



HAL
open science

Agile versus traditional project management approach: is Agile really the best project management methodology out there in the IT sector, or do we need to delve deeper?

Monali Bhowmik

► To cite this version:

Monali Bhowmik. Agile versus traditional project management approach: is Agile really the best project management methodology out there in the IT sector, or do we need to delve deeper?. Business administration. 2022. dumas-03922852

HAL Id: dumas-03922852

<https://dumas.ccsd.cnrs.fr/dumas-03922852>

Submitted on 1 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

Master Thesis

Agile versus Traditional Project Management approach.

Is Agile really the best project management methodology out there in the IT sector, or do we need to delve deeper?

Presented by : BHOWMIK Monali

Host organization : Université Grenoble Alpes, 621 Av. Centrale, 38400 Saint-Martind'Hères

Date: from 01/04/22 to 11/06/22

**Internship supervisor : PEREA Céline
University advisor : DOMINGUEZ Carine**

**Master 2 – Advanced Research in Management of Information Systems
2021 - 2022**

Master Thesis

Agile versus Traditional Project Management approach.

Is Agile really the best project management methodology out there in the IT sector, or do we need to delve deeper?



Presented by : BHOWMIK Monali

Host organization : Université Grenoble Alpes, 621 Av. Centrale, 38400 Saint-Martin-d'Hères

Date: from 01/04/22 to 11/06/22

**Internship supervisor : PEREA Céline
University advisor : DOMINGUEZ Carine**

**Master 2 – Advanced Research in Management of Information Systems
2021 - 2022**

Disclaimer:

Grenoble IAE, University Grenoble Alpes, does not validate the opinions expressed in theses of masters in alternance candidates; these opinions are considered those of their author.

In accordance with organizations' information confidentiality regulations, possible distribution is under the sole responsibility of the author and cannot be done without their permission

Abstract

For the successful implementation of product development initiatives, there are a variety of process models. Like the classic project management methods, like the waterfall, spiral or the V model, agile project management methods, such as Scrum and XP are becoming more popular. The challenging decision about which software development life cycle (SDLC) to utilize must be made by organizations producing software solutions.

In this paper, I've tried to lay out the internal and external contingency elements that will help managers determine the best project management technique for an IT project in the future. I conducted a literature review to investigate the distinctions between agile and traditional or plan-driven techniques, as well as a few hybrid methods that have been explored by academics and can be a mix of agile and traditional methodology, as well as within agile methodologies itself. Because project managers are migrating to agile for a variety of reasons these days, we conducted a literature review to see if agile is suited for all types of projects or if managers should choose the proper methodology before starting an IT project.

The paper is structured in the following way: Introduction, Literature review, Research problem, Research Methodology, Conclusions and finally the References

KEYWORDS: Agile Project Management, Traditional Project Management, Waterfall, Hybrid methods, Contingency Factors, Project Managers

Acknowledgment

I would like to thank all my professors for their invaluable patience and feedback. I could not have embarked on this journey without the help of my supervisor, Madame Celine Perea, who generously shared her knowledge and expertise.

I also want to thank my classmates for their editing assistance, late-night feedback sessions, and moral support. Thank you also to the university librarians, research assistants, and study participants who influenced and inspired me.

Finally, I would be remiss if I did not mention my parents. Their faith in me has kept my spirits and motivation high throughout this process.

Table of Contents

- FOREWORD..... 7**
- INTRODUCTION..... 8**
- LITERATURE REVIEW 10**
 - 1. Beyond the two opposite views of IS Project Management 10**
 - 1.1. Agile versus Traditional Project Management 10**
 - 1.2 Many Agile Methods and Hybrid Methods 14**
 - 2. Choosing the right Project Management Method 18**
 - 2.1. How the Project managers are possibly making the decisions 18**
 - 2.2 How the decision can be taken with respect to the Contingency theory 21**
- Research Problem 22**
- Research Methodology 24**
- Conclusion..... 30**
- Bibliography 31**

Foreword

This thesis was written as part of my master's degree program in Advanced Research in Management of Information Systems (ARAMIS) at Grenoble IAE in France. This thesis's topic is related to agile and. This is a very interesting research topic because it deals with agile and traditional project management methods, as well as project managers' perspectives on selecting a method in the IT industry.

I'd like to thank a few people here after thanking Almighty and my family for their unending support. Madame Carine Dominguez, my course co-ordinator, and Madame Celine Perea, my supervisor, for their guidance, reviews, and recommendations. They were always available to assist me with my research whenever I needed it. Working with these people was a huge learning curve for me because they allowed me to improve my research skills.

June 2022,
Monali Bhowmik

Introduction

While object-oriented (OO) procedures are a viable way to iteratively build information systems, a wave of new methods labeled agile development methodologies or lightweight methodologies claim to go much further in overcoming the limits of traditional plan-driven procedures [Nerur et al.2005]. Software development methodologies include crystal clear method, extreme programming (XP), scrum, feature-driven development, dynamic systems development, and adaptive software development [Highsmith, 2003]. Although conventional methods such as life cycle-based structured and object-oriented techniques continue to win the systems development field, according to Nerur, agile methodologies are getting popular, as seen by many opinion pieces and studies. With the emergence of these methodologies, the software development community has been divided into 2 camps: traditionalists and agilists, both claiming being the supreme methodology.

