



COMPARATIVE ANALYSIS OF POTENTIAL FRAMEWORKS FOR AGILE DEVELOPMENT OF LARGE SOFTWARE PROJECTS

Petra Balaban¹,
[0000-0002-1513-324X]

Dejan Viduka^{2*},
[0000-0001-9147-8103]

Ana Bašić³
[0009-0009-4137-3296]

¹The Higher Education Technical School of Professional Studies in Novi Sad, Novi Sad Serbia

²Faculty of applied management, economics and finance, Belgrade, University Business Academy, Novi Sad, Serbia

³Information Technology School - ITS, Belgrade, Serbia

Abstract:

The current business environment requires an agile approach to software development in order to deliver products quickly and efficiently. This paper explores several different but well-known agile frameworks and analyzes their strengths and weaknesses in the context of large and complex software projects. Through comparison, we discover that each of these frameworks offers specific approaches to scaling project tasks with their unique characteristics. The final choice of framework depends on the specific needs and goals of the organization or the preferences of the decision-making team managers. The paper emphasizes the importance of careful assessment and analysis of each framework, consulting with experts, and gathering feedback in order for organizations to effectively scale their operations on a given project and deliver high-quality products within the set deadlines and expected quality.

Keywords:

Agile Framework, High-Quality Software, Software Projects, Large Software Projects, Complex Software Projects.

INTRODUCTION

Agile software development is a modern approach to development [1] that emphasizes adaptability and flexibility to changes. This iterative and incremental approach aims to deliver software products [2] that function efficiently and on time while focusing on continuous improvement and customer satisfaction.[3]

Agile development finds its roots in the Agile Manifesto, which emerged in 2001. [4] This manifesto, crafted by a group of programmers, underscores four fundamental values. [5] These values prioritize the significance of individuals and interactions above processes and tools, the delivery of working software, collaboration with customers, and adaptability over rigid adherence to plans. [6]

In Agile software development, various methods and frameworks are employed, including Scrum, Kanban, Extreme Programming (XP), and Lean software development. [7] While each of these approaches entails its unique practices and roles, they all share a common emphasis on collaboration, communication, and continual enhancement. [8]

Correspondence:

Dejan Viduka

e-mail:

dejan@viduka.info



One of the main advantages of Agile software development is its ability to quickly adapt to changing customer requirements and needs. [4] Teams deliver software that works in short iterations, allowing them to gather feedback and adjust their approach as needed. [9] This often results in higher customer satisfaction [10] and more successful software products.[11]

Agile software development has become extremely popular in recent years [12] as organizations strive to deliver software products faster and more efficiently. However, implementing Agile practices can be challenging, especially in large and complex organizations. [13] To address this challenge, various frameworks and tools have been developed to help organizations scale [14] Agile practices and manage large software projects.

In this paper, the concept of agility as a business model is explained and several different agile frameworks are explored. The strengths and weaknesses of various agile frameworks in the context of large and complex software projects are analyzed.

2. AGILITY AS A BUSINESS METHOD

The Agile philosophy emphasizes four core values:

- Prioritizing individuals and their interactions over processes and tools underscores the significance of teamwork and communication. It recognizes that software development is fundamentally a human endeavor, where the quality of interaction among team members is paramount. While tools play a role, the emphasis lies on collaborative effort rather than the tools themselves.
- Placing a higher value on delivering value over extensive documentation acknowledges that while documentation serves its purpose as a resource, the primary aim of software development is to deliver tangible business benefits through functional software, not exhaustive documentation.
- Emphasizing collaboration with the customer over negotiation stresses the importance of close communication and frequent engagement with customers. By actively listening and incorporating feedback, development teams gain a deeper understanding of stakeholders' needs and preferences.
- Prioritizing responsiveness to change over rigid adherence to a plan acknowledges the inevitability of change in software development. It highlights the necessity for project plans to be adaptable and flexible, allowing for adjustments as circumstances evolve.

