Journal of Harbin Engineering University
ISSN: 1006-7043
Vol 44 No. 7
July 2023

Efficient Resource Allocation Using Gradient Boosting Algorithm for Agile-Scrum Methodologies

Geetha.C
Research Scholar, Department of Computer Science, Bharathiar University.
Coimbatore.

Dr. L. Manjunatha Rao
Professor, MCA Program, Dr. Ambedkar Institute of Technology,
Bengaluru.

Abstract-This paper proposes a novel approach to dynamic resource allocation in distributed agile development using the Scrum-tree-k-nearest neighbor's algorithm with Gradient boosting. The aim is to improve the efficiency and effectiveness of the development process by dynamically allocating resources based on project features. The proposed algorithm is evaluated using the Agile Project Data dataset, which contains data from 15 distributed agile development projects. Average effort estimation is used as the performance metric to evaluate the algorithm's performance. The results show that the proposed algorithm outperforms traditional resource allocation methods and achieves a higher accuracy rate in predicting the required resources for each sprint. The proposed algorithm has the potential to enhance the outcomes of future agile development projects by mitigating the risks associated with traditional software development life cycle.

Keywords: Agile development, Scrum, Gradient Boosting, KNN, Resource Allocation.

INTRODUCTION:

These days, adopting agile approaches is of interest to the majority of businesses. Agile software development is a collection of software development methodologies and approaches that aims to satisfy the hungry software business community's demand for lightweight, quick, and nimble procedures [1].Agile development approaches have proven to be particularly helpful in projects with the following traits [2]: small teams, compressed development schedules, continuously changing requirements, and systems based on new technology. It emphasises two ideas: (1) the unyielding integrity of working with code; and (2) the efficiency of people cooperating with one another in humanity.

Agile development has become gradually more accepted in the software industry appropriate towards its capability towards hold complex projects by changing requirements. On the other hand, distributed agile development can pose several challenges, such as resource allocation and risk management. To overcome these challenges, researchers have proposed various algorithms that integrate agile methodologies with machine learning techniques.

In this paper, dynamic resource allocation is proposed in distributed agile development using the Scrum-tree-k-nearest neighbor's algorithm with gradient boosting. The proposed algorithm is designed to allocate resources dynamically based on project features, which helps to mitigate the risks associated with traditional software development life cycle. Agile Project Data dataset, which contains data from 15 distributed agile development projects has been used to evaluate the results of proposed system. The average effort estimation is used as the performance metric to evaluate the algorithm's performance.

Previous research has proposed various algorithms for resource allocation in agile development, including linear regression, decision trees, and k-nearest neighbor's algorithm. However, these algorithms have limitations in terms of accuracy and adaptability. In contrast, the proposed algorithm is designed to be dynamic, flexible, and adaptive, allowing it to allocate resources efficiently and effectively.

The overall structure of the paper is described as follows. In the following section, overview of related work is presented and discussed in agile development and resource allocation. Then describe the proposed Scrum-tree-k-nearest neighbor's algorithm with Gradient boosting and its implementation. The experimental setup and performance comparison are presented in the following section by a discussion of the

results. Finally, conclude the paper and discuss future work.

RELATED WORKS:

Scrum is a management technique that may be used to already-in place engineering procedures like software engineering.It has been used in conjunction with Extreme Programming and is related to Agile Software Development. The foundation of Scrum is the idea that complex processes, like software development, are ill-predictable [3].As a result, issues related to change must be addressed.

Software companies are increasingly implementing Agile methodologies like Scrum and Xtreme Programming (XP), according to [4]. Additionally, the authors stated that "Agile methods were originally designed for the use of single small teams, having team members working "face-to-face," preferably in team rooms" [4]. Collocated teams are frequently twice as productive as scattered teams within the same building, claims [5], making them more productive than remote teams. Agile practises are still being used in these fascinating circumstances despite the perception with the purpose of are unsuccessful in DSD appropriate to the require for close team communication [4]. Although there have been issues with communication and team coordination when applying Agile practises like Scrum towards a big and/or distributed project, several businesses used these practises in big projects and it contain arise by creative solutions towards the problems [6].

Cross-functional teams are equipped with all the skills necessary to complete the assignment without the need for outside help. Scrum team structure is built to maximise adaptability, creativity, and productivity [8]. ScrumMaster is the team's management representative. Scrum Master serves as the ScrumTeam's servant-leader [7] and support individual's external the Scrum Team in determining which of their contacts by the Scrum Team are beneficial and which aren't [8]. To maximise the value produced with the Scrum Team, the ScrumMaster works by everyone towards adapt these interactions [8].