"Organizations must carefully grow toward the ideal balance of agile and plan-driven methodologies that fits their environment," according to [Boehm, 2002]. Iterative software development, regular client consultation, short and frequent releases, and thoroughly tested code are all examples of agile software development [Cao et al., 2004]. To become more competitive, enhance procedures, and cut costs, several businesses adopt agile principles. Other businesses are dubious about the advantages of agile development. According to experts, large firms face additional challenges in integrating agile principles with current standards and business procedures [Barlow et al..2011]. Agile software development was created to address the limitations of plan-based software development approaches like the commonly used waterfall approach [Royce, 1970]. The lack of response to change is the major flaw of plan-based systems. Because plan-based development is sequential, project managers that plan projects early in the process, are led by little flexibility in subsequent phases of development. However, between the start and finish of a project, the needs can change dramatically. [Cockburn and Highsmith, 2001] Agile development helps firms to react to these changing situations, allowing for more flexible development. Indeed, most businesses cannot ignore the agile tide. However, because the two software development processes are founded on opposing concepts, adopting agile approaches will certainly provide several challenges for enterprises using traditional systems development methodologies. Software development process modifications, according to past research, are sophisticated organizational development phenomena that cannot be realized just by substituting the current tools and technology with new models [Sircar et al..2001]. These changes can have an impact on the organization's structure, management methods, and culture among other things. Understanding the repercussions of a change phenomenon across an organization is thus a crucial step in planning and implementing the changes. While agile being the most popular methodology used in projects, some criteria's needs to be addressed before quickly adapting to agile. Although the agile Software Development Life Cycle (SDLC) is better suited to small-to-medium project development, the traditional SDLC is still preferred for big projects. As a result, the project team must select the appropriate SDLC for the work. The size of the project team, software size, location, and complexity, project kind, business strategy, and technical capabilities, as well as other instances where it may be advantageous, are some of the characteristics that the project team may utilize to determine the optimum SDLC [Leau et al..2012]. He also remarked that the team must conduct research into the matter. Examine the differences, downsides, and benefits of each SDLC before deciding. Moreover, to assess the scenario, the team must examine the business context, industry needs, and corporate strategy considering the proposed SDLC's selection criteria.

In this paper, I will attempt to determine from facts and findings whether agile methodology is indeed easy to adopt by all types of organizations while ignoring other traditional approaches that may also be beneficial to specific types of firms and IT projects. There have been multiple researches about the comparison between agile and the plan driven approaches in the IT sector but there is very less researches done about how the managers should be choosing between agile and traditional methods depending on the demand of the project that is to be completed and based on the characteristics of the team members. Our core purpose is to review the literature to establish whether agile is the best

project management strategy and firms should adopt it immediately, whether a combined approach may be used, or whether there are only two perspectives on project management. We'll also try to answer topics like how project managers choose the best project approach and what criteria they use and propose ways with respect to the contingency theories through which project managers can choose if they should use a plan driven, agile or a mix of the methodologies for their projects.

Choosing a project management technique that is suited for the project

Characteristic	Traditional approach	Agile approach
Requirements	clear initial requirements; low change rate	creative, innovative; requirements unclear
Users	not involved	close and frequent collaboration
Documentation	formal documentation required	tacit knowledge
Project size	bigger projects	smaller projects
Organizational support	use existing processes; bigger organizations	prepared to embrace agile approach
Team members	not accentuated; fluctuation expected; distributed team	collocated team; smaller team
System criticality	system failure consequences serious	less critical systems
Project plan	Linear	complex; iterative

Table 1: Differences between agile and Traditional methods characteristics. Adapted from Spundak, M. (2014)

Assume this IT project: a small IT firm developing specialized software for a huge corporation. Using the criteria from Table 1, Even though the customer provided full documentation for the project and the end users are not involved, such a project could be classified as having certain unmet needs. Because the customer's organization is so formal, official documentation is required all through the project, especially towards the end, and current methods should be used. A smaller team is planned with a mostly linear project plan because the project is smaller (in terms of both duration and workload), (2 iterations since the project requirements are unclear). Lastly, within the company, the developed application is not the most crucial. So, what approach should be adopted for this project, and how should the project methodology be structured?

Project management methodology offers benefits such as better supervision of the project goals and scope, speed to market, more efficient processes such as decision making and quality management, improved customer service, and better information management by allowing exchange of information between projects and letting more time for value added operations [Chin and Spowage, 2010]. Methodology, on the other hand, is not sufficient nor even required for project success. If the firm and project team do not fully appreciate the project scope and context, there have been no techniques or methods in any approach that can guarantee the success of the project [Introna and Whitley, 1997]. Inadequate approach can also hinder project success or make project management more difficult [Cheema and Shahid, 2005].

Then, how can we tell if a methodology will help a project succeed? Project success is a multi-dimensional notion that is influenced by criteria, stakeholder perceptions, context, and project stage, according to [Iriarte and Bayona,2020]. They mentioned in their paper that the most crucial aspect for IT projects is soft skills, also known as people skills. Due to the characteristics of information technology projects, human resources management is a critical component of project success . He conducts a thorough review of the literature in order to compile and summarize project success factors in IT projects. Given the specific features of IT projects, all scholars agreed that examining key success factors is critical. The results indicate that there is no one-size-fits-all approach to project success; his analysis combines IT success factors such as time, money, project management, system quality, user satisfaction, and economic benefit. Furthermore, because there is a large and overlapping list of elements, the study suggests a structure that synthesizes the most referred essential components that share soft features such as involvement, communication, commitment, and support.

One of the preconditions for successful methodology adoption [Charvat, 2003] is that it is consistent with other company processes, which is why many companies have established their own project management methodology. Furthermore, methodology should include sufficient information tied to organizational procedures. It's also crucial to understand the methodology's limitations, especially in the context of its application [Cockburn and Highsmith, 2001].

Our research question here is, "Should firms always adopt agile methods for project delivery and neglect other approaches that may be more beneficial based on the clients requirements and team availability?"

Literature Review

2 Beyond the two opposite views of IS Project Management

Software development is varied, and businesses must quickly adapt to new technology and markets. There is no "Silver Bullet" in software development, according to [Frederick, 1987]. As a result, application developers are still on the search for suitable development methodologies, but they're up against a storm of contextual factors that could make defining ideal development processes challenging [Kalus and Kuhrmann, 2013]. Different software and system development methodologies have been developed to meet these factors, as well as the growing number of fields in which software has become a critical component. These techniques employ a variety of philosophies, including plan-based, lean, incremental, or iterative, and cover the full lifespan, from small task-specific practices (such as daily stand-up) to big and thorough process frameworks (e.g., sequential models). Software development has also grown more vital in the development of safety-critical and trustworthy platforms, such as automotive software, aerospace systems, and medical devices. Software and its evolution benefit from the addition of standards, rules, and regulations in these domains [Khurmann et al., 2017].

2.1. Agile versus Traditional Project Management

Let's start with understanding what does the two methods mean in the perspective of IT project management. The term "project management approach" refers to a collection of rules and criteria that dictate how a particular project is handled. A project management framework is a set of rules, methodologies, processes and templates that must be followed throughout the project lifetime [Introna and Whitley, 1997].

