Journal of Architectural Engineering Technology
ISSN: 2168-9717
Research Article Open Access

Critical Success Factors in Large-Scale Agile Software Development

Anabel Milagros Salazar[1]*, Muhammad Qasim Rana[2], Ana C. Gonzalez Leyba[3], Mandeep Saini[4], Olugbenga Timothy Oladinrin[5] and Angela Lee[2]

[1]*Project Management Department, Energy Systems Catapult, United Kingdom*
[2]*School of Built Environment, University College of Estate Management, United Kingdom*
[3]*Accounting and Finance Department, Inter-American Development Bank, USA*
[4]*Salford Business School, University of Salford, United Kingdom*
[5]*School of Art, Design and Architecture, University of Plymouth, United Kingdom*

Abstract

This study delves into the burgeoning trend of deploying agile software development (ASD) on large-scale software projects, aiming to establish critical success factors (CSFs) to tackle associated implementation challenges. Through a comprehensive review of existing literature and a comparative analysis of the two organisations' experiences, this research identifies challenges and mitigation strategies for large-scale ASD implementation. The findings elucidate typical implementation phases and challenges encountered, demonstrating correlations between challenges and the implementation process. Ultimately, this study fills a gap in the academic literature by addressing challenges in large-scale ASD implementation, offering real-world insights and comparisons with existing literature to provide valuable recommendations.

Keywords: Agile Software Development; ASD; Implementation; Critical success factors (CSF)

Introduction

In recent years, there has been a growing interest in applying agile software development (ASD) methodologies to large-scale projects. ASD has become a focal point in Software Engineering, aiming to provide businesses with a faster and more adaptable software development process, particularly in the dynamic and ever-evolving software industry, including mobile applications. According to [1], since the emergence of the Agile Manifesto in 2001, software development (S.D.) practices have shifted towards embracing change, incremental delivery, and increased involvement of end-users in the development process. Initially, agile practices were predominantly associated with small-scale projects, believed to be suitable for small, single-team endeavours [2], typically with teams of 50 or fewer members [3].

However, the advantages of adopting agile methodologies have garnered attention for large-scale project developments [4], within large organisfations [5], and in cross-border team environments [6]. Implementing agile practices on a large scale presents unique challenges. These challenges may include fostering collaboration among teams, deficiencies in requirement analysis, and issues associated with distributed projects [5]. However, [7] assert that organisations that scale agile practices can reap numerous benefits, including enhanced innovation in day-to-day operations. Additionally, such organisations can adapt to changing conditions, develop responsive solutions, and mitigate critical situations. Furthermore, [6] maintains that agile can be just as effective in large-scale projects through appropriate methods and practices. [7] Studied several companies that have scaled agile practices. While companies like Spotify and Netflix have succeeded in this endeavour, many others struggle to scale Agile practices successfully.

With the increasing adoption of scaling agile practices in large organisations [5], coupled with the notable dearth of academic research and scholarly rigour in this domain, typically dominated by publications authored by practitioners and consultants, this study endeavours to provide recommendations for addressing the challenges associated with implementing large-scale agile software development practices. Two case studies of agile software development implementation in large-scale software organisations are presented, outlining practices that could facilitate broader adoption on a larger scale. Thus, this paper conducts an extensive literature review and juxtaposes case study findings to derive critical success factors (CSFs).

Background and literature review

Software development is a cornerstone in software engineering, forming the backbone of every software project. Its objective is to deliver flawless code to users within allocated timeframes and budgets while remaining adaptable to evolving user requirements. Within software development methodologies, two prominent approaches, 'waterfall' and 'agile,' emerge [8,9]. The waterfall methodology adheres to a linear progression, whereas the agile methodology prioritises flexibility and accommodates changes at each stage of project development. Agile methodologies are fundamentally incremental, operating on the premise that frequent minor releases yield more robust products than infrequent significant releases [8].

According to [9], agile encompasses a collective array of methods and approaches to enhance software solutions' relevance, consistency, flexibility, and business value, evolving over the past two decades. [10] Asserts that agile practices allocate 80% of their focus to ecosystems, individuals, personalities, collaboration, conversations, and relationships. However, agile is not merely a set of techniques or tools; it embodies a mindset embraced by individuals actively engaging with and implementing agile principles [10]. Similarly, [11] argue that agile

***Corresponding author:** Anabel Milagros Salazar, Project Management Department, Energy Systems Catapult, United Kingdom, E-mail: m.rana@ucem.ac.uk

Received: 01-March-2024, Manuscript No: jaet-24-130354, **Editor assigned:** 04-March-2024, PreQC No: jaet-24-130354 (PQ), **Reviewed:** 18-March-2024, QC No: jaet-24-130354, **Revised:** 23-March-2024, Manuscript No: jaet-24-130354 (R), **Published:** 29-March-2024, DOI: 10.4172/2168-9717.1000379

Citation: Salazar AMS, Rana MQ, Leyba ACG, Saini M, Oladinrin OT, et al. (2024) Critical Success Factors in Large-Scale Agile Software Development. J Archit Eng Tech 13: 379.

represents a holistic mode of thinking, transcending reliance solely on specific techniques or tools.

