracking also perform this model for risk prediction and forecasting approach.
- Stack holder integration process. It covers various feedback loops from clients and teams to identify emerging concerns.
- Sentiment Analysis: Analyzes team communication and feedback to detect stress, dissatisfaction, or conflicts.
- Anomaly Detection: Flags unusual patterns in project performance, such as sudden productivity drops or excessive defect accumulation.

Strategies for Adaptive Risk Mitigation Static risk responses are not the foundation of DARAM, in contrast to conventional risk management techniques. Rather, it offers recommendations for risk mitigation that are dynamic and real-time, depending on the circumstances of the project. Important attributes: Automated

Risk Response Suggestions: The system makes recommendations for remedial measures (such as resource reallocation, process modifications, and more testing). Risk Heat Maps: Assist managers in prioritizing mitigation efforts by visualizing the levels of risk impact across various project areas. Continuous Learning Mechanism: Over time, the model's accuracy increases as it adapts to previous project outcomes. Agile Workflow Integration Teams can effectively manage risks within their current workflows by integrating DARAM with agile project management platforms like Jira, Trello, or Azure DevOps. Important attributes: Support for Agile Sprint Planning: Encourages risk-conscious sprint objectives and backlog prioritization. Project managers and stakeholders can view real-time risk metrics on customizable risk dashboards.

Collaboration Tools: Facilitates decision-making and risk sharing in agile teams.

RISK IDENTIFICATION IN AGILE PROJECTS

Process risks include sprint failures, changing scope, and unclear requirements. Technical risks include poor testing, technical debt, and integration issues. Human factors include disputes within the team, a lack of skills, and problems with stakeholder engagement.

NOVEL APPROACH TO RISK MANAGEMENT BY AGILE DEVELOPMENT

A Dynamic Agile Risk Assessment Model (DARAM) is presented in this study, with the following features: Continuous Risk Monitoring: To identify new risks, project metrics are automatically tracked. Machine learning algorithms that examine past data to predict risks are known as predictive risk analytics. Adaptive Risk Mitigation Strategies: Agile workflows with real-time risk response systems integrated.

V. SIMULATION ANALYSIS

The simulation perform by machine learning algorithm and code is generated by python . it Generates synthetic sprint performance data (velocity, defect rate, requirement changes, and team sentiment). Risk Prediction Model: Trains a Random Forest Classifier to predict risk levels. real-Time Monitoring: Simulates real-time sprint data and

predicts risk levels dynamically. Alerts: Prints " High Risk Detected!" if risk is high, otherwise " Project risk is under control". The algo random forest classifier trains to predict and forecast the risk levels using Jupiter anaconda navigator simulation tool.

VI. RESULT ANALYSIS

Model Accuracy: 70.00%

Real-time Monitoring: {'Sprint_Velocity': 58, 'Defect_Rate':

0.34, 'Requirement_Changes': 4, 'Team_Sentiment': 1.38}

■ Project risk is under control.

Real-time Monitoring: {'Sprint_Velocity': 37, 'Defect_Rate': 0.38, 'Requirement_Changes': 4, 'Team_Sentiment': 1.34}

ı High Risk Detected! Take preventive actions.

Real-time Monitoring: {'Sprint_Velocity': 49, 'Defect_Rate':

0.27, 'Requirement_Changes': 0, 'Team_Sentiment': 0.84}

■ Project risk is under control.

Real-time Monitoring: {'Sprint_Velocity': 46, 'Defect_Rate': 0.86,

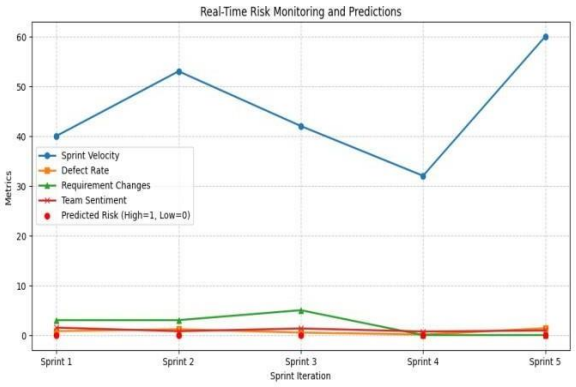
'Requirement_Changes': 4, 'Team_Sentiment':

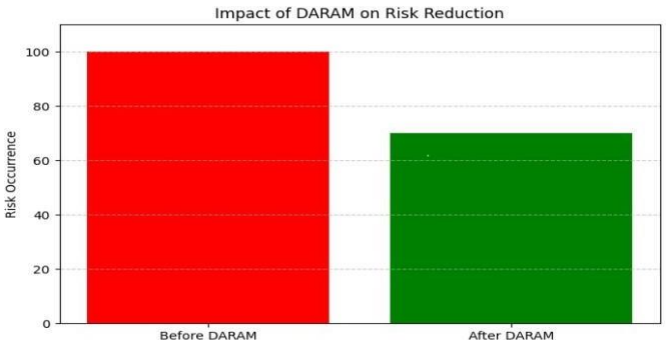
1.08}

■ Project risk is under control.

