

A Survey Paper on Various Scaling Methodologies Available in Agile Software Development

Sunder V¹ and Dr. P. Deepalakshmi²

¹Associate Professor, Department of Computer Science and Engineering, Sree Narayana Guru College of Engineering and Technology, Payyanur, Kannur, Kerala, India

Email: sunderv@sngcet.ac.in

²Professor, Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Tamilnadu, India.

Email: deepa.kumar@klu.ac.in

Abstract—Agile is a project management methodology widely accepted in all IT industries as it is highly adaptable and flexible based on the type of the project. It is considered to be an effective way to carry out a project. Agile development's explosive growth is a result of its capacity to handle complicated issues and ease the switch from conventional approaches. A lack of adoption guidelines is caused by the fact that no single agile technique can be used in every organisation. Some of the most popular agile development methodologies are based on the basics of the software engineering framework and incorporate aspects of well-known techniques including Scrum, Kanban, SAFe, Extreme programming, Less, Spotify Agile, Nexus, Scrum of Scrums, and Disciplined Agile. Organizations can construct software development methodologies based on shared knowledge or common ground by using the Essence of Software Engineering framework. In this paper, a common strategy adopted for industries is discussed and a comparative study is performed and explained on the various agile scaling approaches.

Index Terms— Agile, Scrum, Kanban, Extreme Programming, DSR, Essence Architecture.

I. INTRODUCTION

The adoption of the agile methodology is currently accelerating in organisations [1]. This method has a major impact on project performance and corporate success, allowing organisations to address complex issues in a time of frequent disruptions [2]. Most management levels are familiar with the idea of Agile teams for innovation, which aims to retain close contact with customers while quickly responding to changing business conditions [3]. To deploy Agile, the organisation could use the Agile methodologies. Although a number of Agile methodologies—including Extreme programming, Scrum, SAFe, Kanban, Spotify, LeSS, Disciplined Agile, and Nexus, are available, no single Agile methodology alone is universally applicable [4], due to its lack of adoption standards. Some of the key points that make Agile process a method of choice for industrialist are:

- **Faster Time-to-Market et:** Scrum and Kanban are both agile approaches that place a strong emphasis on iterative development and frequent releases. This strategy enables businesses to respond quickly to shifting consumer demands.
- **Increased Customer Satisfaction:** Agile places a high value on consumer feedback and cooperation. Customers' needs and demands can be met in the final product by incorporating them in the development process. This will ultimately increase customer satisfaction.

- **Flexibility and Adaptability:** Agile methodologies promote adaptability and flexibility. Teams can adjust their priorities and direction as they receive new information or as the business environment changes.
- **Improved Quality:** Agile methodologies emphasize continuous testing and quality assurance throughout the development process. These results in a higher-quality product as issues are identified and addressed early, reducing the need for extensive rework later.
- **Collaborative Work Environment:** Agile methodologies encourage cross-functional collaboration and self-organizing teams. This promotes a more collaborative and empowered work environment, which can lead to increased employee satisfaction and productivity.
- **Transparency and Visibility:** Agile methods, such as daily stand-up meetings and visual boards, offer insight into the status of projects. Stakeholders may take informed decisions and quickly resolve problems because of this visibility.
- **Risk Mitigation:** Early in the development process, agile approaches assist in identifying and addressing risks. By delivering working increments of a product regularly, teams can assess project status and risks continually, allowing for timely mitigation strategies.
- **Cultural Shift:** Agile implementation frequently necessitates a change in an organization's culture. It promotes a mind-set that values cooperation above hierarchy, learns from mistakes, and strives for constant development.
- **Scaling Agile** Many organizations have chosen frameworks like SAFe (Scaled Agile Framework) and LeSS (Large-Scale Scrum) to deploy Agile concepts at the business level. Agile scaling might be difficult, but it enables larger firms to gain from its techniques.
- **Challenges and Pitfalls:** While Agile has many benefits, its successful adoption also comes with challenges. Organizations may struggle with cultural resistance, changing roles and responsibilities, and maintaining discipline in Agile practices.

