The delivery of large-scale software products through the adoption of the SAFe framework

Cristian FAGARASAN, Ciprian CRISTEA, Maria CRISTEA, Ovidiu POPA, Cristina MIHELE, Adrian PISLA
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
office.cristif@gmail.com

Abstract— **The paper analyses the Scaled Agile Framework (SAFe) which is currently one of the most important agile scaling frameworks, being adopted by the software development companies to maintain a competitive advantage in the digital age and, at the same time, it represents a way to involve all the organization departments in delivering technology-based solutions. Through this framework, functions like marketing, sales, legal, finance adopt Agile and Lean practices to achieve greater transparency and align project objectives with business goals. As every business is becoming a software business, building solid and automated models is essential to enable and sustain a large-scale software delivery. The SAFe adoption steps are analyzed to identify the prerequisites needed for the organizational transformation. This paper provides a case study presenting the implementation of SAFe at a global software development company. This study provides relevant information for both practitioners and academics. The examined framework is based on robust theories and offers tools that software companies can use to increase the efficiency of their software delivery model. Even if the assessed framework is partially adopted within an organization, its core principles, tools, and techniques enhance the software delivery practices.**

Keywords—change management; agile; lean; scaled agile; software development; SAFe

I. INTRODUCTION

Enterprises operating within the Information Technology (IT) industry have been dealing with change regularly, influenced by the technological landscape and the market shifts [1]. Few companies managed to keep up with the market changes, such as Amazon and Alibaba, which now define the technological landscape, causing other software development organizations with similar software platforms to lose market share and clients [2]. Organizations still have time to realize that a transformation is needed. By leveraging the infrastructure built in the past and by adjusting the managerial frameworks used, businesses can protect what they have built throughout the past ages. Studies show that organizations are currently looking for support on Scaled Agile implementation methods as it is seen as a potential solution to achieve business agility [3]. It has been more than twenty years since the creation of the Agile Manifesto, and still, some companies are not familiar with the Agile methodology. However, the pandemic played a vital role in the growth of Agile adoption

within software development teams, increasing from 37% in 2020 to 86% in 2021 [4]. Within the IT industry, it is believed that Agile is the solution to achieve the desired business outcomes. Figure 1 shows that even if the adoption of Agile is increasing within the Software Development departments, the other company areas are not a priority for most organizations.
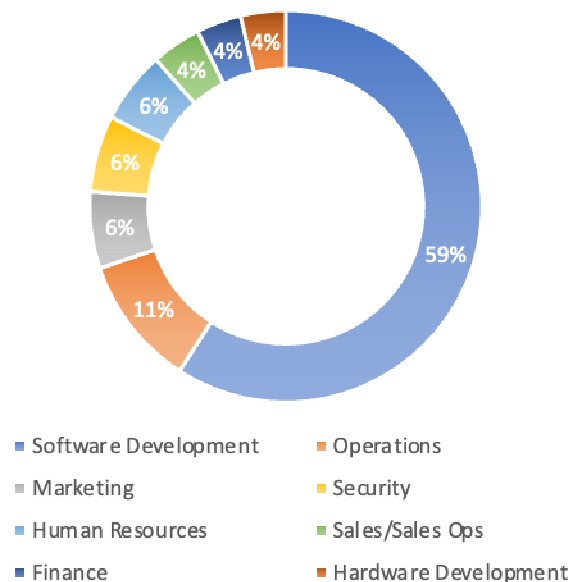

Fig. 1. Agile adoption across organization functions [4]

The non-IT departments started to adopt the Agile mindset, doubling in the last year. Nevertheless, according to a survey run by the Digital.ai team, some barriers block Agile adoption across non-IT departments. Respondents identified that the main barriers are inconsistencies in processes and practices, cultural clashes, and resistance to change [4]. According to Leffingwell [5], for organizations that continue to grow and expand, the Agile methodology complexity and challenges arise in areas like coordination of global and multiple teams, emerging architecture, requirements analysis, and documented specifications. Every organization, no matter in which industry

it operates, is technology-dependent, and in the following years, every organization will become a software company.