- Traditional project management (TPM)

According to the Project Management Institute (PMI, 2013), traditional project management (TPM) is defined as an application of education, skills, instruments, and activities to meet the project activities in order to meet project requirements. TPM also requires the implementation of five phases: initiating, planning, executing, monitoring and controlling, and closing, all with the support and supervision of the project manager and project team (PMI, 2013). The traditional approach assumes that projects are relatively simple, consistent, and straight, with well-defined boundaries, allowing for detailed planning and sticking to that plan without significant changes [Boehm, 2002; Boehm and Turner, 2003]. The classic project management technique's end idea is to increase efficiency and productivity in executing the first project plan, or, to take another example, to meet the deadlines, within budget, and within range [Shenhar et al., 2002]. Although the conventional approach to project management highlights durability as one of its advantages, claiming that the very same techniques and methods can be fairly applied to all projects, it is increasingly

acknowledged as one of its major drawbacks. [Spundak, 2014]. According to him, the traditional strategy is unsuccessful for most today's projects caused by structural complexities, uncertainty in objective formulation, and project time constraints.

- Agile Project management (APM)

The ability of an organization to adjust to changing situations without being compelled to change is known as organizational agility. The ability to act actively in a dynamic, unpredictable, and constantly changing environment is defined as agility. Agile software development methodologies have had a significant influence on APM concepts and methods. [Salameh, 2014]

The APM methodology is focused on short delivery iterations and ongoing learning [Sauer and Reich, 2009]. The project team begins by doing a streamlined planning, requirements definition, and solution design to kick off the project. Following that, in future waves of iterations, the team is involved in more detailed planning, requirements analysis, design, execution, tests, and delivery to customers and clients. The APM technique allows for immediate project change because requirements are checked and evaluated at each iteration. APM also employs a feature-driven management method, which focuses on defining a project's scope and objectives by ranking a list of project features and needs according to their value, including such increased revenue or market share. As a result, customer involvement in the design and analysis of the project's requirements is essential. Customer engagement ensures that the agile project team does not waste time on items or needs that are ineffective or low value [Salameh, 2014]. APM places a strong emphasis on collective development and management in order to achieve results, as well as client feedback, continuous improvement, and additions [Hass, 2007]. In APM's highly iterative and incremental methods, project team members and stakeholders actively participate to comprehend the project domain, define what needs to be worked, and define prioritized functionality. A distinguishing element of agile development and management is that each iteration is self-contained, with duties extending from requirement analysis phase to design, implementation, and testing [Larman, 2004]. At the end of each iteration, the customer receives a release that includes all software components; the client then provides the necessary feedback and modifications in the system's demands and features, which are planned and implemented in future iterations or iterations. Agile software development and management is guided by the concept of value-driven delivery, which attempts to meet customers' demands by providing real and high-priority new features early and frequently. Moreover, agile management does not resist change because it benefits customers by using it to obtain a competitive advantage [Salameh, 2014].

Conceptual Difference

[McCormick, 2012] have wonderfully explained the difference between the agile and the traditional methodologies. As the name implies, the waterfall model is a continuous software development approach, he stated. The production cycle progresses sequentially from one phase to the next, just like water falls gradually through one altitude to another in a river. The steps of software development under this approach are requirement formulation, conceptualization, analysis, design, coding, testing & debugging, installation, then finally maintenance. In this sequentially structured approach, the development team carries on to the next level of development only when the previous phase is done. This technique necessitates that software development companies commit substantial effort to each step of development until all questions have been resolved and all objectives have been completed. This method of software development is predicated on the idea that devoting a large amount of time to the early design work will avoid problems from arising later. When the design stage is finished, the coding stage executes it exactly as planned, without no changes made later. The analysis,

design, and programming teams are commonly divided and assigned to different stages in the project. McCormick stressed the importance of documenting each stage of software development.

He asserted that agile development models emphasize "agility" and "adaptability" in development, as opposed to waterfall development models' "set-in-stone" approach. Instead of a single moment and rigid development timetable, agile models use many progressive development plans that seek to improve the output with each iteration. Every iteration comprises all steps of design, development, and testing. The design is not set in stone and is subject to last-minute changes due to iterative delivery. The group is multifunctional, well-connected, and self-organized. The concept design is never finalized, but it is permitted to change as new ideas are offered with each release. The speed with which a working software is delivered is given more weight than the documentation. Customers may be given demonstrations at the end of each iteration, and their feedback may impact the improvements in the next iteration. The iterative loop continues until the client receives a product that meets all his demands. Table 2 shows the differences between the two techniques as explained by [Nerur et al.,2005].

	Traditional	Agile
Fundamental Assumptions	Systems are fully specifiable, predictable, and can be built through meticulous and extensive planning.	High-quality, adaptive software can be developed by small teams using the principles of continuous design improvement and testing based on rapid feedback and change.
Control	Process centric	People centric
Management Style	Command-and-control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual – favors specialization	Self-organizing teams – encourage role interchangeability
Communication	Formal	Informal
Customer's Role	Important	Critical
Project Cycle	Guided by tasks or activities	Guided by product features
Development Model	Life-cycle model (Waterfall, Spiral, or some variation)	The evolutionary delivery model
Desired Organizational Form / Structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology

Table 2 Agile vs Traditional. Adapted from Nerur et al. (2005)

A recommended set of practical arrangements (e.g., roles, meeting procedures, documentation, and other actual organizational policies) that claim to have a beneficial influence on participant involvement, flexibility, and productivity is often designed as an Agile method statement. Agile techniques highlight the value of development teams' autonomy. During the continuing project, both within the company and in the specific project, the practical design of roles, meeting processes, and so on should be regularly updated to local demands. Any local change after that can become guesswork without a good understanding of how agile arrangements influence participants [Dyba and Dings 2008]. Customizations are frequently

built on pieces of gut feeling, and with a little poor luck, they can become inefficient or even destructive.