Large-scale ASD

The notion of 'agile development at scale' encompasses various interpretations [12-14] delineate two primary perspectives on scaling agile: tactical scaling and strategic scaling. Tactical scaling involves adapting agile delivery approaches to address scalability factors such as team size, regulatory requirements, and geographic dispersion. On the other hand, strategic scaling entails implementing agile practices across an organisation or its I.T. department. These perspectives are distinct yet interconnected; successful execution of tactical scaling is often a prerequisite for achieving strategic scaling.

Large-scale projects entail substantial risks and frequently encounter challenges such as delays, budget overruns, and even project failures [15-17] attribute this heightened risk to the complexity of governance systems in large-scale projects, which typically correlates with the number of teams involved in development. The greater the number of individuals and teams engaged in a large-scale project, the greater its inherent complexity.

Challenges when implementing large-scale ASD

According to Dikert [18], any organisational transformation involving many individuals will face challenges. Based on the annual State of Agile Survey [19], the most significant implementation challenges are related to organisational culture, particularly organisational culture-related resistance to change and deficiency in management support and sponsorship [19]. Figure 1 presents the challenges reported in the 2020 survey.

Moreover, Dikert [18] presented a systematic literature review on large-scale Agile transformations. 35 challenges were reported and classified into 9 categories, as shown in Table 1.

Challenges category subcategories

Difficulty in implementing Agile Misunderstanding agile concepts and lack of guidance from literature, Agile could have been customised better, reverting to the old way of working and excessive enthusiasm.

Change resistance General resistance, scepticism towards the new way of working, and top-down mandate create resistance and make management unwilling to change.

Lack of investment Lack of training, lack of coaching, too high workload, old commitments kept and rearranging physical spaces.

Coordination challenges in multi-team environments Difficulty interfacing between teams, autonomous team model challenges, global distribution challenges, and technical consistency are also challenges.

Different approaches emerge in a multi-team environment. The interpretation of agile differs between teams, and old and new methods are used side by side.

Hierarchical management and organisational boundaries, Middle managers' new role in agile is unclear; management is in waterfall mode, keeping the old bureaucracy and internal silos kept.

Requirements engineering High-level requirements management is largely missing in agile, requirement refinement are challenging, creating and estimating user stories is difficult, and there is a gap between long-term and short-term planning.

Quality assurance Accommodating non-functional testing, lack of automated testing and requirements ambiguity affect QA.

Integrating non-development functions in the transformation other functions are unwilling to change, and there are challenges in adjusting product launch activities, adjusting to the incremental delivery pace and rewarding model, and needing to be teamwork-centric. In essence, challenges can be considered a success factor (S.F.)-S.F.s is not measurements of success but rather something that needs to be done well to achieve objectives [20]. Therefore, a comparison of figure one and table one reveals that the most common S.F.s are general change resistance, scepticism towards Agile practices, lack of training, inconsistent processes and practices across teams and the pervasiveness of traditional mode.

Recommendations for implementing agile practices large-scale

According to [7], many managers preempt large-scale teams, which need help developing long-term projects. However, in essence, the number of agile teams that can be created and the size of the initiative are not limited. Leaders must, however, be practical. Organising every function into agile teams is inappropriate, as agile approaches are inadequate for certain activities. This view is supported by [13], who state that it is crucial to understand if agile approaches can be appropriate for large-scale as well as where and how. Therefore, customising agile methods has frequently been critical for agile implementation. Since every company will have its challenges with