Software companies need to remain competitive in the current emerging market. This drives the enterprises to digitally transform their operations and business solutions by implementing new frameworks that are customer-centric.

The objectives of the current paper are to:

1. Present an overview of the Agile delivery frameworks that are currently used in the IT industry to build software products, and which proved to be successful for many companies operating in this domain. This can help companies newer to Agile to get an understanding about the industry trends, offering insights about the Agile way of working.

2. Analyze the Scaled Agile Framework (SAFe) and its competencies, highlighting the SAFe configurations, roles, and artifacts.

3. Study the transition steps from the current way of working to the SAFe adoption.

4. Examine the SAFe implementation through a case study, explaining the business problem that drove the adoption of SAFe and present its results.

This paper emphasizes the importance of delivering customer value as efficiently as possible using a continuous delivery model to help companies scale their existing software delivery. SAFe is designed to introduce Lean-Agile principles enterprise-wide and is more comprehensive than any existing methodology for large-scale development. Practical steps are introduced that can assist organizations in adopting the SAFe framework as a new way of developing software.

II. DELIVERY FRAMEWORKS USED IN SOFTWARE DEVELOPMENT INDUSTRY

According to Susanne Madsen [6], there is a core difference between management and leadership. Figure 2 shows that traditional project managers are found on the left-hand side of the matrix, focusing on tasks and authority. On the contrary, on the matrix's right-hand side can be seen that leaders focus more on inspiring and coaching people.

Fig. 2. The project leadership matrix [6]

The next generation leaders should learn from past mistakes and strive to improve continually by accepting that solutions come from changing behaviors and the way of work. The Agile software methodology focuses on people with a great emphasis on collaboration, alignment, enhanced communication, and team empowerment.

The Agile Manifesto [7] states that individuals and interactions are valued more than processes or tools, adapting to change is more important than following a plan and customer collaboration comes before contract negotiation. These are just a few out of twelve principles based on which the Agile Manifesto was created. According to the Digital.ai annual report [4], 66% of Agile teams use Scrum, 9% use Scrumban, a hybrid delivery method between Scrum and Kanban, and 6% of the teams use pure Kanban.
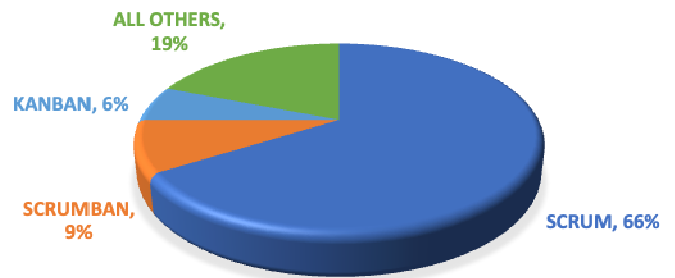

Fig. 3. Agile approaches employed by teams in software development [4]

A. Scrum

Starting with 1986, Takeuchi and Nonaka began to study the benefits of sequential versus overlapping development phases. They referred to Scrum as the rugby approach due to the new concept of having cross-functional teams instead of skill-specific organized teams [8]. Most software development teams prefer to adopt Scrum as a framework to introduce more flexibility in their development process. The main goal of this delivery method is to limit the rigidity of the specifications, increase collaboration and trust within the product development team, and involve stakeholders early and frequently in the development process [9].
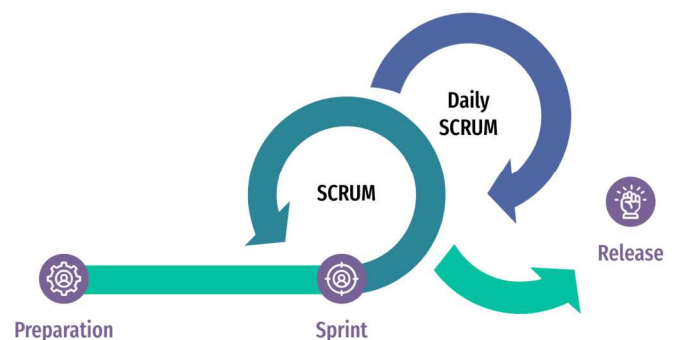

Fig. 4. Scrum framework [10]

The process is lightweight and easy to understand, which will assure a quick and effective implementation. Figure 4 shows all the Scrum Roles, Artifacts, and Events. Scrum is based on time-boxed iterations, known as sprints, having a clear goal

and scope. The incremental delivery model can reduce the time-to-market of software products and potentially positively impact the organization's revenue. All the built-in events ensure transparency, early implementation of feedback, and faster reactions [11].

