East African Journal of Information Technology
eajit.eanso.org
Volume 5, Issue 1, 2022
Print ISSN: 2707-5346 | Online ISSN: 2707-5354
Title DOI: https://doi.org/10.37284/2707-5354

EAST AFRICAN NATURE & SCIENCE ORGANIZATION

Original Article

Analysing the Obstacles in Agile Software Development Approach: A Review.

Amos O. Jarikre[1], Yogesh Kumar Sharma[2], Amoako Kani John[3] & Stercy Kwasi Bailey[1]*

[1] Shiv-India Institute of Management and Technology, Accra, Ghana.
[2] Om Sterling Global University, Hisar, Haryana, India.
[3] Heritage Christian College, Amasaman, Ghana.
* Correspondence ORCID: https://orcid.org/0000-0002-8885-2173; email: jarky4u2c@gmail.com.

Article DOI: https://doi.org/10.37284/eajit.5.1.520

Date Published: *4 January 2022*

Keywords: *Software, Agility, Software Industry, Software Development, Agile SDA.*

ABSTRACT

The development of reusable and extensible software for business purposes has been the hallmark of the day. More developers are taking advantage of numerous approaches towards reaching their goals. One such approach is the agile approach in the development of extensible applications which has become so popular since its introduction over a decade ago. Using an agile approach that has a defined value in developing applications portray numerous benefits which have been identified by various scholars pointing out their outcomes as motivating factors of its adoption. With all such outline benefits, there exist some potential obstacles to agile developmental approach which has not been fully addressed. Hence, this article is aimed at analysing the obstacles which software developers face during agile development through a database search and also to guide them on ways to overcome such obstacles.

APA CITATION
Jarikre, A. O., Sharma, Y. K., John, A. K., & Bailey, S. K. (2022). Analysing the Obstacles in Agile Software Development Approach: A Review. *East African Journal of Information Technology*, *5*(1), 1-6. https://doi.org/10.37284/eajit.5.1.520

CHICAGO CITATION
Jarikre, Amos O, Yogesh Kumar Sharma, Amoako Kani John, & Stercy Kwasi Bailey. 2022. "Analysing the Obstacles in Agile Software Development Approach: A Review". *East African Journal of Information Technology* 5 (1), 1-6. https://doi.org/10.37284/eajit.5.1.520.

HARVARD CITATION
Jarikre, A. O., Sharma, Y. K., John, A. K., & Bailey S. K. (2022) "Analysing the Obstacles in Agile Software Development Approach: A Review", *East African Journal of Information Technology*, 5(1), pp. 1-6. doi: 10.37284/eajit.5.1.520.

Article DOI: https://doi.org/10.37284/eajit.5.1.520

IEEE CITATION

A. O. Jarikre., Y. K. Sharma., A. K. John., & S. K. Bailey. "Analysing the Obstacles in Agile Software Development Approach: A Review", *EAJIT*, vol. 5, no. 1, pp. 1-6, Jan. 2022.

MLA CITATION

Jarikre, Amos O, Yogesh Kumar Sharma, Amoako Kani John, & Stercy Kwasi Bailey. "Analysing the Obstacles in Agile Software Development Approach: A Review". *East African Journal of Education Studies*, Vol. 5, no. 1, Jan. 2022, pp. 1-6, doi:10.37284/eajit.5.1.520.

INTRODUCTION

Over the years, developers of software's have been focusing on the traditional waterfall software development methods in meeting their demands for business applications. Such methods require the developer experience and voluminous documentation to support the developmental process. The traditional waterfall software development methods which were popularly used by developers saw its systematic setback with the introduction of the agile development method some decade ago. This was due to their outcome that motivates the developers as well as its defined values in the development process for business applications.

Agility as it is referred to by some scholars was not all that new idea or approach towards achieving software developmental goals in industry or business environment, but according to Al-Saqqa, Sawalha and AbdelNabi (2020), agility is the ability to adaptively promote quick response to any change, either in the environment, in the user requirements or in any delivery constraints. Proposing agility in the software industry or business environment was completely an innovative concept (Gandomani et al., 2013) which has to be systematically adopted. Agility (agile approaches) comes with better feasibilities when compared with the traditional methods of software development on a small scale. Although there have been notable concerns about these feasibilities, it is the high-quality outcome and ability to meet customers satisfaction that has over time persuaded software developers and practitioners to adopt and utilise these approaches (Glazer, 2010).