It is not only in the field of Software but in all the areas like telecommunication, automobiles, retails, governance, banking etc software development methodology finds a unique place. All software engineering techniques have a common foundation of software engineering methods [5].

II. THE PRINCIPLE VALUES

The current scaling Agile methodologies, including Scrum of Scrums, Disciplined Agile, Spotify, Agile, LeSS, and SAFe are examined here. Focus is given on scaling techniques, including specific Agile techniques like Extreme Programming, Kanban, and Scrum.

A. The Software Engineering Basics

The term "common ground" mentioned here alludes to Ivar Jacobson's concept of "common ground" in software engineering [5]. The idea that no software development process is suitable for everyone creates the common ground. The ability of the common ground is to give the team a shared foundation for comprehending the fundamentals of software development methodologies.

B. Importance of Agile Scaling

The Scrum [6] and Extreme Programming [7] agile methodologies, which are widely used, are effective for small project teams. The complex scenario is challenging to manage when a project needs to be scaled up at the corporate level [8]. The Project Management Institute [5] classifies agile approaches as team and scaling techniques. The enterprise agile approaches known as Disciplined Agile Delivery (DAD) [11] and Scaled Agile Framework (SAF) [10] were developed by Scott Ambler and Dean Leffingwell. The approaches are referred to in the literature as scaling Agile methods [15], enterprise Agile framework [12], scaling methods and approach [12], and scaling framework [13], as there are no formal labels for them. The Scaled Agile Framework (SAFe), Large-Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), and Nexus are a few well-liked scaling Agile methodologies and frameworks [13, 14].

C. The Essence Architecture – The SEMAT Kernel

The 'Essence', which provides a common language and is regarded as a crucial component of software engineering, was created [17] by the Object Management Group [OMG] and includes methods and procedures. Solid theory, tried-and-true concepts, and industry best practices served as the foundation for The Essence. The essence architecture as designed by OMG is shown in the Figure.1. In order to address some issues within the current Software Engineering field, which is characterised by the existence of immature implementation,

SEMAT (Software Engineering Methods and Theory) was developed. These problems include the lack of a theoretical framework that is widely accepted, the quantity of different approaches and their modifications, and the discrepancy between business practices and academic research.

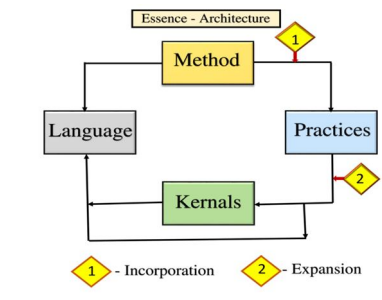


Figure 1: The diagrammatic representation of the Essence – Architecture

III. DATA SCIENCE RESEARCH METHOD

The Data Science Research (DSR) method is suitable for finding a solution for a problem by producing artifacts. The methodology involves a series of steps as indicated below [18]. Figure. 2: depicts the overall methodology. The general practices that are followed in the DSR approach are explained here.

A. Problem Identification

Selection of proper Agile Method is always a challenge for the industry after the problem identification is done. It is mainly due to the lack of proper guidance in the selection process.

B. Objective definition for the solution

Using the Essence architecture concept, a model for the Agile development approach is established and constructed to design the solution.

C. Design and Development

The required model is developed using appropriate Agile methodologies, such as LeSS, Spotify Agile, SAFe, Disciplined Agile, Scrum of Scrums and Nexus.

D. Demonstrating the implementation

Design of the solution for the problem identified for the case study.

E. Evaluating the results

A varied method is available to evaluate the obtained results. Some of the common approaches are interviews, focus group discussions, discussing with the stakeholders surveys etc.

F. Communicating the findings

Submission of the report to some authorities and to get it preserved by maintaining a proper report for future use.

IV. THE ESSENTIALS OF AGILE

The Agile process comprises three main elements that are employed by the industries now. They are Mindset, Values and Principles and it is adopted for project development. The Figure 3 shows the essentials level in an Agile Methodology. The main elements are Mindset, Values, Principles, Practices and finally Tools and Processes. It is found that the essentials, Mindset, Values and Principles don't change often, whereas Practices, Tools and Processes may change often. As moving from right to left of the layers, it becomes more intangible whereas moving from left to right it becomes highly visible.

