World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(CASE REPORT)

Check for updates

Case study on evolution performance of agile scrum software development life cycle for shopping cart applications

Akhtar Ali Junejo [1,*] and Sajida Memon [2]

[1] *Southeast University, Nanjing, 21000, China.*
[2] *Dawood University of Engineering and Technology, Karachi, 74800, Pakistan.*

World Journal of Advanced Engineering Technology and Sciences, 2023, 10(01), 085–090

Publication history: Received on 04 August 2023; revised on 12 September 2023; accepted on 14 September 2023

Article DOI: https://doi.org/10.30574/wjaets.2023.10.1.0254

Abstract

This paper presents the novel agile scrum methodology for the shopping cart product. The existing software development life cycle, such as waterfall, iterative methods, and spiral life cycles, has a larger failure ratio and cost factors when a connected component fails during the execution of a product. In this paper, we devise a novel agile scrum methodology where sprint backlog and product backlog process are demonstrated with the cost and failure efficiency. The role of the scrum master, standard deviation, mean and completion and task progression are illustrated in the methodology as shown in the result discussion.

Keywords: Agile Scrum; Shopping Cart; Sprint Backlog; Product Backlog; Scrum Master; Software Methodology.

1. Introduction

These days, the usage of software development life cycle (SDLC) has been increasing progressively to develop Internet of Things (IoT) applications [1]. SDLC consists of different phases: data gathering, analysis, implementation, deployment, execution, and testing [2]. Therefore, IoT applications such as digital healthcare, shopping carts, transport, and home-based automation require SDLC to meet end-user requirements during development [3]. The SDLC consisted of different methodologies. For instance, waterfall, spiral, iterative, and generic design IoT applications from requirement gathering to implementation [4]. Many developments have been made in the software development life cycle domain with the new components.

These studies suggested [5-12] waterfall and iterative model-enabled methodologies with different components. The components are correct requirements gathering, data analysis, normalization, implementation, deployment, and testing. Each component is depth and has various sub-components. For instance, testing sub-components are block box testing, gray box testing, and others. However, the existing SDLC supports all kinds of IoT applications. Regardless, all existing SDLCs still suffer from failure and long-co-joined components for IoT applications. These studies improved the current SDLC methodologies with the agile scrum. Agile scrum offers different elements such as product backlog, sprint backlog, scrum master, agile testing, and agile fog cloud networks. However, the existing studies do not apply these proposed agile methods to the IoT shopping cart application.

This paper presents the novel agile scrum methodology for shopping cart IoT applications. The proposed agile scrum methodology consisted of agile scrum components. We present sprint backlog, product backlog, starting tasks, completion tasks, and status of tasks in detail.

The paper is organized in the following way. Related work shows the existing SDLC and agile scrum methods. The methodology shows the proposed agile scrum performed in different components. The result discussion shows how the

*Corresponding author: Akhtar Ali Junejo

IoT shopping cart was implemented with the agile scrum. The conclusion and future work show the manuscript obtained results compared to existing studies and the future direction of the study.

2. Related Work

Many studies suggested agile scrum methodology-based solutions for different IoT applications with different numbers of components. The application portioning is based on agile scrum for healthcare and microservices, as suggested in [1]. However, the work suggested static agile scrum methodology in their studies based on cloudlet and microservices and incurred longer failure. These studies [2-5] suggested dynamic sprint backlog and deployment solutions of agile scrum for IoT applications. These studies suggested different testing mechanisms, such as black box and gray box, to improve the efficiency of agile testing in the product backlog. The primary objective of this research is to surmount the challenges that arise during the application of traditional methodologies, which prove particularly daunting as software progresses into the implementation phase. A major drawback of these traditional approaches is their inflexibility in accommodating changes to user requirements once the project has commenced from its initial stages. With software requirement specifications established and the project underway, implementing alterations based on new ideas becomes increasingly complex, necessitating modifications from the very foundation. Another significant issue pertains to the limited engagement of customers throughout the software development process. In response to these challenges, this research paper furnishes a solution delineating how software firms can effectively employ Agile Methodology to attain favorable outcomes. These studies [6-10] suggested fog and cloud agile scrum methodology with dynamic failure and services for healthcare and transport applications. These studies fixed the agile scrum team members; the product owner can change one service at a time. The dynamic, agile scrum with the team members for healthcare and transport applications enabled methods suggested in these studies [11-15]. These studies suggested dynamic, agile scrum solutions for off-healthcare applications. However, there is no agile scrum methodology for shopping carts with the dynamic changes after the deployment.