In order to systematically analyse the obstacles in Agile Software Development Approach (SDA), this study looked at the concepts of agile software development approach, traditional approach towards software development, comparison between traditional and agile approach towards software development, outlines the obstacles to agile approach in software development and suggest approaches to be adopted by developers when faced with such obstacles during their developmental processes.

CONCEPTS OF AGILE SOFTWARE DEVELOPMENT APPROACH

The ability to minimise software developmental risk such as overrun developmental cost, changing developmental requirements and bugs clearly explains the team "agile development approaches" during the phase of additional software functionalities. Such approaches (agile software development approaches) are usually carried out in iterations coming some of the increments of the added new software functionalities. Hence Beerbaum (2019) was able to point out that one benefit of the agile approach in software development is that it allows the development of software through processes of iterations and incremental changes. According to Pereira and de FSM Russo (2018), the agile software development approach is an enabler that accelerates software delivery, manage its priorities changes and increase its productivity. Agile software development methods or approaches are widely used by developers and business practitioners in the software industry as a way to more rapidly developing and deliver new software (Venkatesh et al., 2020) that could meet business needs. The agile development method, according to Ruk et al. (2019), has also engrossed extensive communal and academic consideration as the restrictions of conventional software development techniques become apparent.

Agile development method has Scrum method - delivering the highest value in the shortest time; Test Driven Development (TDD) method - based on building a small iteratively automated testing programs; Extreme programming method -

Article DOI: https://doi.org/10.37284/eajit.5.1.520

improve the software quality by taking the concepts of software engineering to an extreme level (Schmidt, 2016); Feature-driven development (FDD) method - manages short incremental iterations leading to functional software and Dynamic System Development Method (DSDM) - provides rapid application development based on the agile principles (Anwer et al., 2017) as its different forms. Adopting agile software development signifies shoring more cooperative environments and organisational policies that enable self-organisation and team efficiency (Ruk et al., 2019). According to Marandi and Ali (2017), it creates the ability to reduce time and costs, improve software quality as well as fewer defects, meet the requirement of clients and ability of delivering software quality products in a timely basis. All of these signify some of the enormous usages of agile development. Other advantages achieved by adopting agile software development approaches were outlined by Choudhary and Rakesh (2016) as enabling the improvement on communication as well as coordinating team members, smart design flexibility, contain vast reasonable process and possesses quick releases.

Traditional Approach Towards Software Development

Traditional software development approach (SDA) is known for their linear approaches that constitute several stages of development processes that must be completed in sequential order. Such sequential order requires the completion of one phase before embarking on the next phase according to the software development plan. As a sequential approach, traditional SDAs usually are associated with software requirements gathering and documentation, design of the system, coding, testing, which includes unit, system and user's acceptance testing, fixing of bugs and release of software as being their development stages.

The traditional approaches are very useful in developing complex software, which helps to eliminate informal software requirements and deliver high-quality software's that meets the requirement of users within a predefined time limit (Matharu et al., 2015). Waterfall approach, spiral approach, iterative and incremental approach, evolutionary approach etc., are examples of some traditional software development approaches and methods which are often referred to as heavyweight approaches (Mall, 2018). Although the traditional approaches or methods are meant for complex software development referred to as "heavy applications", there exist some issues associated with their complexity which were outlined by Braude and Bernstein (2016) as writing software's requirements, advance planning of projects, design formalities that correspond to the written requirements, building design code in accordance with all the written requirements and testing of the software functionalities and also in compliance with design requirements.