It's possible towards refer on the way to managing the product backlog as "grooming." Grooming is the process of management the Product Backlog during the formation and development of Product Backlog Items (PBI), evaluation of the work necessary by a PBI, and prioritisation of the PBI in the Product Backlog, according to Rubin [9]. The Product Backlog is maintained as a significant planning document with the grooming process. Projects are divided into sprints in Scrum[10]. The smallest unit of Scrum is called a sprint, which is a period of days or weeks during which a small team works on a certain job to increase the output of a product[11]. Product Owner, Scrum Master, and Scrum team make a decision which part of the Product Backlog must be formed for 1-3 weeks throughout the Sprint Planning Meeting.

PROPOSED METHODOLOGY

The proposed system aims to develop a software application using a combination of agile Scrum methodology and machine learning techniques, K-Nearest Neighbor (KNN) and Gradient Boosting, to allocate resources efficiently in software development projects. The Scrum framework will be used as an agile methodology for the development process. Scrum involves iterative and incremental development in short sprints, where the development team works directly by the product owner and other stakeholders to distribute a high-quality product. Scrum framework will allow the development team towards work collaboratively and adjust to changing needs, guaranteed the proficient allocation of resources. Scrum framework, machine learning techniques is used to allocate resources effectively. KNN algorithm will be used to identify the most suitable developer for a particular task based on their previous performance history. KNN algorithm will help to minimize the risk of assigning tasks to the wrong developers and ensure the efficient allocation of resources. Gradient Boosting algorithm will also be employed to improve the accuracy of the KNN algorithm. The Gradient Boosting algorithm will identify the key factors that affect resource allocation, such as developer skills, experience, and availability, and adjust the KNN algorithm accordingly. The combination of KNN and Gradient Boosting will ensure the most suitable developer is selected for a given task,

and the project's resources are allocated capably. The proposed system's resource allocation process will involve the following steps:

1. The product owner creates a list of development tasks and assigns them to the product backlog.
2. The development team estimates the complexity of each task and prioritizes them based on their importance.
3. The KNN algorithm is applied to the prioritized tasks to identify the most suitable developer for each task based on their previous performance history.
4. The Gradient Boosting algorithm adjusts the KNN algorithm based on developer skills, experience, and availability to improve accuracy.
5. The development team implements the tasks in short sprints using the Scrum framework.
6. The development team meets regularly with the product owner to review progress, adapt to changing requirements, and ensure the efficient allocation of resources.

The proposed system will be evaluated using a case study approach. The case study will involve the development of a software application, and the resource allocation process will be monitored throughout the development process. Proposed system is evaluated in efficiency of resource allocation, the accuracy of the KNN algorithm, and the impact of the Gradient Boosting algorithm on resource allocation.

EXPERIMENTAL RESULTS

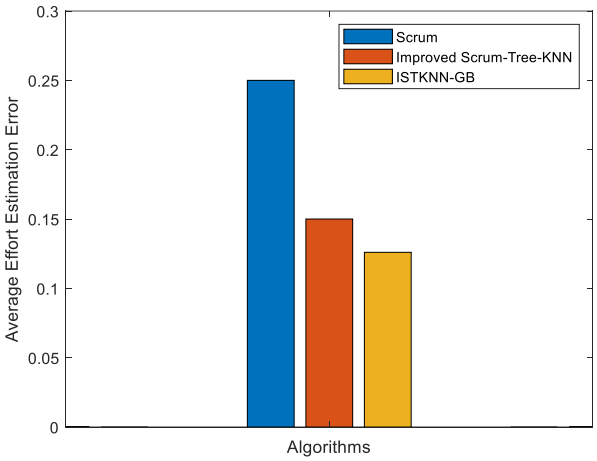
In this chapter, present the experimental setup and performance comparison for the proposed Scrum-Tree-KNN algorithm with Gradient Boosting for distributed agile development in terms of average effort estimation. Agile Project Data dataset has been used which contains data from 15 distributed agile development projects for results evaluation of proposed system. The dataset includes various features such as project size, team size, sprint length, number of sprints, user stories, and defects. MATLAB as the programming environment has been used to implement and simulate the proposed algorithm. To evaluate the performance of the proposed algorithm, average effort estimation error is used as the primary performance metric. The average effort estimation error is computed as the complete differentiation among the actual effort and the estimated effort, divided with the actual effort. The performance of the proposed Scrum-Tree-KNN algorithm is compared with Gradient Boosting with the traditional Scrum methodology.