V. COMPARATIVE STUDY ON AGILE SCALING METHODS

Scaling Agile approaches currently include Agile team-level techniques like Extreme Programming, Kanban and Scrum, which are widely adopted by businesses. This provides a common platform for scaling Agile techniques as depicted in Figure 4.

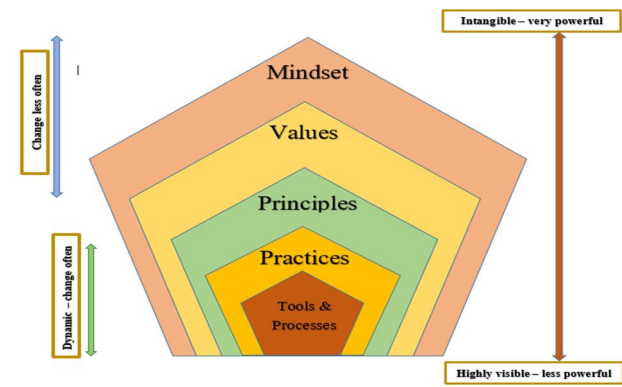


Figure 2. The essentials of Agile

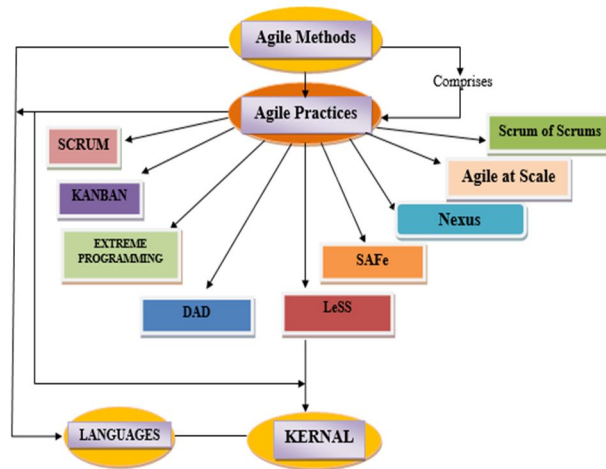


Figure 3: Research Methodology using popular scaling methods

Based on the Essence architecture, a model for Agile software development is already shown in the Figure 3. The process of extending Agile practices and ideas to numerous teams or even entire organisations is known as scaling Agile. Agile can be scaled using a variety of frameworks and methods; which one is best depends on the organization's needs and environment. Let us discuss some of the key points of various scaling methods:

- Scrum of Scrums - This is a straightforward method where representatives from each team independently practise Scrum and then convene on a regular basis (typically daily or weekly) to coordinate their work and discuss any dependencies or difficulties.
- SAFe (Scaled Agile Framework) - One of the most extensively used scaling frameworks is SAFe. It offers a thorough framework for growing Agile throughout the company, complete with roles, responsibilities, and rituals. Levels like team, programme, major solution, and portfolio are included in SAFe.
- LeSS (Large Scale Scrum) - LeSS is a straightforward strategy for expanding Scrum. It focuses on streamlining organisational structure and procedures to increase agility and reduce pointless complexity.
- Nexus - Scrum scaling is the sole purpose of the Agile framework Nexus. By adding further roles, activities, and artefacts to Scrum, it enhances the ability of numerous Scrum teams to collaborate successfully.
- DAD (Disciplined Agile Delivery) - DAD is a paradigm for making process decisions that provides a toolkit of Agile and lean practises, such as Scrum, Kanban, and others. It offers suggestions on how to modify Agile methodologies to match the circumstances of your organisation.
- Agile at Scale - Based on their particular requirements and limitations, some organisations decide to develop their own tailored strategy for expanding Agile. Although thorough preparation and testing are required, this can be beneficial.