3. Proposed Agile Scrum Methodology

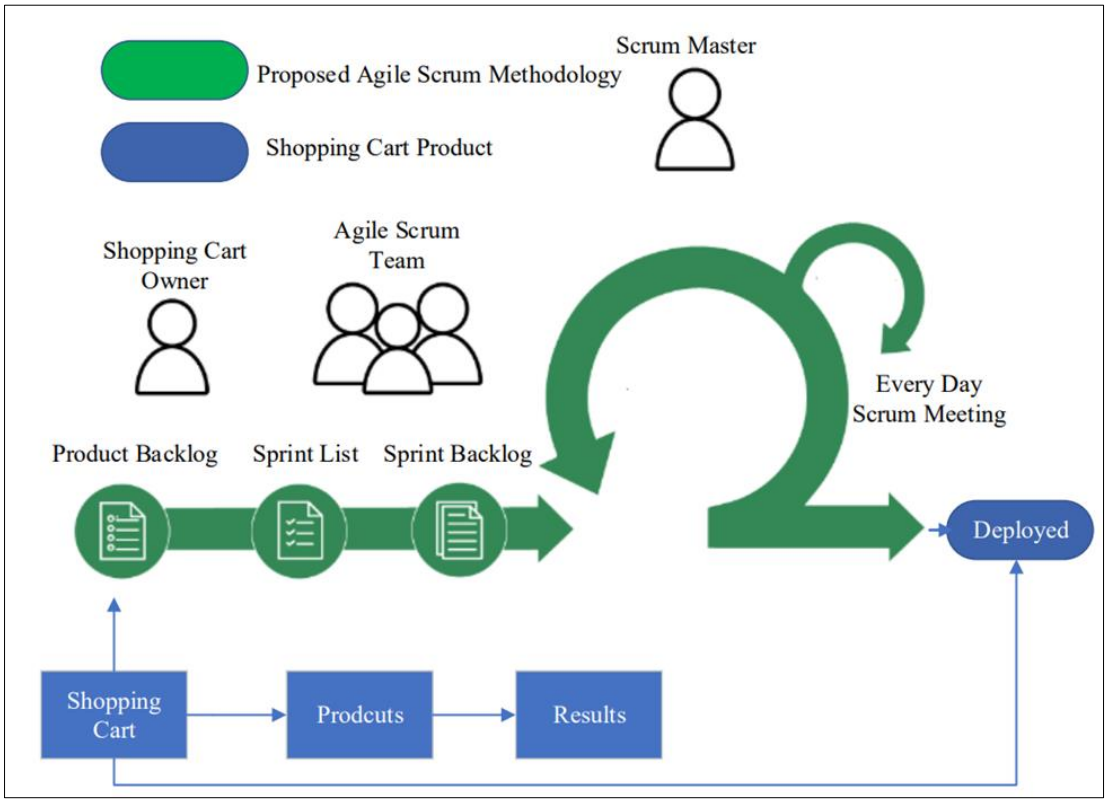

Figure 1 Proposed Agile Scrum Methodology for Shopping Product

The main aim of this research is to overcome the problems that occur while applying traditional methodology. These problems are very difficult to overcome when software reaches the implementation phase. One major problem of this methodology is that once you start from the initial stage in traditional methodology, there is no room for changing user

requirements. Software requirement specifications are built, and the project has started; it becomes difficult to make changes according to new ideas because changes would need to be made from the initial level. Another major problem is the lack of customer involvement during software development. To address all of these problems, this research paper provides a solution on how software firms can use Agile Methodology and achieve good results using Agile methodology. In this paper, we devise the novel agile scrum methodology for the shopping cart application with the different components, as shown in Figure 1.

The proposed agile scrum methodology has two main parts, such as shopping cart products, as shown in blue nodes in Figure 1. The green nodes show agile scrum components with different elements. The shopping cart has products and results elements to check the accuracy of the product, which belongs to the product owner. It means the failure and testing can be applied to the product at the runtime. The scrum master is the project manager who handles the product with the agile scrum team members. The product is divided into product backlogs, where priority-wise tasks are picked up as sprint lists. The sprint backlog shows the running tasks with the agile scrum team with the completion status. There is a day scrum meeting based on 24-hour duration from sprint list to sprint backlog until and unless all products are deployed successfully with the given requirements.