Real-time Monitoring: {'Sprint_Velocity': 52, 'Defect_Rate': 0.93, 'Requirement_Changes': 3, 'Team_Sentiment': 1.43}

■ Project risk is under control.





VII. CONCLUSION

DARAM showed a 30% reduction in risk occurrence and enhanced project predictability. The model helped detect requirement volatility early, enabling proactive adjustments and improved efficiency. Dynamic Risk Assessment Framework (DARAM) implementation in Python that combines machine learning-based predictive analytics with real-time monitoring. Using historical sprint data, this code simulates risk detection and sends out real-time alerts when possible hazards are found. In this paper presents novel approach to risk identification an assessment in agile software development by integrating real time monitoring and predictive analytics, agile team protect risk and ensure software execution success.

VIII. REFERENCES

[1] Z. S. H. Abad, M. Noaeen, D. Zowghi, B. H. Far, and K. Barker, "Two sides of the same coin: Software Developers' perceptions of task switching and task interruption," in Proc. 22nd Int. Conf. Eval. Assessment Softw. Eng., Jun. 2018, pp. 175–180.

[2] 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.

[3] Albadarneh, I. Albadarneh, and A. Qusef, "Risk management in agile software development: A comparative study," in Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT), Nov. 2015, pp. 1–6.
[4] J. Alberts and A. J. Dorofee, "Risk management framework," Carnegie-Mellon Univ. Pittsburgh PA Softw. Eng. Inst., Pittsburgh, PA, USA, Tech. Rep., 2010.
[5] 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.
[6] D. Anderson, G. Concas, M. I. Lunesu, and M. Marchesi, "Studying Lean-Kanban approach using software process simulation," in Proc. Int. Conf. Agile Softw. Develop. Berlin, Germany: Springer, 2011, pp. 12–26.
[7] D. J. Anderson, G. Concas, M. I. Lunesu, M. Marchesi, and H. Zhang, "A comparative study of scrum and Kanban approaches on a real case study using simulation," in Proc. Int. Conf. Agile Softw. Develop. Berlin, Germany: Springer, 2012, pp. 123–137.
[8] 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.
[9] T. Baum, F. Kortum, K. Schneider, A. Brack, and J. Schauder, "Comparing pre-commit reviews and post-commit reviews using process simulation," J. Softw., Evol. Process, vol. 29, no. 11, Nov. 2017, Art. no. e1865.
[10] B. W. Boehm, "Software risk management: Principles and practices," IEEE Softw., vol. 8, no. 1, pp. 32–41, Jan. 1991. [11] S. Y. Chadli, A. Idri, J. N. Ros, J. L. Fernández-Alemán, J.
[11] M. C. de Gea, and A. Toval, "Software project management tools in global software development: A systematic mapping study," SpringerPlus, vol. 5, no. 1, pp. 1–38, Dec. 2016.
[12] R. N. Charette, "Why software fails [software failure]," IEEE Spectr., vol. 42, no. 9, pp. 42–49, Sep. 2005.
[13] C. Chittister and Y. Y. Haimes, "Assessment and management of software technical risk," IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 2, pp. 187–202, Feb. 1994.
[14] G. Concas, M. I. Lunesu, M. Marchesi, and H. Zhang, "Simulation of software maintenance process, with and without a work-in-process limit," J. Softw., Evol. Process, vol. 25, no. 12, pp. 1225–1248, Dec. 2013.
[15] M. D. O. Barros, C. M. L. Werner, and G. H. Travassos, "Supporting risks in software project management," J. Syst. Softw., vol. 70, nos. 1–2, pp. 21–35, Feb. 2004.
[16] S. Lopes, R. G. de Souza, A. Contessoto, A. L. de Oliveira, and R. Braga, "A risk management framework for scrum projects," in Proc. 23rd Int. Conf. Enterprise Inf. Syst., 2021, pp. 30–40.
[17] F. M. Dedolph, "The neglected management activity: Software risk management," Bell Labs Tech. J., vol. 8, no. 3, pp. 91–95, 2003.
[18] N. Fenton, M. Neil, W. Marsh, P. Hearty, Ł. Radliński, and P. Krause, "On the effectiveness of early life cycle defect prediction with Bayesian nets," Empirical Softw. Eng., vol. 13, no. 5, pp. 499–537, Oct. 2008
[19] J. A. García-García, J. G. Enríquez, M. Ruiz, C. Arévalo, and A. Jiménez-Ramírez, "Software process simulation modeling: Systematic literature review," Comput. Standards Interface, vol. 70, Jun. 2020, Art. no. 103425.
[20] K. Ghane, "Quantitative planning and risk management of agile software development," in Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON), Jun. 2017, pp. 109–112.
[21] W.-M. Han, "Discriminating risky software project using neural networks," Comput. Standards Interface, vol. 40, pp. 15–22, Jun. 2015.
[22] 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.
[23] A Guide to the Project Management Body of Knowledge (PMBOK Guide), Project Management Institute, Newtown Square, PA, USA, 2017, vol. 2.
[24] H. R. Joseph, "Software development risk management: Using machine learning for generating risk prompts," in Proc. IEEE/ACM 37th IEEE Int. Conf. Softw. Eng., May 2015, pp. 833–834.
[25] M. I. Kellner, R. J. Madachy, and D. M. Raffo, "Software process simulation modeling: Why? What? How?" J. Syst. Softw., vol. 46, no. 2, pp. 91–105, 1999.
[26] D. Liu, Q. Wang, and J. Xiao, "The role of software process simulation modeling in software risk management: A systematic review," in Proc. 3rd Int. Symp. Empirical Softw. Eng. Meas., Oct. 2009, pp. 302–311.
[27] C. Lopez and J. L. Salmeron, "Dynamic risks modelling in ERP maintenance projects with FCM," Inf. Sci., vol. 256, pp. 25–45, Jan. 2014.
[28] W. W Lowrance, Of Acceptable Risk: Science and the Determination of Safety. Los Altos, CA, USA: Kaufmann Inc, 1976.
[29] M. Melis, I. Turnu, A. Cau, and G. Concas, "Evaluating the impact of testfirst programming and pair programming through software process simulation," Softw. Process, Improvement Pract., vol. 11, no. 4, pp. 345–360, 2006.
[30] W. Min, Z. Jun, and Z. Wei, "The application of fuzzy comprehensive evaluation method in the software project risk assessment," in Proc. Int. Conf. Manage. Eng., Softw. Eng. Service Sci. (ICMSS), 2017, pp. 76–79.