"When employing agile methodologies, the traditional position of a project manager as planner, organizer, and controller vanishes, to be replaced by the role of a facilitator who effectively oversees team members' participation without suffocating their creativity," according to [Highsmith, 2003]. Some critics have suggested that an agile program manager should be a teacher or guide [Coram and Bohner, 2005], a facilitator [Nerur et al., 2005], or a "protector" to guarantee the project team's effective collaboration [Boehm and Turner, 2005]. These interpretations imply that, in contrast to the traditional approach's process-centric style, agile project managers should work in a more people-centric manner [Nerur et al., 2005]. With the passage of time, more and more firms are adopting agile project management approaches, whether it is owing to customer demands or a desire to keep up with the competition. Agile approaches typically account for most of the application development industry. Most firms must reorient toward an agile model sooner or later, given the constantly changing (economic, social, political, cultural, and so on) conditions in their surroundings. [Stoica et al., 2016]. According to [Veiga, 2017], Customers' needs are nearly never static, and they change over time owing to new business needs, technical advancements, or simply shifts in the company's priorities. This is especially important in larger projects. Organizations have realized that employing the Waterfall method of development does not provide enough flexibility to satisfy changing needs based on stakeholder expectations. Even if there is no perfect solution for project management and performance, executives and managers are looking to agile project management as a vital tool to help them overcome this challenge. As the need for high quality in a dynamic and rapidly changing global market grew, professionals were forced to develop Agile Project Management (APM) methodologies [Fitsilis, 2008].

Traditional methodologies, on the other end, can be inadequate for complex projects, especially IT and software projects, due to the subjective and dynamic nature of requirements. APM has evolved into a highly iterative and incremental approach in which the project team and stakeholders work together to understand the domain, identify what needs to be produced, and prioritize functionality. The results are quick and market sensitive as a result of frequent customer interaction and early concept testing [Salameh, 2014]. These outcomes promote client satisfaction, which in turn increases customer loyalty, engagement, and commitment, as well as financial benefits such as higher sales, profits, and financial results. Agile development, in general, promotes flexible development that adapts to changing customer needs, improves software quality, and improves overall customer happiness. According to several research and polls, these are some of the key reasons for firms transitioning to Agile. When measuring success in terms of budget, features, and time, Agile-developed software projects typically outperform traditional software projects, especially for medium and large-scale projects. Although the differences between the smallest projects aren't as significant, they benefit from the use of Agile approaches as well. In general, literature reveals that not only Agile development has a greater ratio of project success (designed as product and project success), but also that organizations have recognized this and have adopted Agile as a strategy to attain better project success ratios. [Veiga, 2017]. Agile methodologies are gaining popularity as a result of their high adaptability to changing environments, project requirements, and project circumstances. According to [Fernandez and Fernandez, 2008], Agile practices, such as project management, evolved from the need to manage projects with high levels of complexity and uncertainty while being flexible and adaptable. When aims and outcomes are uncertain and there is a lot of unpredictability, different project management approaches are extremely relevant. If project managers are conversant with diverse project management methodologies, they would be better able to compare the peculiarities of the current project. The effort to embrace agile project management principles and understand how to be adaptable may be well worth the investment for many project managers. This flexibility could be highly useful when dealing with specific sorts of projects and project settings.

On the other hand, [Shaydulin and Sybrandt, 2017] have offered a comparative analysis of several of the most popular software development approaches in his paper. The findings demonstrate that each of the "agile" approaches is missing a crucial core criterion, such as technical debt control or cost estimation. These difficulties can be neglected in smaller projects since the costs are low and the potential for technical debt is low. These constraints make "agile" methodologies inapplicable for extremely big and complex projects. As a result, the author claims that "agile" approaches' fundamental qualities limit their application to a certain group of problems: tiny and manageable in small teams. The agile methodologies they looked at, on the other hand, do not adapt well.

2.2. Many Agile Methods and Hybrid Methods

It's important to recognize that there are six various types of agile approaches that are commonly utilized inside a company, which will aid project managers and software engineers in comprehending the fundamental characteristics, culture, and practices of agile methodologies. As a result, they are better able to select the best methods for their software project based on their requirements.

[Vijay Anand et al., 2006] discussed the many Agile approaches as well as their characteristics in his study, which are important for any firm to consider before adopting agile.

Comparison of Agile Methodologies:

Characteristic	SCRUM	XP	FDD	TDD	DSDM	KANBAN
Approach of Development	Iterative and incremental	Iterative and incremental	Iterative and incremental	Iterative	Iterative and incremental	Incremental
Time period for one iteration	2-4 weeks	1-6 weeks	<2 weeks	Reasonable Amount of Time based On product Built	2-3 days	Time taken From the start of work until Bugfix is live
Team Size	7+/-2	<20	Many Members (More than One team)	First 2/3 developers Get used to TDD. Later members are added accordingly	Independent Teams of any size	Any skilled or cross-functional team can work
Project Size	All types of projects	Small Projects	Complex projects	Small scale projects	All types of projects	Small projects usually with < 3000 LOC
Team Communication	Informal	Informal	Based on Documentation	Open communication b/w managers and developers about issues	Based on documentation	Informal face to face
Customer Involvement	Product owner acts on behalf of customer throughout the process	Throughout the process	Customer through reports	Involved only at acceptance testing	Involve in frequent releases	Customer through incremental release

Table 3. Comparison of Agile Methods Adapted from (Vijay Anand et al., 2006)

Organizations dedicated to a successful agile methodology deployment should analyze the many agile approaches to identify which methodology is the greatest fit for their firm, according to [LiverMore, 2008]. Significant approaches necessitate different changes in the management and software development cultures of the firm. The methodology implementation will be strongly influenced by choosing the approach that will provide the most benefits while needing the fewest substantial. [Anwer et al., 2017] gave a thorough examination of the many agile models utilized in the software industry. He concluded in his paper that the acceptability of agile development in the software industry has been demonstrated by a significant number of agile software development approaches. TDD, FDD, DSDM, Crystal techniques, and other agile models each have their own set of benefits and shortcomings that make them suited for a variety of project types, team sizes, and development environments. Agile software development models include various characteristics that make them ideal for present software projects, including quick response to changing requirements, give comments, early design delivery, cost reductions, and time management skills.

[Arbain et al., 2014] presented extensive research using 56 conferences, 7 journals, 14 books and 4 web site references and concluded in his paper that there is substantial evidence that integrating IT organizations that use the FDD paradigm to create secure software would benefit significantly. People in today's environment prefer to use software for a variety of tasks, such as communicating with friends, navigating via GPS, writing e-mails, handling banking activities, linking to social networking sites, and performing complex or daily transactions involving valuable information. Dangerous and unanticipated effects may emerge if a good software development procedure is not followed to generate secure software.