The most significant agile software development methodology is called "extreme programming," and it is a software engineering methodology [19,20]. Extreme Programming, like other agile approaches, differs from conventional methodologies primarily in that it prioritises adaptability above predictability [19]. Advocates of extreme programming see continuous requirements changes as a natural, necessary, and desirable aspect of

software development projects;[21] they believe it is more practical and preferable to adapt to changing requirements at any point in the project rather than attempting to define all requirements at the beginning of a project and then struggling to control changes in requirements [19]. Extreme Programming outlines a set of daily practices for managers and developers that should uphold and advance specific concepts. Theoretically, putting these ideas into reality will lead to a more flexible ("agile") development process than utilizing conventional procedures, which are classic methods of software engineering taken to so-called "extreme" levels [22].

Scrum is a technique employed by the Agile approach that was established utilizing lean manufacturing ideas. Scrum software development is described as an incremental and iterative project management strategy that reduces risk and ensures predictable project execution [24]. The core premise of this system is empirical process control. Scrum uses timed events to organize and execute projects. Scrum meetings are held to analyze artifacts and put new solutions to the project's difficulties into action. These programs are intended to encourage research, adaptation, and openness in the development process [25].

Kanban is the name given to the just-in-time delivery method used in software development project management. The Kanban technique focused on the work that needed to be done and its deadlines [26]. To do this, the jobs are prioritized, the procedure is detailed, and the delivery lead time is specified. The Kanban is adaptive since it prioritizes and focuses on the most important project activities [27]. Scrum and Kanban are said to be similar because they are both lean and have the ability to break work down into smaller pieces, have self-organized teams, focus on producing usable software frequently, adapt to changes quickly, limit Work in Progress (WIP), use pull scheduling, and use transparency [28]. Kimber et al. [29] provide a thorough summary of the project's scope as well as tools for improvement. Furthermore, they place a great importance on the project's value and completing the deliverables. Kanban and Scrum differ greatly in execution methods, iteration criteria, methods for allocating work to members of cross-functional teams, whether WIP is limited, and how work is split within established time constraints. There are some differences, like whether cross-functional teams are necessary, whether time-boxed iterations are necessary, whether a team commits to a certain amount of work for a given product iteration, whether WIP is limited, and whether work needs to be divided up to be finished within a certain amount of time. These differences show the variety of contexts in which Scrum or Kanban are effective [29]. The Table 1 below summarizes the common determinants of scaling in Agile techniques.

TABLE I: THE SUMMARY OF COMMON DETERMINANTS OF SCALING IN AGILE TECHNIQUES

FACTORS	COMMON TECHNIQUES IN VARIOUS AGILE SCALING METHODOLOGY
Principles	All techniques adhere to the mindset, values, and principles outlined in the Agile Manifesto. Based on the necessity for scaling Agile, DAD, LeSS and SAFe improve the Agile Manifesto to their methodologies.
Process	Each approach has a specific process. The Agile team uses Scrum. Scrum, XP, DevOps, and KANABN are used by SAFe. Scrum is mostly used by LeSS, Nexus, Spotify, and Scrum of Scrum in their methodologies. DAD uses Scrum but adds other Agile techniques.
Organization	The scrum, coordination, and support teams constitute the bare minimum team structure for expanding the Agile technique. Every technique has a set team structure.
Business Requirements And Work Product	Each Agile team creates a single work product. The integrated work product is produced by the entire team.
Implementation Approach	The scope of the implementation extends from a single project to the corporate level.

Based on the above table of summary, each scaling method can be classified as shown in the Table 2.

Table 2 lists the common ground that popular scaling Agile techniques like SAFe, LeSS, DAD, Spotify, and Scrum of Scrums share.

The Figures from 5 to 10 is a diagrammatic view of the various scaling methodology.

- a) SAFe - SAFe, which stands for the Scaled Agile Framework, is an approach to scaling Agile principles and methods for large organizations, including those in the field of software engineering, mentioned in *Figure 5*. SAFe provides a framework and set of practices to help organizations adopt and implement Agile at scale, facilitating collaboration, alignment, and delivery across multiple Agile teams. SAFe is a comprehensive framework that encompasses various roles, activities, and artifacts to enable large organizations to effectively develop and deliver software and systems.
- b) LeSS - Large-Scale Scrum (LeSS) is an Agile framework designed to extend the principles and practices of Scrum to large organizations and address the challenges of scaling Agile software development. LeSS focuses on simplicity and the fundamental principles of Scrum, avoiding unnecessary complexity mentioned in *Figure 6*.