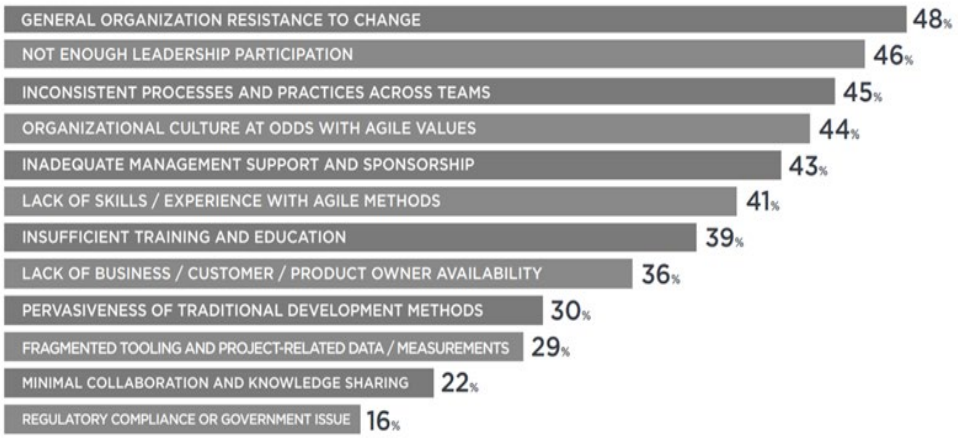

Figure 1: Agile Survey Challenges.

Table 1: Implementation Challenges.

Challenges Category	Subcategories
Difficulty in implementing Agile	Misunderstanding agile concepts and lack of guidance from literature, Agile could have been customised better, reverting to the old way of working and excessive enthusiasm.
Change resistance	General resistance, scepticism towards the new way of working, and top-down mandate create resistance and make management unwilling to change.
Lack of investment	Lack of training, lack of coaching, too high workload, old commitments kept and rearranging physical spaces.
Coordination challenges in multi-team environments	Difficulty interfacing between teams, autonomous team model challenges, global distribution challenges, and technical consistency are also challenges.
Different approaches emerge in a multi-team environment	The interpretation of agile differs between teams, and old and new methods are used side by side.
Hierarchical management and organisational boundaries	Middle managers' new role in agile is unclear; management is in waterfall mode, keeping the old bureaucracy and internal silos kept.
Requirements engineering	High-level requirements management is largely missing in agile, requirement refinement is challenging, creating and estimating user stories is difficult, and there is a gap between long-term and short-term planning.
Quality assurance	Accommodating non-functional testing, lack of automated testing and requirements ambiguity affect Q.A.
Integrating non-development functions in the transformation	Other functions are unwilling to change, and there are challenges in adjusting product launch activities, adjusting to the incremental delivery pace and rewarding model, and needing to be teamwork-centric.

agile implementation, when selecting agile approaches to implement, the focus of the business areas should be carefully considered [5,7] propose 3 significant steps to scale up Agile:

Leading agile by being agile: According to [7], when leaders have not understood and implemented agile practices themselves, they will attempt to scale up agile the same way they have approached other change initiatives: through top-down plans and directives, which may cause resistance [5]. Consequently, the trajectory is better when leaders act as an agile team [7]. In the same vein, an exciting success factor reported in the study by [18] was the absence of a top-down mandate since it allows grassroots levels of empowerment. This view is supported by [21], who state that even though the top-down approach achieved some business results, it is typically unable to rally the whole organisation. This is because the approach was "do this because it is an obligation" versus "do this because the value of Agile is understood."

Rolling out agile in a sequence of steps: Leaders acknowledge that they do not know the number of agile teams needed, how fast they should be added, and how to deal with the bureaucratic barriers without creating chaos within the business. Consequently, they usually release an initial group of agile teams, collect information about the value generated by the teams and the barriers they face and then determine if, when, and how to move forward. This allows them to measure the value of the agility expanding against its costs. Consequently, if the advantages exceed the costs, leaders keep increasing agility, releasing another group of teams, unblocking barriers in less agile areas of the enterprise, and repeating the cycle. Meanwhile, if the advantages are lower than the costs, they can pause, monitor the business environment, and discuss ways to improve the value of the already placed agile teams and reduce the cost of change.

Conversely: large-scale projects are more challenging and need complete leadership dedication, a responsive culture, and sufficient capable and experienced agile practitioners to create many teams without diminishing other capacities-also; a high tolerance for risks with mitigation plans to assess unforeseen failures. Thus, organisations that are low on those assets will have more success rolling out agile in a series of steps [7].