[Kirovska and Koceski, 2015] conducted a web survey on a software development team that use KanbanMAK (custom web-based application), and the usage of the application lasted for a period of 14 months with around 23 projects. A total of 45 team members participated in the web survey, with 88.8% responding. The results of this survey revealed that the daily development process has improved significantly, with enhanced communication and better estimation of working job performance. He concluded through the experiment that the little adjustments made when Kanban approach is employed in the current process set it apart from all other agile methodologies. Kanban approach also includes specialized metrics for stochastic monitoring and analysis. Predicting the development process improves the team's effectiveness and collaboration.

Scrum, one of the most widely used agile methodologies, is an iterative and incremental basis model for developing software with defined mechanisms, such as a module of software that may be developed in tiny pieces in an iterative manner. [Hayat et al., 2019] conducted an industrial survey of 21 distinct software organizations that are at various levels of the CMMI, have completed many projects and products on time and efficiently, and are currently working on most domestic and international assignments. The findings of the study on several elements of project-related processes are listed below.

- Project Scope Management

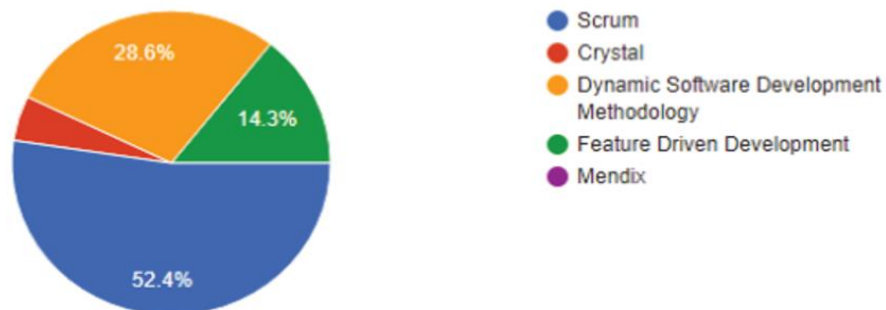


Fig 1: Influence of Scrum in Project Scope Management. Adapted from Hayat et al., (2019)

- Project Time & Cost Management

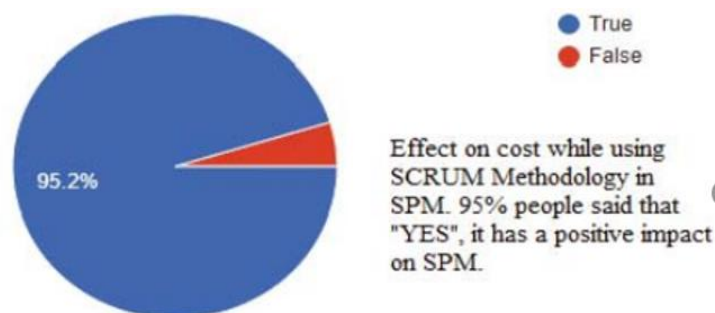


Fig 2: Cost Effect while using scrum. Adapted from Hayat et al., (2019)

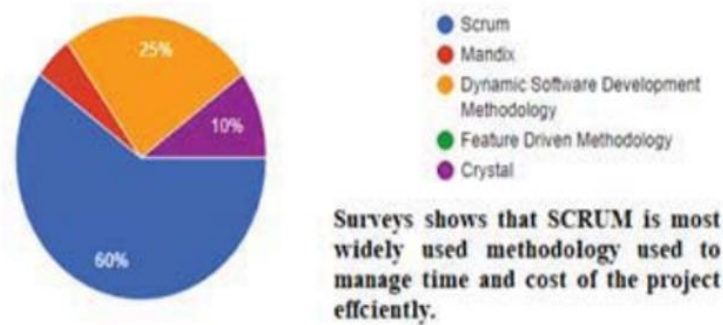


Fig 3: Influence of Scrum on Project Cost and Time Management. Adapted from Hayat et al., (2019)

- Project Human Resource Management:

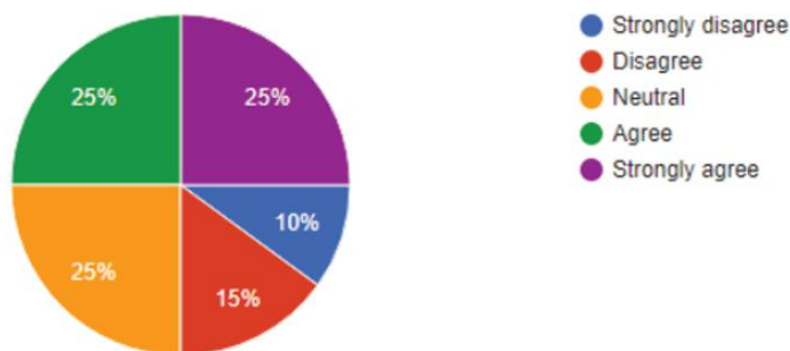


Fig 4: Influence of Scrum in Project HR Management. Adapted from Hayat et al., (2019)

Scrum has a positive impact on software project management knowledge domains, according to the findings. Scrum improves project timeliness, cost, scope, quality, risk, and scope. Scrum has a big impact on human resource management since some firms prioritize goal-oriented staff recruitment while others do not. According to the report, scrum significantly reduces risk, keep expenses under control, and deliver the best product on schedule. Cultural changes are necessary.

Hybrid Approaches

According to [Khurmann et al., 2017], Hybrid Software Development Approaches are defined as follows: “Any fusion of agile and traditional (plan-driven or rich) methods that an organization determines adopts and customizes to its specific context needs is referred to as a hybrid software development methodology (e.g., application domain, culture, processes, project, organizational structure, techniques, technologies, etc.)”.

[Stoica et al., 2016] stated in her paper that there is no "absolutely ideal" agile development process, as each project has its own set of goals and criteria. Nonetheless, Scrum and Kanban techniques have produced the Scrumban hybrid, which may be a "compromise"

option for fans and detractors of the parent methodologies. Scrumban frees development teams from project methodology constraints, allowing them to adapt to production needs and stakeholder interests. Scrumban inherits from Kanban the principle of removing parts that may result in unfavorable outcomes, hence minimizing needless operations.