B. Kanban

The Kanban delivery method has its origins in the Lean and Agile methodology and is widely spread in the manufacturing industry, focusing on eliminating waste and continuous improvement [12]. The IT industry adopted the Kanban method to provide a work management system for the development of software products. It is based on principles like visualizing the workflow, setting limits for the work that is in progress, organizing inspect and adapt sessions to support continuous feedback [13].

C. ScrumBan

The ScrumBan delivery method is becoming more popular with teams who do not want to develop products in iterations but still use the time-boxed Scrum events. By combining the values and principles of Scrum and the work management system used in Kanban, ScrumBan provides a hybrid delivery method that addresses some of the Scrum limitations by introducing more flexibility into the software development framework in regard to team size and roles, team members expertise and skill and documentation [14].

III. DELIVERING SOFTWARE PRODUCTS AT SCALE – THE SCALED AGILE FRAMEWORK

The Agile methodology was intended for small teams, ranging from 3 to 9 people, ideally located in the same room to enhance collaboration which was equipped to deliver functionality that was not seen as critical. Nevertheless, it was proven that it is possible to adopt agile at scale involving global teams [15]. Organizations continue to grow, and the world is changing uncontrollably, forcing the leaders to rethink the enterprise's structures. According to Kotter [16], the new organizations need to operate on a dual system that can introduce the much-needed business agility, powered by the active agents of change. All organizations in a mature state must have a well-designed hierarchy in place that helps with the daily operations. Furthermore, if managerial processes exist and are transparent and clear to everyone, employees will perform with increased performance. Almost all industries face a digital transformation and are in the process of becoming software businesses. Achieving business agility means that all the departments of the organization, even marketing, finance, legal, are engaged in delivering innovative business solutions by using Lean and Agile practices with a relentless improvement mindset to meet customer demand faster than the competition.

According to Leffingwell [17], the key benefits of the SAFe adoption can include improvement in engagement, time-to-market, quality, and productivity:

- 25-75% improvements in quality.
- 20-50% increase in productivity.

- 30-75% increase in the time-to-market.
- 10-15% happier and more motivated employees.

A. Core Competencies

Figure 5 shows all capabilities that need to be understood to implement the SAFe framework. Together, their role is to introduce business agility in any organization and deliver value in the shortest lead time. All competencies focus on customer-centricity, which along with Design Thinking, are in the center of the scaled delivery framework. Through the Lean-Agile Leadership competency, the enterprise leaders and managers are responsible for the success of the organizational change. They must lead by example and empower teams to reach their full potential by promoting a Continuous Learning Culture. Through Organizational Agility, enterprises and Agile teams optimize their managerial and business processes, even their strategy, to enhance the business opportunities that emerge.

Fig. 5. SAFe core competencies [17]

The Lean Portfolio Management competency is needed to align strategy, funding with execution and offer governance of the organization. Agile high-performing and cross-functional teams build business solutions that please the customers by leveraging the Team and technical agility competency. In the center of product delivery, the Agile model adds the customer to the product strategy, and development iterations guarantee predictable deployment of value into production but release on demand. The Enterprise Solution Delivery focuses on system engineering innovation and practices to build and deploy complex, large systems that must always be fully operational, not impacting the organization financially.

B. SAFe Configurations

The framework was built in a sustainable and configurable way to be adapted by the business needs. Four configurations can support systems that require a smaller number of teams to systems that are being developed by hundreds of teams.

1) Essential SAFe: This is the minimal configuration, and it has a reduced number of teams, roles, and events. It is often called the basic building block of SAFe [16]. Business and technical teams are organized in an Agile Release Train (ART), a fundamental structure in SAFe.