Building agility across the business: According to [22], ignoring other parts of the organisation when scaling agile is a frequent error. Thus, implementing large-scale agile, which involves the entire organisation, implies an impact and change in all parts of the organisation's work style, and everyone must be synchronised. Furthermore, although a significant step towards increasing the agility of the organisation is to increase the number of agile teams, how those teams communicate with the rest of the company is just as critical [5,7]. In summary, other areas of the organisation cannot be side-lined when launching dozens or hundreds of agile teams. Organisational conflict arises if the new agile units are continually hindered by bureaucratic processes or missing collaboration between operations and innovation teams, causing breakdowns and poor outcomes [7]. Therefore, changes are needed to ensure that the processes that are incompatible with agile support those that do [5,7]. To ensure that bureaucratic functions do not interfere with the work of the agile teams or be unsuccessful with the implementation and commercialisation of the innovations created by those teams, most organisations pressure for more change in a minimum of the following four areas: values and principles, operating architectures, talent acquisition and motivation, and annual planning, and budgeting cycles [7]. Thus, a management team looking to scale up agile must inculcate agile principles and values within the business. This includes parts of the organisation that do not form agile teams [7]. If employees understand agile principles, they will also understand why the transition is being carried out and be motivated to support it. Based on [21], instilling agile values and practices can be achieved by arranging social events and cherishing agile communities. Change must be extensively communicated to the entire organisation. Thus, many communication channels should make the new work approach evident. Equally important, management commitment should also be visible, as it is crucial throughout any transformation [5,21]. Stated that the system or operational architecture drives productivity for many years, for good or bad. Thus, it must align with the business objectives. For instance, Amazon can deploy software updates numerous times within a day. This is because the operational architecture was created to help developers increase the frequency and rapidness of releases without threatening the organisation's complex systems [7]. Accordingly, regardless of the speed at which the developer can code programs, large companies can only deploy software a couple of times per day or week because of how their architecture is structured [7]. The misalignment of human resources (H.R.) with agile principles was one of the challenges reported by Dikert, [18], which need to be addressed to acquire the full benefits of agile. Notably, a business cannot recruit solely on expertise; agile requires expertise and an eagerness to work on a collaborative team. Similarly, rather than judging individuals according to whether they achieve individual goals, it is now required to evaluate the success of agile teams, and assessments are to be made of each other within the team. Incentive schemes are usually associated

with individual performance, which generally works against team-centred thinking and the agile method [5,7] assert the need to reward groups instead of individual achievements, and therefore, organisations need to restructure their compensation schemes.

Finally, financing mechanisms are distinct in organisations with numerous agile teams. Funders need to accept that the original concept will change. They must anticipate that teams will remove certain features and launch others without waiting for the next annual cycle. Consequently, finance mechanisms must evolve to mimic those of venture capitalists, who usually see funding processes as opportunities to buy options for additional discovery [7].

Research methodology

This study seeks to identify critical success factors (CSFs) for implementing large-scale agile software development (ASD). The research contribution involves conducting a desktop study and analysing empirical data by comparing findings from a literature review with insights gained from two case studies on organisations' implementation of ASD practices at scale and their strategies for overcoming associated challenges. By juxtaposing theoretical insights with practical experiences, the study aims to deepen understanding of the CSFs involved in large-scale ASD implementation and the approaches used to address them.

The research began with a comprehensive literature review on large-scale ASD to elucidate critical concepts, challenges, and implementation practices. Success factors (S.F.s) were initially identified and refined into critical success factors (CSFs). Subsequently, the study adopted a multiple case study approach as its research strategy, drawing on [23] definition of a case study as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not evident." Consequently, the research focuses on conducting in-depth investigations into the contemporary phenomenon of large-scale agile practices within real-world or organisational contexts. Two factors notably guided the selection of case studies

- They included the challenges that the organisation faced when implementing ASD practices on a large scale and how those challenges were mitigated throughout the transformation.
- Included is a description of how the organisation implemented ASD on a large scale.

Case Studies: Comparative Analysis

Case Study 1: Ericsson

This case study was conducted by.

Background: Ericsson is a software development company that developed an XaaS platform, and a related range of products was architecturally composed of parts called components. Third parties created a few components, and Ericsson created a few until 2011 when Ericsson acquired the product. After Ericsson acquired the product, the newly built company had to concentrate on the knowledge transfer from external consultants to the employees of Ericsson and the newly recruited consultants. Moreover, Ericsson traditionally uses a plan-driven software method. Nevertheless, the organisation began a global implementation of agile software development in recent years.

Transformation approach: Ericsson adopted the experimental approach for transformation because they had discovered from prior experience that it is impossible to plan extensively and implement development with a "waterfall mindset". Consequently, trainers and managers selected the experimental approach to concentrate on one fundamental change or improvement goal. In this way, the main phases of transition were not designed prior but were determined on a need-based basis, one at a time. The transformation is cacategorisednto three main stages:

1. Introducing agile.
2. Finding common ground through value workshops.
3. Towards continuous integration and deployment.
4. Case Study 2: Best Buy

This case study was conducted by [24].