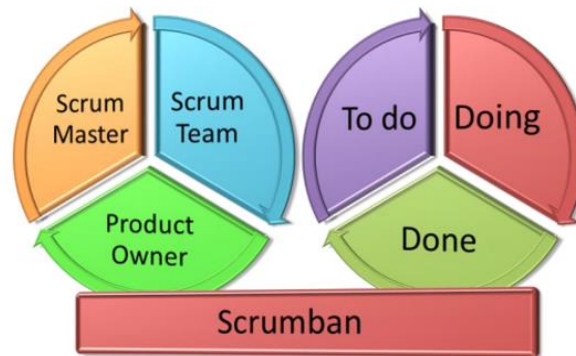


Fig 5: Scrum+Kanban=Scrumban Adapted from Stoica et al., (2016)

Many firms are now mixing agile and traditional approaches for information systems projects, according to study, and this strategy appears to support both radical and incremental capacities better [Vinekar et al., 2006]. Structure and agility, according to new research, can complement each other when employed in combination on the same task [Batra et al., 2010; Fernandez and Fernandez, 2008]. Such findings are backed up by research in the field of innovation, which implies that combining structure and chaos (planning and emergence) can lead to the most innovative results (in the case of product production) [Cunha and Gomes, 2003]. While a hybrid model could add more overhead in terms of construction documents and planning, which is typically not the central objective of agile methodologies [Karlström and Runeson, 2005], it can also save time and money. The ability to adapt the project management methodology to the scenario at hand, rather than adopting a single method to all projects [Hass, 2007], is one of the advantages of a hybrid approach. Another is the ability to focus upon value proposition rather than time and money alone [Vinekar et al., 2006]. Since conventional project management techniques are entirely defined in compared to "empirical" agile ones. According to [Fitsilis, 2008], Agile approaches place a high value on the following fields of knowledge:

- Scope Management, since the focus is on requirements management
- Human resource management, due to the emphasis on teamwork.
- Quality management is promoted using rules, testing, and periodic reviews, even when it is not expressly defined.

Agile methodologies, but at the other hand, do not properly address the following areas of knowledge:

- Risk is not explicitly controlled; Cost management isn't included in agile methodology.
- Methods for managing is not mentioned.

As a result, integrating agile approaches with the PMBOK will help the software project management community, Fitsilis concluded.

Similarly [West et al., 2011] claimed that in most Agile implementations, hybrid Agile methodologies are used. This is due in part to the fact that Agile adoption has mostly been driven by practitioners, causing teams to focus on areas where they can make a difference,

most prominently the team itself. Because elements outside of their control, like as data analysis and release management, continue to employ more traditional approaches, Scrum adoption is restricted to the development team. Compliance constraints also drive hybrid techniques, necessitating strong governance structures both before and after development.

According to [Shaydulin and Sybrandt, 2017] despite being bulky and prone to producing duplicate paperwork, it is a tried-and-true process that has proven to be capable of producing workable products. Given that the benefits of many agile approaches are, at best, fuzzy, it's reasonable that many managers choose to come down on the side of caution. Because no single methodology can fully cover all parts of the software development process. Using a combination of methods can help you achieve better outcomes. By using waterfall for project planning, scrum for short-term goals, and TDD for software accuracy, a team can avoid the problems of each method while showcasing their strengths. Scrum and Kanban were combined with Waterfall to establish a hybrid framework termed 'Scrumbanfall,' with Kanban at the core of integration and inside the bounds of Scrum principles, according to [Bhavsar et al., 2020]. Scrumbanfall is a cross of Waterfall, Scrum, and Kanban, each with its own set of benefits and drawbacks when it comes to tackling software development difficulties as a single framework. Scrumbanfall inherits all their strengths in order to empower its capability in achieving Software Engineering Management (SEM) goals.

Similarly, according to [Masood and Farooq, 2017], the agile methodology provides numerous benefits for enterprises, project teams, clients, and partners. Individuals and teams are developed, performance visibility is improved, and creativity and innovation are increased. Re - work costs are minimized, tasks can be accomplished faster, customers are satisfied, individuals and groups are developed, performance visibility is improved, and creativity and innovation are increased. Even with these benefits, APM is not a fail-safe project management strategy. Scheduling work, quality control, knowledge management, major project management, distant team management, and people management are all challenges with this strategy. In his paper, he suggests that the project team and organization consider integrating some elements of traditional project management, such as project planning documentation and the creation of a formal project organization structure, to address these issues. Project teams will be able to track quality, knowledge, and timeframes more easily with more APM documentation. Coordination will be improved with a proper organizational structure, especially in large, multi-site projects.

3 Choosing the right Project Management Method

Even though it is obvious and for organizations to quickly adapt agile to go in parallel with other organizations and respond to client needs, [Spundak, 2014] stated in his article that both traditional and agile techniques have advantages and disadvantages when compared to specific project features. For a single project and inside a single methodology, it is conceivable to integrate both approaches while keeping in mind when to employ which approach. It is possible to combine both approaches for a single project and within a single methodology, keeping in mind when it is better to use which approach. Approach choice must be handled carefully, taking into consideration both design features and organization - related attributes, and it is possible to integrate both approaches for a single project and within a unified framework, taking into consideration when something is better to use strategy.

3.1. How the Project managers are possibly making the decisions

The next section discusses three different approaches that scholars have proposed for evaluating the strategic suitability of an agile strategy.

3.1.1. Agility Measurement Index

The following section discusses three alternative ways for evaluating the tactical viability of an agile approach that experts have proposed. Because they're the only ones available that included a high-level approach to agility, these three tools were chosen. We sought for general characteristics that would aid us in determining the optimum project structure early on.

An Agility Measurement Index [Datta, 2009] is a metric for determining whether Waterfall, UP, or extreme Programming (XP) should be used. The waterfall technique is a plan-driven approach, the UP method is a medium method, and the XP method is an agile method. Five aspects of a software project, according to the author, should be considered:

Duration: The length of time till the deadline.

Risk: The impact of the project deliverable if it malfunctions while in use.

Novelty: If the context for this type of goods is completely new.

Effort: The amount of time the consumer is willing to invest in the project.

Interaction: The extent to which project participants and customers engage.