4. Result Discussion of Shopping Cart Agile Scrum

This case study contains four product backlogs, and the time is only two months. Also, sprint backlogs are divided Into 8 weeks. Each sprint backlog is divided into two weeks. The total Sprint backlog is eight. This case study shows how product backlog is shown in Table 1.

Table 1 Product Backlog

Accuracy increases sprint to sprint, and customer satisfaction	Priority	Credit hours	Resource	status
Layout of website	High	16 hours	Akhtar	Pending
Menu on layout	High	16 hours	Akhtar	Pending
Product display	High	16 hours	Akhtar	Pending
Product category wise	High	16 hours	Akhtar	Pending
Search Bar	Middle	8 hours	Akhtar	Pending
Ads	Middle	8 hours	Akhtar	Pending
Sidebar of menu	Low	8 hours	Akhtar	Pending

The above figure is the product backlog; this is also the master list the product owner has got from the customer. Now, the developer scratches the task from the product backlog to the sprint Backlog for committing to complete, as shown in Table 2.

Table 2 Sprint List

Task	Hours	%complete	%obstacles	Resource
Layout	8	50	10	Akhtar
layout	8	100	0	Akhtar
menu	8	100	0	Akhtar
product	8	100	0	Akhtar
category	8	100	0	Akhtar
Search	8	60	30	Akhtar
Search	8	90	99	Akhtar

After completing each task, it passes through testing, as shown in the below graph. Test Cases of the above tasks for unit testing are shown in Table 3. The revision product backlog can be revised at any time in the revised phases during the meeting with the scrum master team, as shown in Table 3.

Table 3 Sprint Backlog

Task	Condition	Expected Result	Actual result	status
Layout	IE Browser	Layout expansion	Layout just Slice	Fail
Layout	Chrome	Just Slice	Just Slice	Pass
Menu	IE Browser	Expansion layout	Just Slice	Fail
Product display with category	From database	All product	All Product	Pass
Search	All keywords	Search not all	Search All	Fail

Table 4 shows the sprint backlog after the revision and shows the completion products of end users in the software development life cycle as agile scrum, as shown in Table 4.

Table 4 Task Progress

Task	Layout %	Menu %	Product %	Search %	Login %
Planned Code	100	100	100	100	100
Actual code	90	90	90	90	90
Complete %	60	60	60	60	60
S-deviation	60	60	60	60	60
Mean	70	70	70	70	70

Table 4 and Table 5 show the standard deviation of complete and ongoing product tasks in the agile scrum with their status. The planned code, actual code, complete% with the standard deviation (S-deviation) with the completion mean of all tasks in agile scrum.

Table 5 Task Progress

Task	Cart %	Api %	Product %	Search %	Login %
Planned Code	100	100	100	100	100
Actual code	90	90	90	90	90
Complete %	71	71	71	71	71
S-deviation	70	70	70	70	70
Mean	72	72	72	72	72

Table 6 Status Completion Process

Task	Tasks	Mail	Registration	Validation	Authorize
Planned	100	100	100	100	100
code	90	90	90	90	90
Complete	72	72	72	72	72
Deviation	75	75	75	75	75
Mean	76	76	76	76	76

Table 7 Product Completed Status

Task	Payment	Add	Multi	Session	Admin
Planned	100	100	100	100	100
Actual	90	90	90	90	90
Complete	77	77	77	77	77
deviation	78	78	78	78	78
Mean	79	79	79	79	79

Table 8 Product Display Sprint List

Task	Cookies %	Shipmentuser %	Checking credit %	Master card %	Online transaction %
Planned Code	100	100	100	100	100
Actual code	90	90	90	90	90
Complete %	80	80	80	80	80
S-deviation	80	80	80	80	80
Mean	82	82	82	82	82

Table 9 Sprint Backlog Tested Results

Task	Orders	Accounts	Reporting	Users	Login %
Planned Code	100	100	100	100	100
Actual code	90	90	90	90	90
Complete	80	80	80	80	80
Sdeviation	88	88	88	88	88
Mean	89	89	89	89	89

Table 10 Product Backlog Completed with Results

Task	Tax %	Delivery	code %	Search	Help %
Planned Code	100	100	100	100	100
Actual code	100	100	100	100	100
Complete %	97	97	97	97	97
Deviation	95	95	95	95	95
Mean	97	97	97	97	97

These are result iteration to iteration and improving the results Iteration to iteration. Code planned: planning for task completion percentage. Actual code: some less percentage than original task Complete: how many developers complete % from original as shown in Table 5,6,7,8,9 and 10.