Background: In 2016, Best Buy (located in Canada) confronted the challenge of staying ahead of its competitors while remaining relevant and thriving. Thus, Dave Evans (the senior director of ecommerce) was tasked to guide the organisation's software development effort. The company dramatically changed its strategy to deliver value in ecommerce. Before this, Evans claimed that he did not feel work was satisfactory as there was excessive waste. Instead of using the traditional approach, Best Buy started using ASD, focusing on projects and products.

Transformation approach: Best Buy's transformation process can be classified by the following three main significant actions:

1. Hiring an external agile coach
2. Expanding agility/ changing the culture
3. Empowering employees.
5. Comparative Analysis of the Transformational Process

A comparison of the transformational processes in each case study will be explored about the three main steps for scaling agile outlined by [7], facilitating a structured analysis alongside existing literature. As mentioned, these three steps include rolling out agile incrementally, leading agile by example, and fostering agility throughout the organisation. It is noteworthy that neither of the organisations in the case studies had a predefined master plan at the outset of the agile transformation process. Ericsson opted for an experimental approach, deliberately addressing one primary change or improvement objective. This meant the critical transformation steps were not predetermined but determined case-by-case as needs arose.

In contrast, Best Buy did not explicitly describe its approach as experimental. However, the case study indicates that they made decisions in response to the challenges encountered during the transformation. Consequently, agile implementation was carried out incrementally in both case studies, aligning with the literature's recommendations.

The literature suggests that an initial cohort of agile teams is typically established, focusing on collecting data regarding the value generated by these teams and the challenges they face [7]. Additionally, both case studies organisations' leadership in their agile transformation by embodying agile principles, a factor the literature emphasises as crucial for successful agile transformations. Moreover, the literature indicates that a purely top-down approach to leadership is not considered effective in driving agile transformations. Imposing change through managerial directives without recognising its value and necessity does not facilitate widespread adoption. Both case studies organisations had a different approach to their transformation; instead,

they granted teams considerable freedom and autonomy at the outset of the process. For instance, during the initial transformation phase at Ericsson, teams were free to determine their working methods, as there was no standardised framework guiding their daily operations. Similarly, at Best Buy, the focus was on making the organisation more transparent while empowering teams to devise strategies for achieving it. In essence, decision-making authority was delegated to the teams to promote the adoption of new work practices and acknowledge their learning experiences.

Moreover, at the initiation of the transformation, teams within each case study or organisation self-organise, with team members selecting themselves, thus are empowering them to address challenges independently. However, during the transformation process at both Ericsson and Best Buy, there came a point where teams were granted excessive autonomy, leading to difficulties. Therefore, striking a balance between the freedom and autonomy afforded to teams and the boundaries established by management is imperative.

As suggested by the literature, the final crucial step in implementing large-scale agile software development is to cultivate agility throughout the entire organisation. This entails ensuring that changes are not confined to agile teams alone but extend to all areas of the organisation that are not directly involved in agile practices. Consequently, adjustments must be made to align non-agile functions with those employing agile methodologies. For instance, during the second phase of the transformation at Ericsson, efforts were made to foster alignment through value workshops, as different company sites held varying perspectives on work methodologies. These workshops aimed to synchronise the company's segments to facilitate the success of the agile transformation and foster a cohesive product and organisation.

Similarly, at Best Buy, introducing new agile teams initially created tension with other non-agile departments within the organisation. Consequently, leadership had to enhance communication across the entire organisation to mitigate this friction. While Best Buy did not explicitly mention any specific workshops or activities in their transformation process, unlike Ericsson, management support was evident in both cases, aligning with literature highlighting it as a key driver in motivating employees to embrace new working methodologies.

Regarding the operational framework, both case studies are shared organisations. Best Buy maintained a unified priority list for all projects across departments, whereas Ericsson's backlog documented each new feature and enhancement. However, initial challenges arose for both organisations and adjustments had to be made to suit the specific circumstances. Initially, Ericsson aimed to form fully cross-component and cross-functional teams, but this was deemed impractical due to product complexity and competency disparities. Consequently, teams shifted focus to specific business flows, requiring expertise in select product components rather than the entire project. Conversely, the human resource cost of team members navigating multiple product domains at Best Buy was deemed excessive. While Best Buy did not explicitly create business flows, they formed predominantly static teams within product families, allowing members to deepen their expertise in specific product domains and enhance delivery efficiency.

It is important to note that neither Best Buy nor Ericsson extensively detailed changes in talent acquisition and motivation in their case studies. However, literature suggests that granting teams autonomy and empowering them to make decisions can motivate them, a practice both organisations have adopted in their work methods. Similarly, the case studies briefly mentioned annual planning and budgeting cycles, although many organisations are encouraged to reform these areas when implementing large-scale agile practices.