Each dimension is given a minimum x score and a maximum y score based on how complex the numerous parts of each organization are. Following that, a grade is involved in the project (based on the range between x and y). Each dimension is given a minimum x score and a maximum y score based on how complex the numerous parts of each organization are. Following that, a grade is assigned to the project (based on the range between x and y). The Agility Measurement Index, or AMI, has the following definition:

$$AMI = \frac{\sum a_i}{\sum y_i}$$

Adapted from Royce, (1970)

A low score indicates that the project will be completed quickly and with minimal risk, hence a Waterfall method is recommended [Dutta, 2009].

3.1.2. Using Risk to Balance Agile and Plan driven Methods

Another method to consider these different types of projects is to consider the risks of using one or the other for a particular project. Their first step is to do a risk analysis to determine whether the project falls into the agile or plan-driven categories [Boehm et al., 2005]. The following dimensions were used in this evaluation:

1. Application: Primary goals, Size, and Environment.
2. Management: Customer relations, Planning and control, and Communications.
3. Technical: Requirements, Development, and Tests.
4. Personnel: Customers, Developers, and Culture [Boehm et al., 2005]

In terms of aspect 1, the agile home-ground is swift value and change response. Predictability, stability, and high assurance are the characteristics of the plan-driven home-ground in this scenario. In comparison to strong plan-driven, agile projects should have fewer teams. The environment is similarly tumultuous, with a high rate of change and a greater focus on the individual project. Aspect 2 is that in agile, client relationships are close and focused on prioritized increments, whereas in plan-driven, customer interactions are as-needed, and contracts are the focus. In agile projects, the planning and control systems are more qualitative, while in plan-driven projects, they are more quantitative. In terms of communication, the agile project appears to have tacit interpersonal knowledge, whereas the plan-driven project appears to have explicit written information. In agile, needs are more informal fast changing user stories, but in plan-driven, requirements are established on an exhaustive design with a foreseeable requirements evolution. Agile development is done in small steps with the notion that refactoring is inexpensive. Agile does not have established test plans and methods, whereas plan-driven does. Aspect 4 claims that agile projects have devoted collocated customers, engineers who are far more technically skilled, and a culture of comfort, empowerment, and a wide range of flexibility. Purely plan-driven projects thrive on order and enjoy the security and empowerment that laws and procedures provide [Boehm et al., 2005]

3.1.3. Agile Adoption Framework

[Sidky, 2007] advises using the Agile Adoption Framework to determine which agile approaches an organization is ready to implement. He justifies its use by claiming that, while agile development has many success stories, they are not truly generalizable. Sidky then challenges the paradigm proposed by Boehm since it only addresses agility in its broadest sense and not specific practices. According to Sidky, the move to agile principles is difficult in several ways. These are intended to:

1. Provide structure to a complicated and unpredictable process such as agile development.
2. Independent of agile approaches, measure and assess agility.
3. Consider project and organizational features that encourage agile adoption.
4. Make sure the framework is efficient and effective in guiding adoption efforts

[Sidky,2007] then asserts that his technique addresses all these issues. The strategy is built around a two-part tool. The Agile Measurement Index (the same name as Datta, but it is a separate tool) is the first part, and it serves the objective of-

- A tool for assessing and measuring an organization's agile potential, independent of any particular -agile strategy.

-A scale for determining the target level of agility.

-When it comes to arranging and grouping agile activities in an organized way based on fundamental agile attributes and business principles, this tool comes in handy.

-Ability to give a hierarchy of measurable indications for determining an organization's agility.

The second element involves applying the Agile Measurement Index in a four-step procedure to determine whether there are any discontinuing factors, a project-level evaluation, an organizational readiness assessment, and a reconciliation phase. The high-level discontinuation factors are the focus of this research. The ceasing factors (also known as deal breakers or showstoppers) are important overall aspects that must be present for agile software development to succeed. These are:

- Inappropriate Agility Need
 - Budgets and schedules for previous projects
 - Issues with the existing software development methodology
 - Change in Project Requirements Rate
 - Project Time to Market is Required
- Scarcity of prudent funds
 - Funding Availability
- Executive Support is Missing
 - Executive Leadership Buy-in

3.2. How the decision can be taken with respect to the Contingency theory

In a project organization, agile transformation responds to a variety of contingency variables and conditions in each corporate environment and environmental situation. Changes in client expectations and needs, as well as rival movement, cause this reaction. The move from traditional to agile project management methodologies is an example of far-reaching changes in managerial methods prompted by changing situational factors in the project organization's environment. In connection to the usage of agile transformation processes, there are both externally and internally contingency elements that can be recognized and characterized. The context of contingency theory is important in the agile transformation process because it facilitates the understanding of issues, obstacles, conditions, the nature of the process itself, and the results of changes made in project management methodologies from an optimistic theoretical point of view and uses the same studies to initiate some workable positive changes in project management plan.

Theoretical Perspective

In terms of the third component – some environmental contingency variables that have an impact on this organization – Using contingency theory as a theoretical perspective, a model for adopting a new project management technique (an independent variable) on substantial change (a dependent variable) across the entire software project organization was presented. [Joslin and Müller, 2015]. A shift from traditional regulation to Agile governance in project management is an example of a thorough change that is linked with the individual environment context aspects of each software system organization. Both external and internal contingency factors influenced the transition, and their evaluation offered a better and clear view of the effects of methodology transformation from a

favorable theory point of view even while suggesting a few useful advancements from an institutional theory perspective [Joslin and Müller 2015]. Knowledge management practices and continuous learning activities were recognized by [Cegarra et al., 2016] as context variables driving organizational adoption of company growth. Using the contingency theory perspective, [Sherehiy et al., 2007] provided a set of conceptual frameworks for organizational agility manufacturing. They argue that organizations can't be thought about or examined in separation from their environment [Sherehiy et al., 2007]. Organizations should also adapt its features and practices to these contingency elements in order to preserve their effectiveness [Sherehiy et al., 2007; Denning, 2016a]. The theory of constraints (TOC) method, on the other hand, was recognized by [Gurd and Ifandoudas 2014] as a short-term improved organizational agility. TOC is a continual improvement system for managing restrictions to improve the overall performance of a system [Gurd and Ifandoudas, 2014]. Simultaneously, they advocated addressing some major strategic problems, enabling a more agile environment, and shifting toward an agile corporate culture as long-term organizational goals utilizing a modified scorecard based on agility principles (Gurd and Ifandoudas, 2014).