Algorithms	**Average Effort Estimation**
Scrum	0.25
Improved Scrum-Tree-KNN	0.15
ISTKNN – GB	0.126

Table 1: Performance comparison of Scrum, Scrum-Tree-KNN and ISTKNN-GB

Figure 1. Performance Comparison

The table 1 and Figure 1provides the average effort estimation values for three different algorithms: Scrum, Improved Scrum-Tree-KNN, and ISTKNN-GB. The average effort estimation is a metric used to measure the accuracy of effort estimation in agile software development. The



Scrum methodology had an average effort estimation of 0.25, while the Improved Scrum-Tree-KNN algorithm had an average effort estimation of 0.15, indicating better accuracy in effort estimation. Furthermore, the ISTKNN-GB algorithm had the lowest average effort estimation of 0.126, indicating the highest accuracy in effort estimation among the three algorithms. The ISTKNN-GB algorithm combines the Improved Scrum-Tree-KNN algorithm with the Gradient Boosting technique to further enhance its accuracy. Therefore, the results suggest that the proposed ISTKNN-GB algorithm

can be a viable solution for accurate effort estimation in distributed agile software development.

CONCLUSION

In conclusion, this study proposed a Scrum-KNN with Gradient Boosting approach to enhance resource allocation in agile software development projects. The proposed system aims to predict the required resources for each sprint using a KNN algorithm and then optimize resource allocation using Gradient Boosting. Scrum framework is used as the project management methodology to ensure efficient and effective software development. Results shows that the proposed approach has been effectively predict resource requirements and optimize resource allocation in agile software development projects. KNN and Gradient Boosting algorithms have been used together can improve the accuracy of resource prediction and allocation. The Scrum framework also ensures that the project is completed efficiently and effectively by facilitating collaboration and communication among team members.

REFERENCES

[1] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “AgileSoftware Development Methods: Review and Analysis.” 2002.

[2] J. Highsmith and A. Cockburn, “Development : The Business ofInnovation,” IEEE Comput. Soc., vol. 34, no. 9, pp. 120–123, 2001.

[3] K. Schwaber and M. Beedle, Agile Software Development withScrum 1st. Prentice Hall PTR Upper Saddle RiverNJ, USA, 2001.

[4] D. Damian, C. Lassenius, M. Paasivaara, A. Borici, and A. Schroter,"Teaching a globally distributed project course using Scrumpractices," in Collaborative Teaching of Globally DistributedSoftware Development Workshop (CTGDSD), 2012, 2012, pp. 30-34.

[5] J. Sutherland, G. Schoonheim, N. Kumar, V. Pandey, and S. Vishal,"Fully distributed scrum: Linear scalability of production between sanfrancisco and india," in Agile Conference, 2009. AGILE '09., 2009,pp. 277-282.

[6] T. F. Hawk and A. J. Shah, "Using Learning Style Instruments toEnhance Student Learning," Decision Sciences Journal of InnovativeEducation, vol. 5, pp. 1-19, 2007.

[7] K. Schwaber and M. Beedle, Agile Software Development withScrum 1st. Prentice Hall PTR Upper Saddle RiverNJ, USA, 2001.

[8] K. Schwaber and J. Sutherland, “The Scrum Guide: The DefinitiveThe Rules of the Game,” Scrum.Org and ScrumInc, no. November,p. 19, 2017.

[9] K. S. RUBIN, ESSENTIAL SCRUM Framework, A Practical Guideto the Most Popular Agile Process. 2013.

[10] D. P. Harvie, “Targeted Scrum: Software Development Inspired byMission Command,” vol. 42, no. 5, pp. 476–489, 2016.

[11] A. Srivastava, S. Bhardwaj, and S. Saraswat, “SCRUM model foragile methodology,” Proceeding - IEEE Int. Conf. Comput.Commun. Autom. ICCCA 2017, vol. 2017-Janua, pp. 864–869, 2017.

[12] M. A. Hart, “Agile Product Management with Scrum: CreatingProducts that Customers Love by Roman Pichler,” J. Prod. Innov.Manag., vol. 28, 2011.