TABLE II: SUMMARY OF VARIOUS SCALING METHODS BASED ON COMMON DETERMINANTS

Scaling methods Factors	SAFe [30]	LeSS [30]	DAD [30]	Nexus [30]	Spotify [30]	Scrum of Scrums [30]
PRINCIPLES	Mindset, Values, Principles	Principles	Principles	Scrum Value and Agile Manifesto	Agile Manifesto	Scrum Value and Agile Manifesto
PROCESS	Agile product Delivery	First for a small set of teams and then to a larger group	Disciplined DevOps, Disciplined Agile Delivery as the bottom process, Disciplined Agile IT	Utilising the linked components of the Scrum framework	Using enhanced methods in Scrum	Scrum, Scrum of Scrums
ORGANIZATION	Agile Team, Business Product manager and testers	The LeSS Structure is arranged according to customer value, organizational structure, and communities and includes teams, scrum masters, and feature teams.	Principal Positions: Stakeholder, Team Lead, Product Owner, Team Member, and Architectural Owner. Supporting roles: Integrator Technical Expert, Specialist, Domain Expert, and Independent Tester.	Nexus positions include members of the Nexus Integration Team, Product Owners inside the team, Scrum Masters within the team, and the Nexus Integration Team itself.	Tribes, Squads, Chapters and Guilds.	Scrum of Scrums
BUSINESS REQUIREMENTS AND WORK PRODUCT	Program increment	Product backlog, Potentially Shippable Product Increment	Release solution	Product Backlog, Integrated Increment	Work product	Work product
IMPLEMENTATION APPROACH	Essential SAFe, Large Solution, Portfolio	LeSS, Less Huge	Disciplined DevOps, DAD as the bottom process, DAE and DAIT	Single product development	Single product development	Single product development

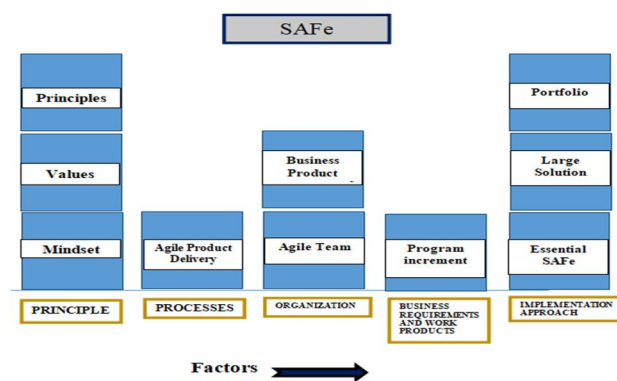


Figure 4: The Scaling Method – SAFe

- c) DAD - Disciplined Agile Delivery (DAD) is a process framework and toolkit for software development and delivery. It offers a pragmatic and context-based approach to software engineering that incorporates

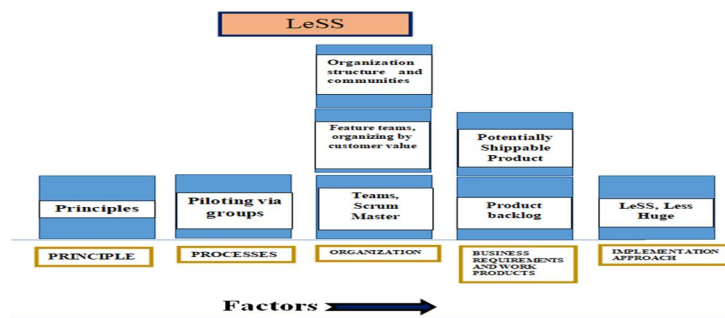


Figure 5: The Scaling Method – LeSS

a wide range of Agile and lean practices mentioned in *Figure 7*. DAD is designed to help organizations make informed decisions about which practices to adopt based on their specific needs and constraints.