Comparison between Traditional and Agile Approach towards Software Development

Traditional SDA relies mostly on the assumption that technology innovations are created in an academic environment, which later are then transferred to industry using a sequential flow of activities (Mikkonen et al., 2018) but this assumption is slowly fading out as modern approaches have seen close collaboration between academia and industry both in small and as well as large scale. For agile SDA, it focuses on quickly delivering complete and functional software products. The agile software development process provides the ability to cope with ever-changing requirements (Ferdinansyah & Purwandari, 2021). Previous studies have shown that agile development reduced the cost of system development which includes testing, and also enhanced IT – business alliances (Tarhini, Yunis & El-Kassar, 2018). One major comparative advantage of the agile development process is that it delivers software that fulfils customer needs rapidly and continuously (Karhapää et al., 2021).

Each of these SDA's (traditional and agile) possesses some distinct characteristics that feature as their main goal toward software development. This statement clearly shows that there exist some characteristics which are pertinent to specific SDA's. *Table 1* below presents some of these characteristics based on some software parameters between traditional (often referred to as heavy) and agile (light) SDA's.

Article DOI: https://doi.org/10.37284/eajit.5.1.520

Table 1: Comparison between traditional and agile approaches towards software development using some specific parameters

No.	Software parameters	Agile SDA	Traditional SDA
1	Development approach	Iterative approach	Sequential approach
2	Scalability	Light-weighted	Heavy weighted
3	Size of project	Small projects	Large and complex projects
4	Attitudinal nature	Adaptive nature	Predictive nature
5	Size of team	Small	Medium
6	Development duration	Short term (speedily)	Long term
7	Project cost	Cost-effective (low cost)	Expensive
8	Project management	Enhance collaboration (decentralised)	Strictly based on commands and control (autocratic)
9	Budgets determinant	per-sprint basis	per-project basis
10	Software documentation	Few (low)	Detailed (high)
11	Risks involvement	Risks are unknown having major impact	Known risks having a minor impact
12	Software modification	Easily modifiable	Difficult to modify

With the above-mentioned parameters on SDA's, the agile SDA will be attractive to customers who need urgent software for their business purposes since according to Al-Saqqa et al. (2020), they are more adaptive to the requested changes from customers. This occurs as a result of the regular involvement of the customers during all its iterative processes. Hence, the agile SDA fosters collaborative trust between software developers and their customers. As attractive as the agile SDA may be when compared to traditional SDA, software developers also encounter some obstacles during the developmental process using agile SDA. The next session below outlines some of these obstacles.

OBSTACLES TO AGILE APPROACH IN SOFTWARE DEVELOPMENT

Migrating from traditional SDA to agile SDA need careful planning and consideration because it comes with some obstacles if not properly handled. Some of the known obstacles are discussed in the next sections below.

Lack of Business Owner Interest

One major obstacle to agile SDAs is the involvement of newcomers such as business developers who hesitate to use software without considering business requirement documentation as a necessary requirement needed. For business owner's view the use of agile SDA as mainly a contract binding that exists between IT and themselves hereby giving up on the business requirements document. This poses a threat to the entire business environment because these owners will not be able to control or monitor the business direction since there is not any interest and awareness of the product.

Use of Obsolete Development Tool

Most business analysts are still using Microsoft excel and word to author business development requirements which are mainly obsolete when compared to the recently used Application Life Cycle Management (ALM) software tool, which most software developers do use for agile applications making it easier for developers to be able to decompose users' stories into useful developmental tasks. According to Gandomani et al. (2013), business owners should endeavour to use tools that can supply incremental evolution, version management, re-working, continuous integration and other available agile technologies. The use of obsolete tools makes it difficult for Business Analysts and stakeholders to actualise the full benefits of the agile manifesto, which lack the cross-departmental collaboration that is expected to occur.

Article DOI: https://doi.org/10.37284/eajit.5.1.520

Lack of Collaborated Teams

The agile manifesto makes it clear that a cross-departmental team are needed for better collaboration in order to deliver the product, but it did not specify that such teams are inclusive of technical and business minds; rather, it projected the view that it is all about developers which many believe to be core coders. But it means the term is making it seems as if it is all about the technical developers rather than an all-inclusive team that includes business analyst and stakeholder. Such terms used should be clearly explained in order for all concerns in the software development process to get involved in ensuring a potential release of a useful working product.