Neither organisation had completed the agile transformation process at the time of publication. Despite challenges, the transformational journey was deemed successful, with both companies expressing overall satisfaction with the outcomes. A comparison of the success factors from both organisations will be provided.

Comparison of the S.F.s

Table 2 compares the CFSs from both case studies against the literature. Please refer to table one to review the subcategories of each challenge.

Difficulty in implementing agile: Both case study organisations need help achieving a good balance of control and autonomy. On the one hand, Ericsson did not have a common agile framework, which provided too much freedom. On the other hand, for Best Buy, the management team was excessively eager to provide teams with autonomy when, in some cases, the teams were not prepared to judge the company's needs appropriately. Ericsson set up a Coaching Community of Practice (CoP) to mitigate this challenge and provide similar coaching throughout teams and locations. Alternatively, Best Buy used a natural way of communicating with its team members to provide critical feedback without authority to prevent disempowerment.

Change resistance: This is a frequent issue faced when implementing agile software development on a large scale [5,19]. This was visible in the case of Ericsson, particularly with the leadership team, because they preferred to concentrate on deliveries instead of transforming the business. This caused Ericsson to reorganise the leadership team with individuals with an agile mindset. Controversially, in the Best Buy case study, no challenge of change resistance was mentioned by any interviewees. Therefore, it is assumed that this was not a hurdle for this organisation during the transformation process.

Lack of investment: From the subcategories listed in Table 1, lack of coaches, lack of training, and trying to maintain old commitments were identified in the case of Ericsson. First, Ericsson needed to mitigate the lack of coaches fully. Although Ericsson slightly increased the number of coaches, it did not significantly impact improvement since teams grew simultaneously. However, this issue was reported as not impacting the transformation significantly. Second, the lack of training was mitigated by replacing training with coaching.

Table 2: Comparison of CSF against the literature.

Success factors identified from the literature	Ericsson	Best Buy
Difficult to implement Agile	✓	✓
Change resistance	✓	
Lack of investment	✓	✓
Coordination challenges in a multi-team environment		
Inconsistency of processes and practices across teams	✓	✓
Requirements engineering challenges	✓	
Quality assurance challenges	✓	
Integrating non-development functions		✓
Inadequate management support and sponsorship		
Lack of skills/ experience with Agile methods	✓	
Minimal collaboration and knowledge sharing	✓	
Lack of business/ customer/ owner availability	✓	
Fragmented tooling and project-related data/ measurements		
Regulatory compliance/ government issues		

On the contrary: Best Buy faced only one of the subcategories at one point of the transformation, which was too high of a workload. To further explain, there were many software requests, leading to a low customer-centric experience. Nevertheless, Best Buy demonstrated a substantial investment as the chief technology officer hired coaching, training, and consulting businesses to evaluate and pilot the change to help catalyse a transformational agile process.

Inconsistency of processes and practices across teams: Ericsson did not use a common framework, and this caused different practices across the teams. As mentioned previously, the creation of the CoP mitigated this problem. Similarly, for Best Buy, maintaining some standardisation of practice across diverse teams was identified as a challenge, and the issue was approached similarly to that of Ericsson. They focused on how communities of practice could have a regular cadence for the possibility to engage and discuss their practices and started to identify the approach leads in every community.

Requirements engineering challenges: In the Ericsson case, since the business's backlog was initially not common but had various independent ones, they experienced difficulties breaking down big features into appropriate-sized epics and user stories. However, they rolled a common backlog for the whole business to mitigate this challenge. Also, a significant technical debt created problems because the tasks had to be prioritised against creating new features. However, the common backlog served as a solution for this as well. Moreover, no challenges were identified in this category in the Best Buy case.

Quality assurance challenges: Initially, for Ericsson, there were difficulties due to the absence of test automation and continuous improvement (CI) practices. This is because a large amount of testing was required manually, decreasing the time available for the development in every iteration. Accordingly, the actions that were taken to mitigate this included a critical investment in developing a functioning CI system, as well as CI roadshows to increase acknowledgement of CI and help instil the correct mentality in the teams. On the contrary, the Best Buy case did not mention challenges related to this category.

Integrating non-development functions: For Ericsson, no significant challenges were reported in this category. On the contrary, when Best Buy initiated the transformation in the development department, the rest of the enterprise could not keep up with the teams due to the rapid delivery pace of the teams. Therefore, it created pressure against the remaining part of the business. In other words, there was scepticism towards agile practices from the rest of the organisation, and Best Buy had to communicate intensively with those parts to alleviate the issue.