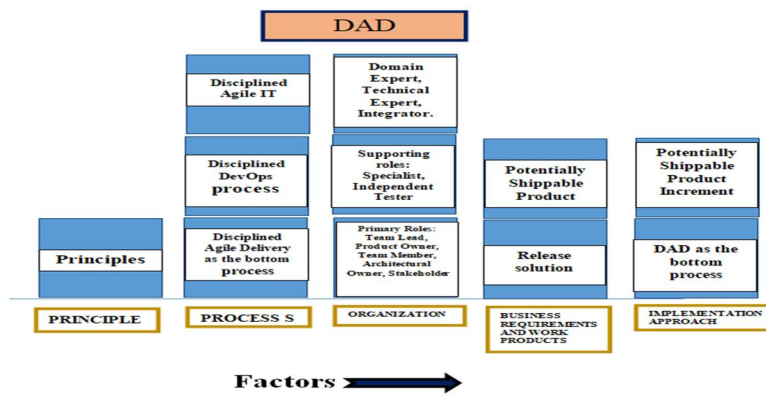


Figure 6: The Scaling Method – DAD

- d) Nexus - Nexus is a framework for scaling Scrum, specifically designed to help organizations coordinate the work of multiple Scrum Teams working on a single product. It is part of the Scrum.org suite of frameworks and tools for Agile and Scrum practices mentioned in *Figure 8*. The primary goal of Nexus is to enable the effective development of large and complex software products while maintaining the core principles and values of Scrum.

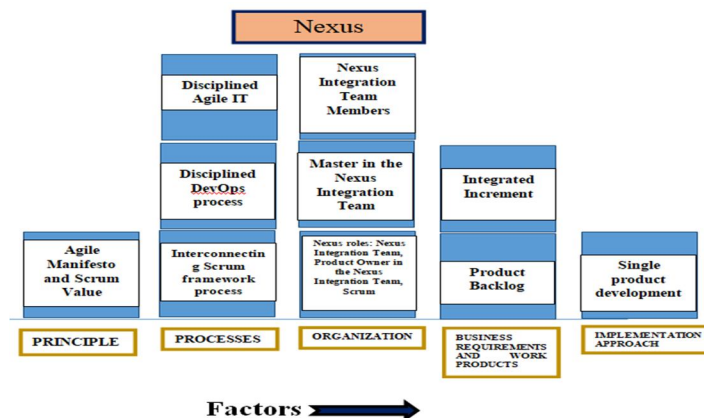


Figure 7: The Scaling Method – Nexus

- e) Spotify - Spotify, in the context of software engineering, refers to the Spotify Model, which is an organizational framework for managing and scaling Agile software development and delivery within a company. The model was originally developed and implemented by Spotify to enable their rapid growth and maintain a high degree of agility and innovation. It has since gained popularity and has been

adopted by other organizations looking for a flexible approach to scaling Agile practices mentioned in Figure 9.

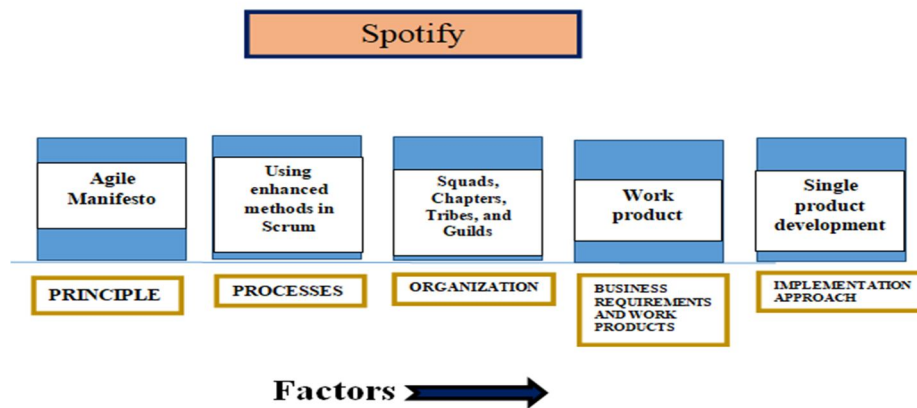


Figure 8: The Scaling Method – Spotify

- f) Scrum of Scrums - Scrum of Scrums (SoS) is a scaled Agile framework used in software engineering and project management to manage the work of multiple Scrum Teams that are collaborating on a larger project or product. It's designed to address the challenges of coordination and communication when multiple Scrum Teams need to work together to deliver a complex solution. Its principles are mentioned in Figure 10.