3. AGILE DEVELOPMENT APPROACH

3.1. AGILE DEVELOPMENT METHODS

Numerous Agile software development methods are prevalent in the industry. Among them, notable ones include [10, 15]:

- Scrum, a widely utilized Agile methodology, prioritizes transparency, inspection, and adaptation in software development processes.
- Kanban, another Agile methodology, centers on visualizing workflow, constraining work in progress, and promoting continuous delivery.
- Extreme Programming (XP) accentuates software quality and customer satisfaction within Agile frameworks.
- Lean software development, drawing from Lean manufacturing principles, emphasizes waste reduction, value delivery, and ongoing improvement.
- Crystal, an Agile methodology, places emphasis on people, communication, and team dynamics, incorporating practices like incremental delivery, frequent feedback, and continuous enhancement.
- Dynamic Systems Development Method (DSDM), an Agile approach, highlights collaboration, communication, and the delivery of business value.
- Feature-Driven Development (FDD) underscores feature delivery and incremental development within Agile methodologies.

Each of these Agile methodologies comes with its distinct set of principles, practices, and tools. The selection of a methodology hinges on the specific requirements and objectives of a project, as well as the culture and structure of the organization.

3.2. CHOOSING AN AGILE SOFTWARE DEVELOPMENT METHOD

When deciding on an Agile software development approach, it's crucial to consider various factors to ensure adaptability and project success. [16]

- For small and straightforward projects: Scrum or Kanban offer viable options. Scrum fosters team collaboration and communication, while Kanban emphasizes workflow visualization and continuous delivery. Both methods are flexible and allow for adjustments to changing requirements and feedback.



- For larger and more intricate projects: Large-Scale Scrum (LeSS) or Scaled Agile Framework (SAFe) are suitable choices. These frameworks provide practices and guidelines for managing multiple teams and coordinating efforts across the organization.
- If prioritizing software quality: Extreme Programming (XP) stands out. XP employs practices like test-driven development, continuous integration, and pair programming to ensure high software quality.
- If the primary focus is on delivering business value and customer satisfaction: Feature-Driven Development (FDD) or Dynamic Systems Development Method (DSDM) are recommended. FDD emphasizes feature delivery and incremental development, while DSDM emphasizes collaboration, communication, and delivering business value. [17]

The ultimate selection of an Agile method should stem from a deep understanding of project requirements, team capabilities, and organizational culture and structure. Continual evaluation and adaptation of the methodology are crucial to meeting the evolving needs and objectives of the project. [18]

4. FRAMEWORK COMPARISON

Large-scale Agile software development involves applying Agile principles in the context of larger and more complex projects. [19] There are several available methods for large-scale Agile software development,

each with its own advantages and disadvantages. Some of the most popular methods include: SAFe, LeSS, and Nexus.

- Scaled Agile Framework (SAFe) is a popular framework for large-scale Agile software development. [5] It provides a comprehensive set of guidelines for organizations to implement Agile principles in a structured and scalable manner. SAFe is based on three primary components: Agile Teams, Agile Release Trains (ARTs), and Solution Trains.
- Large-Scale Scrum (LeSS) is another popular framework for large-scale Agile software development. [5] Based on Scrum principles, with some modifications to adapt to larger and more complex projects. It relies on fewer roles, artifacts, and events compared to SAFe.
- Nexus is a scaling Scrum framework that focuses on coordinating the work of multiple Scrum teams to deliver one integrated product. [5] Nexus provides a set of practices and roles that help teams work more efficiently together. It is designed to be lightweight and flexible, which may ease adoption compared to SAFe or LeSS.

Figure 1 shows a comparison of scalable agile frameworks. X-Axis of the diagram represents applicability of framework. Reflect the potential coverage of the framework in the context of the enterprise. The higher the value, the more the scope of the framework can be applied to an entire enterprise or a large company. Y-Axis of the diagram represents breadth which reflect the quantity of values, principles, artifacts, rules and knowledge available in those frameworks.