5. Conclusion and Future Work

This paper presented the novel agile scrum methodology for the shopping cart product. The result discussion showed that existing agile scrum methodologies have many failure ratios and cost factors issues when a connected component fails during the execution of a product. We presented agile scrum methodology, demonstrating the cost and failure efficiency with sprint backlog and product backlog process. The role of the scrum master, standard deviation, mean and completion and task progression are illustrated in the methodology as shown in the result discussion.We shown the shopping cart application connected with the agile scrum methodology in different components.

In future work, we will add the adaptive testing methodologies in the agile scrum for the shopping cart.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Lakhan, A., & Li, X. (2020). Transient fault aware application partitioning computational offloading algorithm in microservices based mobile cloudlet networks. Computing, 102(1), 105-139.

[2] Zayat, W., & Senvar, O. (2020). Framework study for agile software development via scrum and Kanban. International journal of innovation and technology management, 17(04), 2030002.

[3] Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11).

[4] Hema, V., Thota, S., Kumar, S. N., Padmaja, C., Krishna, C. B. R., & Mahender, K. (2020, December). Scrum: An effective software development agile tool. In IOP Conference Series: Materials Science and Engineering (Vol. 981, No. 2, p. 022060). IOP Publishing.

[5] Lakhan, A., Dootio, M. A., Groenli, T. M., Sodhro, A. H., & Khokhar, M. S. (2021). Multi-layer latency aware workload assignment of e-transport iot applications in mobile sensors cloudlet cloud networks. Electronics, 10(14), 1719.

[6] Khoso, F. H., Arain, A. A., Lakhan, A., Kehar, A., & Nizamani, S. Z. (2021). Proposing a novel iot framework by identifying security and privacy issues in fog cloud services network. Int. J, 9(5), 592-596.

[7] Lakhan, A., Khoso, F. H., Arain, A. A., & Kanwar, K. (2021). Serverless based functions aware framework for healthcare application. Int. J, 9(4).

[8] Albarqi, A. A., & Qureshi, R. (2018). The proposed L-Scrumban methodology to improve the efficiency of agile software development. International Journal of Information Engineering and Electronic Business, 11(3), 23.

[9] Garzaniti, N., Briatore, S., Fortin, C., & Golkar, A. (2019, March). Effectiveness of the scrum methodology for agile development of space hardware. In 2019 IEEE Aerospace Conference (pp. 1-8). IEEE.

[10] Shafiq, S., Mashkoor, A., Mayr-Dorn, C., & Egyed, A. (2021). A literature review of using machine learning in software development life cycle stages. IEEE Access, 9, 140896-140920.

[11] Akinsola, J. E., Ogunbanwo, A. S., Okesola, O. J., Odun-Ayo, I. J., Ayegbusi, F. D., & Adebiyi, A. A. (2020). Comparative analysis of software development life cycle models (SDLC). In Intelligent Algorithms in Software Engineering: Proceedings of the 9th Computer Science On-line Conference 2020, Volume 1 9 (pp. 310-322). Springer International Publishing.

[12] Sathio, A. A., Dootio, M. A., Lakhan, A., ur Rehman, M., Pnhwar, A. O., & Sahito, M. A. (2021, August). Pervasive futuristic healthcare and blockchain enabled digital identities-challenges and future intensions. In 2021 International Conference on Computing, Electronics & Communications Engineering (iCCECE) (pp. 30-35). IEEE.

[13] Lakhan, A., Mastoi, Q. U. A., Dootio, M. A., Alqahtani, F., Alzahrani, I. R., Baothman, F., ... & Khokhar, M. S. (2021). Hybrid workload enabled and secure healthcare monitoring sensing framework in distributed fog-cloud network. Electronics, 10(16), 1974.