2) Large Solution SAFe: The large solution includes Essential SAFe, and it is for enterprises that build large

applications and other systems. The Solution component is introduced to integrate more value streams that have multiple ARTs.

3) Portfolio SAFe: The Portfolio configuration provides business agility through its competencies and practices. This solution includes the Lean portfolio management competency, which aligns strategy, funding, and execution.

4) Full SAFe: The Full solution is the most complex configuration of SAFe. It is built to support hundreds of people developing colossal systems with countless integrations and dependencies. The largest enterprises use this as a solution to maintain and implement large software products. However, complex solutions may require more SAFe instances.

5) SAFe for Government: Lean-Agile practices can be implemented in the public sector. SAFe offers a set of templates and success patterns that can be used for successful SAFe adoption in the public sector.

C. Roles and Artifacts used for implementing Essential SAFe

The fundamental structure in Essential SAFe is the ART, which represents a business value stream, and it usually has between 5 to 12 cross-functional agile development teams. The roles encountered on the Agile Release Train are as follows:

- Release Train Engineer (RTE) – acts as a Scrum Master for the train.

- Product Management (PM) – owns and prioritizes the Program Backlog.

- System Architect/Engineering – provides architectural guidance to the train teams.

- System Team – provides the means to integrate and evaluate the completed ART work.

- Business Owners – critical stakeholders of the ART.

Each Agile Team part of the ART has the following Scrum Roles: Agile Team, Scrum Master, Product Owner. In the end, in Essential SAFe, we have two operating levels: the Team and the Program ones.

IV. SAFE IMPLEMENTATION STEPS

Any change in the organization can be a tedious process. There will be a need to define and identify gaps and flaws in the current software delivery system and understand how adopting SAFe will improve the current ways of working. It is recommended to use SMART objectives to track the adoption process and make sure company goals are achieved when the adoption is complete.

A. Identify the training needs and create a SAFe Center of Excellence

The analysis that was done to determine the flaws of the current ways of working can help identify the organization members that need to be involved in the adoption process. SAFe requires that the whole team is involved and aligned. It is recommended to start training from the top and train the current leaders to be SAFe change agents. This means that Executives, Managers should become evangelists of the SAFe framework. In this way, the probability of a successful adoption is increased and their ability to provide solutions to other team members will be augmented.

B. Identify the value streams and the ARTs

One of the main steps to adopt SAFe is to identify the backbone of this framework: the business Value Streams and ARTs, which are the primary elements through which value is delivered in SAFe. There are two types of value streams: operational and development. The latter creates the solutions that the operational value stream will use to deliver end-user value. The solutions the Operational Values Stream uses to provide value to the end-users need to be identified so the development value streams can be mapped. The final step is defining the ART, which is recommended to have between fifty and one hundred twenty-five people and a focus on a solution, product, or service. Depending on the situation, there are cases where an ART can support one or more value streams. Another scenario would be where there are multiple ARTs on a single value stream.

C. Create a timeline for the implementation

For a change of this magnitude, a rigorous implementation plan has to exist. Time needs to be dedicated to planning and strategizing, but with an incremental approach in mind. Creating the plan should be similar to how the development iterations are planned in small increments. Now is the time for the first value stream to be selected, and implicitly the ART. After the Value Stream canvas is completed and the first ART is confirmed, the Product Increment (PI) roadmap will be created. One product increment usually takes ten weeks, and it is recommended to have a roadmap for the following three PIs.

D. Prepare the first ART launch

During this phase, the ART is defined, and the cadence of the program increments is decided. The Agile teams and additional roles are established. According to the implementation roadmap, any planned training is provided for new ART roles. In the meantime, Product Management prepares the program backlog for the first Product Increment planning.

E. Provide final training and launch the ART

In this stage, the focus is on the Agile teams and delivering any training needed to fully understand the new team roles. The first ART will begin with a PI planning in which all ART team members will participate to draw a plan for the next ten weeks. This is when all the dependencies are mapped, and any question is answered. After the ART is launched, experienced team members must offer guidance, and coaching about ART execution is provided.