Figure 1. A comparison of the scaling Agile frameworks.



The higher the value, the most comprehensive (and possibly restrictive) is the framework.

As shown in Figure 1, the Scrum framework is provided as a point of comparison. Scrum is actually the most common team-level Agile framework around the world. All the scaling frameworks analyzed here are based on Scrum and are therefore considered to be broader than Scrum. Additional values, principles, artifacts, and rules are added on top of Scrum.

5. DISCUSSION

When it comes to scaling Agile software development, organizations have various tools and frameworks at their disposal. Some of the prominent frameworks include Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS), and Nexus.

The choice of framework depends on various factors, such as project size and complexity, organizational culture and structure, and desired level of flexibility and autonomy. [20] Each framework has its advantages and drawbacks, so it is important to carefully evaluate each one before making a decision. [21]

SAFe provides a comprehensive and structured approach to large-scale Agile software development. It offers clear hierarchy and management structure, which is particularly useful for organizations with complex and distributed teams. However, some organizations may find SAFe too rigid or overly bureaucratic, preferring a lighter approach.

LeSS is a flexible and lightweight approach to large-scale Agile software development based on Scrum principles. It emphasizes collaboration, transparency, and continuous improvement and can be adapted to different organizational structures and cultures. However, some organizations may find that LeSS provides too little structure or guidance.

Nexus is a scaling Scrum framework that provides a lightweight and flexible approach to large-scale Agile software development. It is suitable for organizations that value flexibility and autonomy, but may require more guidance or structure to effectively scale Agile practices. However, some organizations may find that Nexus provides too little structure or instructions.

SAFe, LeSS, and Nexus are effective methods for large-scale Agile software development, but each has its own advantages and weaknesses. Organizations should carefully assess their specific needs and constraints before choosing a method and should be prepared to adapt and evolve their practices based on feedback and results over time.

6. CONCLUSION

In today's rapidly evolving and intricate business environment, scaling Agile software development to larger scopes is gaining paramount significance. Organizations aiming to effectively expand Agile practices have at their disposal a plethora of tools and frameworks to leverage. Each of these frameworks has its advantages and disadvantages,[22] so it is important to carefully assess each framework before making a decision. SAFe provides a comprehensive and structured approach to large-scale Agile software development. It offers clear hierarchy and management structure, but some organizations may find it too rigid or overly bureaucratic. On the other hand, LeSS provides a flexible and lightweight approach based on Scrum principles. It emphasizes collaboration, transparency, and continuous improvement, but may provide less structure and guidance. Nexus is a lightweight and flexible framework for scaling Scrum, but for some organizations, it may provide too little structure or guidance.

The ultimate selection of a framework hinges on the particular requirements and objectives of the organization. It's vital to meticulously assess each framework and opt for the one that aligns best with the organizational culture, structure, and goals. Once the appropriate framework is established, organizations can adeptly scale their Agile practices, ensuring the timely and efficient delivery of high-quality software products.

7. REFERENCES

- [1] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile Software Development: Methodologies and Trends," *Int. J. Interact. Mob. Technol.*, vol. 14, pp. 246-270, 2020. DOI: <https://doi.org/10.3991/ijim.v14i11.13269>.
- [2] Z. Dragičević, S. Bošnjak, "Agile architecture in the digital era: Trends and practices," *Strategic Management*, vol. 24, no. 2, pp. 12-33, 2019. DOI: <https://doi.org/10.5937/StraMan1902011D>.
- [3] D. Ciric, B. Lalic, D. Gracanin, I. Palcic, N. Zivlak, "Agile Project Management in New Product Development and Innovation Processes: Challenges and Benefits Beyond Software Domain," in *IEEE International Symposium on Innovation and Entrepreneurship (TEMS-ISIE)*, 2018
- [4] C. Baham, R. Hirschheim, "Issues, challenges, and a proposed theoretical core of agile software development research," *Information Systems Journal*. DOI: <https://doi.org/10.1111/isj.12336>.