Lack of skills/experience with agile methods: For Ericsson, not all individuals had the same level of agile knowledge. Nevertheless, although the organisation had various individuals with agile experience, the overall agile knowledge was not yet appropriate. However, it was mentioned that the collaboration between the coaches throughout the sites would increment agile knowledge in time. Contrarily, the Best Buy case did not mention a lack of skills or experience with agile approaches.

Minimal collaboration and knowledge sharing: The case study for Ericsson did not demonstrate significant challenges regarding minimal collaboration. This may have been due to the implementation of component-based communities of practices. However, knowledge sharing was a challenge at some point due to the cross-site team distribution. At the time of the case, this problem was mitigated by pair working to broaden team members' knowledge. This was mitigated by allowing the teams to specialise in particular business components. Furthermore, for Best Buy, no significant challenges were mentioned within this category.

CSFs: Recommendations for industry

After conducting a comprehensive review, the subsequent critical success factors (CSFs) are extrapolated from the literature review and the comparative case study analysis. CSFs represent pivotal elements necessary for a project to achieve its mission or objective [20]. It is advised that these three CSFs, which underpin the success factors delineated in Table II, be adhered to when implementing large-scale agile software development practices. Nevertheless, it is imperative to ememphasisehat effective leadership and management support is indispensable throughout the transformation process. Additionally, organisations need to adapt these recommendations based on specific or organisational actors unique to their business. The three CSFs are:

Lead the change by being agile at the top: Leadership teams should implement the change by being and practising agile. Therefore, organisations should understand what agile is and what it involves before beginning the process. Moreover, an essential agile practice is creating self-management teams, providing autonomy, and increasing decision-making responsibilities. The top-down approach does not seem appropriate for implementing ASD, at least not strictly alone. Besides, forcing team members to do something just because the leadership or management team commanded does not allow individuals to understand the value and benefits of implementing Agile practices. It is likely to discourage the employees from supporting the change.

In addition, when employees feel empowered to accept this new way of working, there is less chance of resistance to the change. Nevertheless, there should be a balance between the autonomy given to the teams and the limits set by the management team. Giving too much autonomy can cause problems maintaining standardised consistency with the processes and practices across the teams. Specifically, creating a community of practice can provide this consistency across teams as it allows individuals to discuss and engage their practices and approaches, and similar coaching can be provided across teams and locations.

Roll out Agile in a sequence of steps: It is recommended to roll out Agile in a sequence of steps rather than a big-bang implementation. This is because organisations will likely encounter unforeseen issues along with the transformation, requiring changes to the initial set of objectives. Consequently, a pilot team or an initial set of teams can be released to evaluate the new approach within the organisation. In addition, information should be gathered after the initial set of teams is released, such as the value the teams achieved and the barriers they faced. Depending on the teams' results, the next step is to determine how the transformation should proceed. This facilitates the transformation because, usually, not everything can be planned.

Building agility: Across the organisation Although it is an essential part of implementing agile software development, the number of teams released and how they interact with the rest of the organisation are equally important. Therefore, the rest of the organisation should not be ignored when launching teams, even if they do not operate as agile. Hence, it is crucial to make the changes necessary to ensure that the areas of the organisation operate differently than agile support the ones that do. Otherwise, it can cause conflicts with the rest of the organisation. Specific recommendations for this CSF are the following:

- Changes in values and principles across the organisation

need to be implanted by leadership and management teams. Agile principles and values can be implanted throughout the organisation through joint value workshops and active communication regarding change within the organisation.

- Changes in the business's operating architecture: The architecture is an important aspect that affects productivity positively or negatively. Therefore, it is essential to make changes in this business area to align it with the business objective. Consequently, creating a common backlog with a single priority list is recommended to increase the productivity and product deliveries of the teams. In the case that not every team can adopt any task from the list either because of knowledge gaps or costs, the following suggestions can be implemented:

  o The teams' focus should be changed to specialising in specific business flows. This would not require them to know all the product components but only a few. Consequently, a business flow-based backlog list can be created.

  o Create statics teams within product families.

Conclusion

This paper introduces the critical success factors (CSFs) necessary for implementing large-scale Agile Software Development (ASD), offering valuable insights for organisations seeking to expand their agile methodologies. By juxtaposing findings from a literature review with the analysis of two case studies, this research enhances comprehension of this phenomenon, aiding organisations in adopting ASD practices. Despite criticism surrounding the effectiveness of large-scale Agile implementation, this study demonstrates that agile practices can indeed be effectively deployed on a large scale with proper methods and strategies.