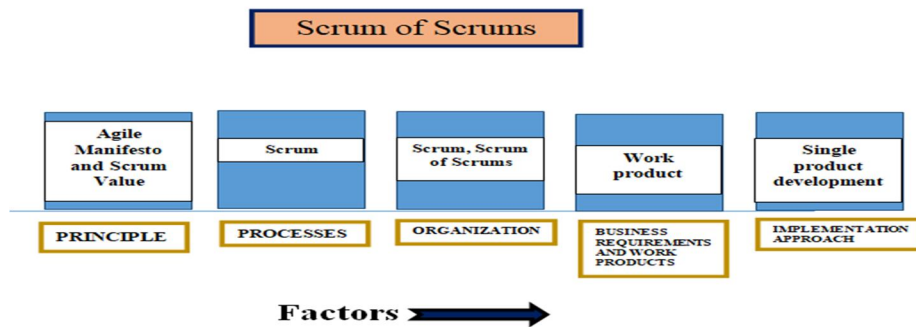


Figure 9: The Scaling Method – Scrum of Scrums

VI. CONCLUSION

The study aims to understand the feasibility of adopting Essence Architecture in Agile Methodology in software industries. Because it already includes the Agile approach, the expected scope of this Agile method is up to Agile scaling. The idea, business requirement, method, work output, organisation, and execution strategy are the same ground considered. Organisations can gain important advice for creating their Agile techniques by adopting this approach. Agile is employed in many business sectors. Agile development is renowned for handling complicated problems and swiftly adjusting to business changes by working with clients in the age of fast transformation. Because project management requires a more flexible and dynamic approach, agile project management is believed to be able to handle this challenge. Today's businesses heavily utilize the Scrum and Kanban approaches as powerful software development methodologies. Although there has long been debate about which of these approaches is best, there hasn't been enough statistical analysis-based evidence to support a conclusion. The study mainly aims to determine whether the influence of the Extreme Programming, Scrum and Kanban methodologies on various facets of software development project management differs. It can be concluded from the study that though the Scrum and Kanban methodologies play an upper hand when compared to Extreme Programming due to its versatility in nature of execution, that does not lay a firm foundation to conclude that these methods are the best. Industry has to make a thorough study on which methodology will suit best to meet their requirements before they start implementation. Considering the various factors including the project risk, the best suitable methodology can be adopted.