- [5] F. Almeida, E. Espinheira, "Large-Scale Agile Frameworks: A Comparative Review," *Journal of Applied Sciences, Management and Engineering Technology*, vol. 2, issue 1, pp. 16-29, March 2021.
- [6] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, "Agile software development methods: Review and analysis," VTT publication 478, Espoo, Finland, 2002. [Online]. Available: <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>.
- [7] M. Alqudah, R. Razali, "A Review of Scaling Agile Methods in Large Software Development," *IEEE Transactions on Software Engineering*, vol. 6, no. 6, 2016.
- [8] M. S. Raunak, D. Binkley, "Agile and Other Trends in Software Engineering," in *IEEE 28th Annual Software Technology Conference (STC)*, 2017.
- [9] H. Edison, X. Wang, K. Conboy, "Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review," *IEEE Transactions on Software Engineering*, 2021.
- [10] M. Al-Zewairi, M. Biltawi, W. Etaiwi, A. Shaout, "Agile Software Development Methodologies: Survey of Surveys," *Journal of Computer and Communications*, vol. 5, pp. 74-97, 2017. DOI: <https://doi.org/10.4236/jcc.2017.55007>.
- [11] H. Alahyari, R. Berntsson Svensson, T. Gorschek, "A study of value in agile software development organizations," *Journal of Systems and Software*, vol. 125, pp. 271-288, 2017. DOI: <https://doi.org/10.1016/j.jss.2016.12.007>.
- [12] R. Hoda, N. Salle, J. Grundy, "The Rise and Evolution of Agile Software Development," *IEEE SOFTWARE*, 2018.
- [13] R. Vallon, B. J. da Silva Estácioc, R. Prikladnickic, T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, pp. 161-180, 2018.
- [14] P. Jain, A. Sharma, L. Ahuja, "The Impact of Agile Software Development Process on the Quality of Software Product," *IEEE*, 2018.
- [15] B.A. Andrei, A.C. Casu-Pop, C.C. Gheorghe, C.A. Boiangiu, "A study on using waterfall and agile methods in software project management," *Journal Of Information Systems & Operations Management*, 2019.
- [16] R. Kumar, P. Maheshwary, T. Malche, "Inside Agile Family: Software Development Methodologies," *International Journal of Computer Sciences and Engineering*, 2019.
- [17] G. S. Matharu, A. Mishra, H. Singh, P. Upadhyay, "Empirical Study of Agile Software Development Methodologies: A Comparative Analysis," *ACM SIGSOFT Software Engineering Notes*, vol. 40, issue 1, pp. 1-6, 2015.
- [18] T. Raharjo, B. Purwandari, E. K. Budiardjo, R. Yuniarti, "The Essence of Software Engineering Framework-based Model for an Agile Software Development Method," *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, 2023.
- [19] A. Putta, M. Paasivaara, C. Lassenius, "Adopting Scaled Agile Framework (SAFe): A Multivocal Literature Review," in *Proceedings of the 19th International Conference on Agile Software Development: Companion*, 2018.
- [20] A. Putta, M. Paasivaara, C. Lassenius, "Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review," in *PROFES 2018, LNCS 11271*, pp. 334-351, 2018.
- [21] W. Alsaqaf, M. Daneva, R. Wieringa, "Analysing Large-Scale Scrum Practices with Respect to Quality Requirements Challenges," in *Proceedings of the 15th International Conference on Software Technologies*, 2020.
- [22] F. Almeida, E. Espinheira, "Adoption of Large-Scale Scrum Practices through the Use of Management 3.0," *Informatics*, vol. 9, no. 1, 2022. DOI: <https://doi.org/10.3390/informatics9010020>.