V. CASE STUDY: SAFE IMPLEMENTATION CHALLENGES AND SOLUTIONS IN A GEOGRAPHICALLY DISPERSED COMPANY

The current case study analyzes the implementation of the Scaled Agile Framework in a software company that has a geographically distributed workforce with clients in more than one hundred countries. The adoption was driven by the company's vision to explore better and quicker ways of working that could potentially accelerate the product development delivery.

A. Identified challenges for geographically dispersed companies that want to adopt SAFe

For this analysis, there were two factors that were taken into consideration:

- The organization **standard operating procedures** employed in terms of methodologies, procedures and tools used to develop software products.

- The organizational **culture and geographical location**.

Figure 6 presents the identified challenges based on the above factors. The list includes common problems encountered in a global organization when trying to standardize and scale the current ways of working.

Fig. 6. Identified challenges for geographically dispersed companies that would like to adopt SAFe

B. Applied solutions of the SAFe framework used to overcome the identified challenges

The SAFe workflow patterns and principles promote proper orientation, alliance, and delivery synchronization, and can be utilized to overcome previously presented challenges. The following steps were followed to overcome the problems presented as part of the Figure 6:

1) Align processes between scrum teams: SAFe processes were used to achieve team alignment. The simplified process applied was split into three main phases: project initiation, discovery, and implementation through continuous improvements. The main activities of this solutions were the following:

- Project Initiation, develop the vision and the Program Feature Backlog.

- High-level Release Planning, continuous feature elaboration, creating and revising the roadmap, leading to a Revised Release Planning and PI Objectives.

- Splitting the Program Feature Backlog into multiple teams' backlog and starting the PI iterations. Next, all teams work needs to be integrated and proceed to the Release stage.

2) Ensure delivery alignment by developing on cadence: overcoming this is ensured through synchronization. All teams' iterations were synchronized and grouped into common Program Increments. The target should be to have all iterations and PIs synchronized across the organization.

3) Integrate Agile and Waterfall: the first step is to separate the features based on the delivery method: Agile and Waterfall. Next, through a release planning exercise, the teams' delivery needs to be synchronized as ilustrated in Figure 7.
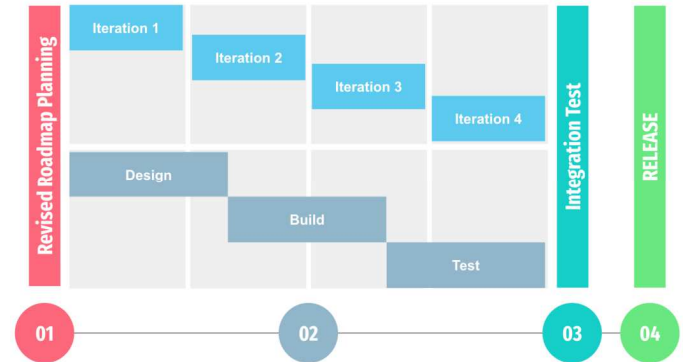

Fig. 7. Delivery synchronization between Agile and Waterfall teams

4) Performance and status tracking alignment: the "Definition of Done" and the estimation method need to be consistent for all scrum teams. Aligning on the same Definition of Done aims to have a shared understanding across all scrum teams of what "Done" means. In regards to the estimation method, it is usual in Agile for scrum teams to use the relative estimating method and implicitly, story points. Because of the nature of relative estimations, the next steps were followed to achieve alignment of story point sizes:

- A set of story types was created for each team, relevant to each team backlog.

- Equivalent stories were created across all scrum teams.

- Using the equivalent stories as guidelines, story point sizes were extrapolated for other stories.

Analyze how the current organization structure can support SAFe activities: *the organization structure of the program is presented in Figure 8.* The challenge here is that the local and the remote teams are on multiple time zones, which can be resolved through a solid communication structure and a good split between responsibilities. The client is in United Kingdom, the client's team is in India and the vendor is located in Romania. The team located in India had the following responsibilities:

- Release planning and estimations, to offered as an input in the PI planning session.

- Application support, maintenance, support ongoing deployments.