Behavioural Change

The behavioural change of individuals is certain to most sectors of an economy as it is very difficult for people to adapt to change whenever it is needed. So, introducing the agile development process to businesses or developers might resist some kind of attitudinal reactions which might stand as a setback if not carefully handled. Since it is difficult for persons to change behaviour, the agile transformation process should be slowly introduced to people by means of training. Such a view is supported by Srinivasan and Lundqvist (2010) that insist on managers selecting the appropriate personnel and providing them with the necessary training and creating a set of work practices that promote process excellence.

The above obstacles range from managerial to technical issues, which should be carefully before business owners adopt the agile transformation process. Nevertheless, expert decisions should be taken while recognising all of the above-mentioned obstacles before fully adopting the agile SDA.

CONCLUSION

This research considers looking into agile SDA while also explaining traditional SDA and some comparisons between agile and traditional SDA. Although the concept of the agile software development method is innovative and feasible with some similarities to other developmental processes, its inefficiency can be witnessed when used in large business organisations. Hence, it is advisable to adopt the agile approach only on a small scale for better efficiency and productivity.

REFERENCES

Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies*, 14(11).

Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. S. (2017). Agile software development models TDD, FDD, DSDM, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2), 1-10.

Beerbaum, D. (2021). *Applying Agile Methodology to regulatory compliance projects in the financial industry: A case study research*. https://dx.doi.org/10.2139/ssrn.3834205

Braude, E. J., & Bernstein, M. E. (2016). *Software engineering: modern approaches*. Waveland Press.

Choudhary, B., & Rakesh, S. K. (2016). An approach using agile method for software development. In *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)* (pp. 155-158). IEEE.

Ferdinansyah, A., & Purwandari, B. (2021). Challenges in Combining Agile Development and CMMI: A Systematic Literature Review. In *2021 10th International Conference on Software and Computer Applications* (pp. 63-69).

Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M., & Nafchi, M. Z. (2013). Obstacles in moving to agile software development methods; at a glance. *Journal of Computer Science*, 9(5), 620.

Glazer, H. (2010). Love and marriage: CMMI and agile need each other. CrossTalk, 23: 29-34.

Karhapää, P., Behutiye, W., Rodríguez, P., Oivo, M., Costal, D., Franch, X., & Abherve, A. (2021). Strategies to manage quality requirements in agile software development: a

Article DOI: https://doi.org/10.37284/eajit.5.1.520

multiple case study. *Empirical Software Engineering*, 26(2), 1-59.

Mall, R. (2018). *Fundamentals of software engineering*. PHI Learning Pvt. Ltd

Marandi, A. K., & Ali, D. (2017). An Approach of Statistical Methods for Improve Software Quality and Cost Minimisation. *International Journal of Applied Engineering Research*, 12(6), 1054-1061.

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. *ACM SIGSOFT Software Engineering Notes, 40*(1), 1-6.

Mikkonen, T., Lassenius, C., Männistö, T., Oivo, M., & Järvinen, J. (2018). Continuous and collaborative technology transfer: Software engineering research with real-time industry impact. *Information and Software Technology*, 95, 34-45.

Pereira, J. C., & de FSM Russo, R. (2018). Design thinking integrated in agile software development: A systematic literature review. *Procedia computer science*, 138, 775-782.

Ruk, S. A., Khan, M. F., Khan, S. G., & Zia, S. M. (2019). A survey on Adopting Agile Software Development: Issues & Its impact on Software Quality. In *2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-5). IEEE.

Schmidt, C. (2016). *Agile software development teams*. Springer International Publishing

Srinivasan, J., & Lundqvist, K. (2010, February). Agile in India: Challenges and lessons learned. In *Proceedings of the 3rd India software engineering conference* (pp. 125-130).

Tarhini, A., Yunis, M., & El-Kassar, A. N. (2018). Innovative sustainable methodology for managing in-house software development in SMEs. *Benchmarking: An International Journal.*

Venkatesh, V., Thong, J. Y., Chan, F. K., Hoehle, H., & Spohrer, K. (2020). How agile software development methods reduce work exhaustion: Insights on role perceptions and organisational skills. *Information Systems Journal*, 30(4), 733-761.