REFERENCES

- [1] PMI. 2017. PMI's PULSE of the PROFESSION 9TH Global Project Management Survey
- [2] McKinsey & Company. 2018. Agile compendium. October (2018)
- [3] D. Rigby, J. Sutherland, and A. Noble. 2018. Change management - Agile at Scale. Retrieved from <https://hbr.org/2018/05/Agile-at-scale>
- [4] Project Management Institute. 2017. Agile Practice Guide. Project Management Institute, Inc
- [5] I. Jacobson, P.W. Ng, P.E. McMahon, I. Spence, and S. Lidman. 2012. The Essence of software engineering: The SEMAT kernel. *Queue* 10, 10 (2012), 40–51.
- [6] Scrumguide.org. 2017. The Scrum Guide. Retrieved from <https://www.scrumguides.org/scrum-guide.html>
- [7] K. Back. 2005. Extreme Programming Explained Embrace Change. Addison-Wesley Professional.
- [8] D. Rose. 2018. Enterprise Agility. Jon Wiley & Sons, Inc.
- [9] Scrum.org. 2019. Scrum Master Trends. (2019)
- [10] D. Leffingwell. 2020. Scaled Agile Framework. Retrieved from <https://www.scaledAgileframework.com>
- [11] Project Management Institute. 2023. Disciplined Agile. Retrieved from <https://www.pmi.org/disciplined-agile>
- [12] VersionOne. 2019. The 13th annual STATE OF AGILE Report - 2018. CollabNet | VersionOne 13, (2019)
- [13] D. Car-Pusic, I. Marovic, and G. Bulatovic. 2020. Development of a Hybrid Agile Management Model in Local Self-Government Units. *Technical Gazette*. <https://doi.org/10.17559/TV-20190205140719>
- [14] S. M. R.Masud, M. Masnun, and M. A. Sultana, "DevOps Enabled Agile: Combining Agile and DevOps Methodologies for Software Development", *International Journal of Advanced Computer Science and Applications (IJACSA)*, Volume 13, No. 11, 2022
- [15] M. Alqudah and R. Razali. 2016. A review of scaling agile methods in large software development. *Int. J. Adv. Sci. Eng. Inf. Technol.* 6, 6 (2016), 828–837. DOI=<https://doi.org/10.18517/ijaseit.6.6.1374>
- [16] OMG. 2018. kernel and Language for Software Engineering Methods (Essence). *Object Manag. Gr. Versión 1.2* (2018), 300. Retrieved from <https://www.omg.org/spec/Essence/1.2>
- [17] Semat.org. 2020. Software Engineering Methods and Theory. Retrieved from [semat.org](https://www.semat.org)
- [18] K. Peffers, T. Tuunanen, Rothenberger, M.A., and Chatterjee, S. 2007. A design science research methodology for information systems research. *Journal of Management Information Systems* 24, 45–77. DOI=<https://doi.org/10.2753/MIS0742-1222240302>
- [19] Al-Gasaymeh, A. S., Kaddumi, T. A., & Qasaimeh, G. M. (2021). Measuring risk eExtreme Programmingsure in the banking sectors: Evidence from gulf cooperation countries. *Journal of Financial Economic Policy*, doi:10.1108/JFEP-01-2020-0008
- [20] The Agile Alliance Home Page. <http://www.agilealliance.org/home>
- [21] S Barakat, K Yaghi, Khalil Yaghi. Extreme programming in mobile applications. *The Seybold Report*, ISSN 1533-9211
- [22] Aydınlyurt, E. T., Taşkın, N., Scahill, S., & Toker, A. (2021). Continuance intention in gamified Mobile Applications: A study of behavioral inhibition and activation systems. *International Journal of Information Management*, 61, 102414.
- [23] Chaitanya Arun Sathe, Chetan Panse. An Empirical Study on Impact of Project Management Constraints in Agile Software Development: Multigroup Analysis between Scrum and Kanban
- [24] Anderson, D.J., Concas, G., Lunesu, M.I., Marchesi, M.and Zhang, H. (2012), "A comparative study of Scrum and Kanban approaches on a real case study using simulation", In *International Conference on Agile Software Development*, Springer, Berlin, Heidelberg, pp. 123-137.
- [25] Oliveira, E.R. et al. (2023), "Scrum method assessment in Federal Universities in Brazil: multiple case studies", *Brazilian Journal of Operations and Production Management*, Vol. 20, No. 1, e20231496. Doi: <https://doi.org/10.14488/BJOPM.1496.2023>
- [26] Alqudah, M.and Razali, R. (2017), "A comparison of scrum and Kanban for identifying their selection factors", *The IEEE 6th International Conference on Electrical Engineering and Informatics (ICEEI)*,pp. 1-6.
- [27] Ozkan, N., Bal, S., Erdogan, T.G. and Gök, M.Ş.(2022), "Scrum, Kanban or a Mix of Both? A Systematic Literature Review", In *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, IEEE, pp. 883-893. Doi: <https://doi.org/10.15439/2022F143>
- [28] Marnada, P., Raharjo, T., Hardian, B.and Prasetyo, A. (2022), "Agile project management challenge in handling scope and change: A systematic literature review", *Procedia Computer Science*, Vol.197, pp. 290-300.
- [29] Kniberg, H.and Skarin, M. (2010), "Kanban and Scrum-making the most of both", available from: http://www.agileinnovation.eu/wordpress/wpcontent/uploads/2010/09/KanbanAndScrum_MakingTheMostOfBoth.pdf
- [30] Teguh Raharjo, et.al, "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.