The team located in Romania had more teams, which leads to a greater number of responsibilities on the testing side:

- Business value definition and measurement. Backlog definition, release planning and estimation.

- Integration test planning and execution support, story verification, application testing.

- Application deployments and support. Agile coaching and training.
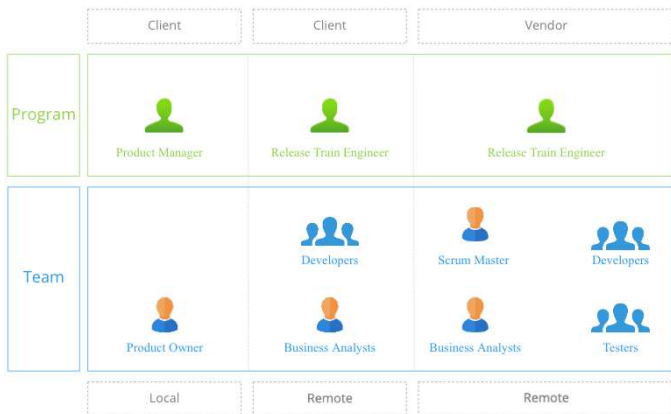

Fig. 8.   Organizational structure of the Program

The communication was done virtually, through dedicated collaboration tools like tele-conference, video conference to facilitate meeting between the local and remote teams. The team calendar was adapted to support daily standing calls where all the team members could participate. Enhancing the communication between Product Manager, Release Train Engineers, Product Owner, Business Analysts and Scrum Master roles was a key factor that increased the flow of information and contributed to the adoption success.

5) Integrate Development and Operations by promoting a DevOps culture: DevOps principles aim to enhance collaboration across development and operations by using automation techniques to achieve faster, predictable and frequent delivery of value to market.
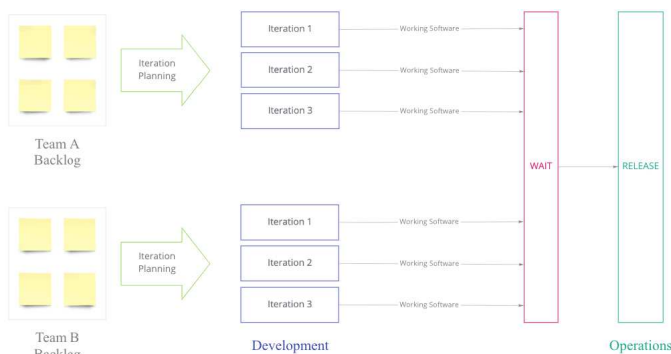

Fig. 9.   Development and Operations integration

The main DevOps principles that can be followed for achieving successful results are:

- Automate as much as possible, experiment frequently, and improve continuously.

- Delivery of small increments, test early and often and ensure a strong source control.

C. Agree on the success criteria for the adoption and set some KPIs to measure performance

It is vital to establish from the start what is the success criteria and how success will be measured. The focus should be on the tangible benefits that can be measured and for which we can set KPIs. Baselines should be obtained before starting the SAFe implementation. A few examples of metrics that were used mainly in this case were the:
- effort spent on software configuration management.
- time spent on release management and deployment.
- number of defects identified as part of the deployment and software integration processes.

D. Early benefits measured through the SAFe implementation

The results of the adoption were measured and determined based on the previous performance baselines. The **tangible benefits** were the following:

- 50% improvement in the integration and deployment of the code across various product releases, based on the actual effort tracked.

- 63% reduction in activities needed to support the software configuration management (manual trackers and environment sync-up removal, maintenance tools optimizations).

- 59% reduction in defects attributed to flaws of the previous software configuration management system (deployments and processes optimizations).

- 90% improvement in effort needed to raise deployment requests. Deployment requests are more optimized, the previous deployment process is now streamlined, a lot of the manual steps were automated and integrated in the deployment tool.

The **intangible benefits** that were observed are the following:

- Traceability increases from the Portfolio level through the Team level.

- Automated build and deployment of the code and real-time status monitoring.

- Seamless multi-team code integrations through the SAFe tooling system.