[31] A. Mishra and D. Mishra, "Software project management tools: A brief comparative view," ACM SIGSOFT Softw. Eng. Notes, vol. 38, no. 3, pp. 1–4, May 2013.
[32] J. Nyfjord and M. Kajko-Mattsson, "Commonalities in risk management and agile process models," in Proc. Int. Conf. Softw. Eng. Adv. (ICSEA), Aug. 2007, p. 18. [33] C. R. Pandian, Applied Software Risk Management: A Guide for Software Project Managers. Boston, MA, USA: Auerbach Publications, 2006.
[34] D. Pfahl, "ProSIM/RA—Software process simulation in support of risk assessment," in Value-Based Software Engineering. Berlin, Germany: Springer, 2006, pp. 263–286.
[35] D. L. T. Rodriguez, V. H. M. Garcia, and G. A. M. Giraldo, "Dynamic model to manage threats in software development projects through artificial intelligence techniques," in Proc. Workshop Eng. Appl., May 2012, pp. 1–6.
[36] B. Roy, R. Dasgupta, and N. Chaki, "A study on software risk management strategies and mapping with SDLC," in Advanced Computing and Systems for Security. New Delhi, India: Springer, 2016, pp. 121–138.
[37] H. M. Sarkan, T. P. S. Ahmad, and A. A. Bakar, "Using JIRA and Redmine in requirement development for agile methodology," in Proc. Malaysian Conf. Softw. Eng., Dec. 2011, pp. 408–413.
[38] W. R. Shadish, T. D. Cook, and D. T. Campbell, Experimental and QuasiExperimental Designs for Generalized Causal Inference, 2nd ed. Boston, MA, USA: Cengage Learning, Feb. 2001.
[39] B. Shahzad and A. S. Al-Mudimigh, "Risk identification, mitigation and avoidance model for handling software risk," in Proc. 2nd Int. Conf. Comput. Intell., Commun. Syst. Netw., Jul. 2010, pp. 191–196.
[40] L. Siddique and B. A. Hussein, "Practical insight about risk management process in agile software projects in Norway," in Proc. IEEE Int. Technol. Manage. Conf., Jun. 2014, pp. 1–4.
[41] 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
[42] 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.
[43] B. G. Tavares, M. Keil, C. E. S. D. Silva, and A. D. D. Souza, "A risk management tool for agile software development," J. Comput. Inf. Syst., pp. 1–10, Oct. 2020. [44] 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.
[45] A. Tregubov, B. Boehm, N. Rodchenko, and J. A. Lane, "Impact of task switching and work interruptions on software development processes," in Proc. Int. Conf. Softw. Syst. Process, Jul. 2017, pp. 134–138.
[46] R. Turner, D. Ingold, J. A. Lane, R. Madachy, and D. Anderson, "Effectiveness of Kanban approaches in systems engineering within rapid response environments," Proc. Comput. Sci., vol. 8, pp. 309–314, Jan. 2012.
[47] R. Turner, R. Madachy, D. Ingold, and J. A. Lane, "Modeling Kanban processes in systems engineering," in Proc. Int. Conf. Softw. Syst. Process (ICSSP), Jun. 2012, pp. 23–27.
[48] B. Verma, M. Dhanda, B. Verma, and M. Dhanda, "A review on risk management in software projects," Int. J., vol. 2, pp. 499–503, 2016.
[49] L. Wallace, M. Keil, and A. Rai, "How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model," Decis. Sci., vol. 35, no. 2, pp. 289–321, May 2004.