However, the adoption of large-scale agile practices presents unique challenges. This study highlights several common obstacles identified by comparing the literature review and cross-case analysis. These include difficulties in agile implementation, insufficient investment, and inconsistent processes and practices across teams. The implementation process outlined in the literature for large-scale ASD practices entails three key steps: leading agile by embodying its principles, rolling out agile incrementally, and fostering agility throughout the organisation, [7]. Through the cross-case study analysis, both Best Buy and Ericsson adhered to these critical success factors (CSFs) while adapting them to address specific challenges or barriers encountered by each organisation

Thus, based on the literature and empirical data analysis, it can be concluded that these three critical success factors (CSFs) are applicable when implementing agile software development practices on a large scale. This study demonstrates that organisations exhibit unique characteristics within their business contexts. However, commonalities are observed in the implementation process, as well as challenges faced by organisations. This was evident in the comparison of the cases, where each company had its specific factors but also shared everyday activities and challenges. Therefore, other organisations aiming to adopt Agile software development practices at a large scale can recognise similar issues and problems discussed in this study and leverage the findings to facilitate their transformation process. As a result, this study will assist them in navigating this transformation journey effectively.

As a final note, the findings from this study are not expected to represent all the large-scale businesses adopting Agile practices. As other researchers carry out more case studies, the contribution to the agile community is expected to expand progressively.

References

1. Dingsøyr T (2012) a decade of agile methodologies: Towards explaining agile software development. J Syst Softw 85: 1213-1221.
2. Boehm B, Turner R (2005) Management challenges to implementing agile processes in traditional development or organisations. IEEE Software 22: 30-39.
3. Williams L, Cockburn A (2003) Agile software development: It's about feedback and change. Computer 36: 39-43.
4. Dingsøyr T (2018) Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. Empir Softw Eng, an Int J 23: 490-520.
5. Dikert K, Paasivaara M, Lassenius C (2016) Challenges and success factors for large-scale agile transformations: A systematic literature review. J Syst Softw 119: 87-108.
6. Bass JM (2019) Agile on a large scale.
7. Rigby DK, Sutherland J, Noble A (2018) Agile at scale. Harv Bus Rev 96: 88-96.
8. Dooley J (2011) Software Development and Professional Practice. 1st edn Berkeley, CA: Apress.
9. Crookshanks E (2014) Practical Software Development Techniques: Tools and Techniques for Building Enterprise Software. 1st edn Berkeley, CA: Apress L.P.
10. Highsmith JA (2002) Agile Software Development Ecosystems. Boston: Addison-Wesley.
11. Denning S (2016) 'HBR's Embrace of Agile. Social Sciences 9: 69
12. Misra SC, Kumar V, Kumar U (2010) Identifying Some Critical Changes Required in Adopting Agile Practices in Traditional Software Development Projects. Int J Qual Reliab Manag 27: 451-474.
13. Dingsøyr T (2014) Agile Methods Large-Scale Development, Refactoring, Testing, and Estimation: XP 2014 International Workshops, Rome, Italy, May 26-30, 2014, Revised Selected Papers. Switzerland: Springer.
14. Ambler SW, Lines M (2016) scaling agile software development tactically: Disciplined agile delivery at scale. Disciplined Agile Consortium, Tech Rep.
15. Flyvbjerg B, Budzier A (2013) why your I.T project might be riskier than you think. Harv Bus Rev 89: 23-25.
16. Flyvbjerg B (2014) what you should know about megaprojects and why: An overview. J Proj Manag, SAGE Publications Sage CA: Los Angeles, 45: 6-19.
17. Blichfeldt BS, Eskerod P (2008) Project portfolio management-There is more to it than what management enacts. Int J Proj Manag 26: 357-365.
18. Paasivaara M (2018) Large-scale agile transformation at Ericsson: a case study, Empirical software engineering. An Int J 23: 2550-2596.
19. Digital AI (2020) 14th Annual State of Agile Report. Int J Appl Basic Med Res 14.
20. Leidecker JK, Bruno AV (1984) Identifying and using critical success factors. LRP 17 23-32.
21. Gruver G, Young M, Fulghum P (2012) A Practical Approach to Large-Scale Agile Development: How H.P Transformed LaserJet Future Smart Firmware. 1st edn. Addison-Wesley Professional.
22. Holler R, Culling I (2010) From Agile Pilot Project to Enterprise-wide Deployment: Five sure-ways to fail when you scale. Version One.
23. Yin RK (2014) Case study research : design and methods. 5th ed. California: Can J Program Eval 30.
24. Sharrock D (2021) Best Buy Canada Makes Agile Transformation. Scrum Alliance.