- Increase in teams' interaction and collaboration across the ART which lead to increased development efficiency.

VI. Conclusions

The paper describes the path that companies need to follow to transform their software delivery framework. If the Agile mindset and way of working is not yet adopted, this is the first step. The paper offers an overview of the widely embraced Agile delivery frameworks that can be considered. After the new method is standardized at a company level, the next step is to practice Agile at scale.

The case study presented in this paper aims to report the challenges of implementing the SAFe framework at a software company with a geographically distributed workforce. A solution inspired by the SAFe framework is proposed and implemented for each challenge, disclosing the steps and the benefits obtained. The tangible and intangible results are presented, and they can serve as an input or a success story for any enterprise considering scaling its delivery framework. The diminution of effort requested for supporting the software configuration management, streamlining implementation processes, traceability enhancement, and the contribution the framework has in strengthening teams' interaction and collaboration across the ART are among the main tangible and intangible benefits of implementing this framework at the software organization.

The SAFe adoption must start from top to bottom, from executives to managers and employees. The SAFe framework changes attitudes and behaviors and enhances skills. If people need to start and do things differently, leaders must shape the path through training and coaching. The SAFe framework provides the tools and techniques that help achieve alignment across the organization and a value stream focus. Many software development companies adopt agile scaling frameworks to adapt rapidly to a dynamic environment. Companies may implement SAFe configuration at different levels within the enterprise. The adoption can start at a team level, where the framework is very similar to Scrum, up to the highest level (portfolio). The higher the number of organizational departments, the more complex it becomes for the companies.

However, it is essential to mention that the adoption of SAFe can have a significant learning cost because of switching to different product development patterns. The authors intend to analyze the learning costs optimization of the SAFe framework adoption in a future study.

References

[1] C. Cristea, "Managementul de proiect in Microsoft Project 2016", UTPRESS, Cluj-Napoca, 2019.

[2] M. Kersten, "Project to Product", IT Revolution, IT Revolution, Portland, Oregon, 2019.

[3] M. Marinho, R. Camara, S. Sampaio, "Toward Unveiling How SAFe Framework Supports Agile in Global Software Development", IEEE Access, Volume 9, 2021, p109671 - 109692.

[4] Digital.ai, "15th Annual State of Agile Development Survey", 2021.

[5] D. Leffingwell, "Scaling Software Agility: Best Practises for Large Enterprises", Pearson Addison-Wesley, New Jersey, 2007.

[6] S. Madsen, "The Power of Project Leadership", 2nd edition, Kogan Page, London, 2019.

[7] Agilemanifesto.org, "Manifesto for Agile Software Development", Utah, 2001.

[8] H. Takeuchi, I. Nonak, "The New Product Development Game", Harvard Business Review, Boston, 1986.

[9] R. Plateaux, O. Penas, S. Mule, P. Hehenberger, S. Patalano, F. Vitolo, "Scrum++ Framework concepts", 2020 IEEE International Symposium on Systems Engineering, 2020, p1-8.

[10] J. Jordan, "My Scrum Diagram", https://jordanjob.me/blog/scrum-diagram, 2015.

[11] K. Schwaber, M. Beedle, "Agile Software Development with Scrum", Pearson Education, New Jersey, 2001.

[12] C. Cristea, "Managementul sistemelor de productie", Todesco, Cluj-Napoca, 2013.

[13] R. Wakode, L. Raut, P. Talmale, "Overview on Kanban Methodology and its Implementation", IJSRD, Volume 3, 2015.

[14] K. Bhavsar, S. Gopalan, V. Shah, "Scrumban: An Agile Integration of Scrum and Kanban in Software Engineering", IJITEE, Volume 9, 2020, p1626-1634.

[15] B. Fitzgerald, K. Stol, R. O'Sullivan, D. Obrien, "Scaling Agile methods to regulated environments: an industry case study", ICSE, 2013, p863-872.

[16] J.P. Kotter, "Accelerate", Harvard Business School Press, Boston, 2014.

[17] D. Leffingwell, R. Knaster, "SAFe Distilled", Pearson Addison-Wesley, New Jersey, 2020.