Research Problem

Making decisions is a continuous process that occurs throughout a person's life. [Tamanini et al., 2015]. Some options are straightforward, but others necessitate extra investigation due to the large number of variables at play. Each decision's influence is primarily assessed considering two opposing viewpoints: one for and one against it. As a result, the exposure of the advantages and disadvantages, as well as their overall analysis, serves as the foundation for decision-making. The term "Multiple Criteria Decision Analysis" refers to an examination of the benefits and drawbacks from several perspectives (MCDA). The examination of these scenarios aids in the generation of knowledge about the decision context, which boosts the confidence of those who make decisions. [Evangelou et al., 2015].

We are attempting to fill a gap in the literature by examining how project managers make decisions about which project management approach to use. Several studies have been conducted to determine which methodology is the best. There have also been various studies on how agile approaches are highly recommended in organizations for project delivery success. and in today's quickly changing market environment, businesses employ agile approaches. [Gren et al., 2019]. Agile methodology is beneficial due to improved communication among team members, quick releases, and design flexibility, as demonstrated by [Begel and Nachiappan, 2007], using a survey-based approach. Scrum methodology is the most widely used, while test driven development and pair programming are the least common. Formal methods are used in agile software development by [Zuo et al., 2010]. They used rCOS, an object-oriented approach to agile methodology, to improve system analysis accuracy and to facilitate system development with object-oriented ideas. According to [Livermore, 2007], XP is widely used in the organization. Scrum is then used, followed by feature driven development, dynamic development methodology, adaptive software development, and other agile methodologies that are customized for the company. Agile methods are intriguing because they appear to be successful. Agile methods are a hot topic in the IT sector, and it is one of the most essential trends. Agile approaches are described as "*examples of ostensibly huge success stories*" in academic literature. [Agerfalk et al., 2009]. [Qumer and Henderson 2008] argue that the phenomena has resulted in "*unprecedented transformations in the world of software engineering.*" Agility does contribute to project success (primarily efficiency and overall stakeholder satisfaction), according to a major empirical study from 2015 that included 1002 projects [Serrador and Pinto, 2015].

Agile approaches, on the other hand, are appropriate for some development environments but not others, according to a study [Pirjo et al., 2011]. For projects with large teams, well-defined requirements, clients that want high confidence, and vast code bases, the typical plan-oriented project profile works effectively. As a result, Agile methods work best when the team is limited, the goals aren't fully specified, the project code base is tiny, and the customer expects rapid progress. Agile methods alone are insufficient as a software project grows from a small prototype to a large sustainable society with a huge team, with commitments to keep and deadlines to meet, and some other mechanism is necessary. XP is difficult to deploy in large, organizational hierarchies, but easier in other types of organizations, according to [Dyba and Dingsoyr, 2009]. Pair programming is expensive, and XP is best suited to experienced employees. There is also a lack of attention paid to design and architectural issues. [Misra et al., 2006] created a framework for identifying the important modifications that must be made and the obstacles that must be overcome for traditional development organizations to embrace agile software development methodologies. [Syed et al., 2006] discovered that agile approach is more people-oriented than process-oriented in a more dynamic situation. Apart from the creators' delight, this is only useful whenever the needs are ambiguous or variable.

Therefore, [Agerfalk et al., 2009] have underlined that research has yet to provide answers to the questions of how, why, and in which settings agile methodologies genuinely work. This presents difficulties for managers who must determine whether to use a plan-driven or agile project methodology. Because theory on agile methods is generally lacking [Dings et al., 2012], no dominant theoretical position on the understanding of agile methods, such as when to use an agile project approach instead of a traditional plan-driven method like Waterfall, has been formed.

Most firms that claim to use an agile approach to project management use a combination of agile and plan-driven approaches [West et al., 2011]. This mix isn't always a negative thing, as the corporate context, for example, may be less suited to a pure agile process. There are numerous characteristics that influence whether a plan-driven or agile method is better suited for a given project, but none of the ones found in the public domain have any empirical support. [Gren et al., 2019].

There is no practitioner decision model that facilitates the selection of an acceptable procedural model for a specific project based on desk research and primary data collection. (Thesing et al., 2021). He conducted substantial data analysis and an empirical survey using the decision model, advancing the state-of-the-art and trying to present a model that methodically claims to support the selection of an acceptable procedural model for a certain project. The results of the interviews conducted are not representative and can only be applied to select businesses because the qualitative research with 15 experts represents a small database. The framework should be applied in consideration of the project's unique circumstances. The cause-and-effect relationships that govern the selection of the procedural model and project success have not been empirically quantified, and they are dependent on the context of the companies.

Similarly, in respect to previously discovered determinants, [Gren et al., 2019] explored how decision-makers select between agile and plan-driven ERP implementation projects. He identified initiatives that decision-makers thought was successful but on the edge of being successful, as well as agile and plan-driven projects that were already in place, and had project managers fill out a survey with key agile and plan-driven features. Except for increased executive buy-in for process change and the priority of low-cost projects in agile initiatives, he discovered that decision-makers' assessments did not differ in any manner. His research, on the other hand, was limited to the ERP industry.

As a result, the goal of the study is to uncover crucial elements based on contingency factors that can help decision makers choose between agile, plan-driven, or hybrid approaches based on team member characteristics. It is critical for decision makers, who are typically project managers

in the IT team, to determine which methodology is best suited to the project at the start of the SDLC.

Research Methodology

Contingency Theory

Because there are no universal management procedures, the contingency theory's central concept is that organization management should use a unique contingency approach. Each organization has a unique profile, operates in a unique environment, necessitates special circumstances, and acts in a unique context, all of which are influenced by several contingency factors. The contingency theory assumes that there is no generic set of management procedures, that those that exist are influenced by a variety of contingency factors and contextual variables, and that they are only appropriate to certain scenarios and conditions [Otley, 1980]. [Hambrick and Lei, 1985; Fisher, 1995] proposed the contingency view in contrast to two other radical research approaches: situation-specific view and universalistic view. Both the proximal and distant organizational contexts affect organizational operability and managerial decisions, according to the contingency approach. The purpose of contingency theory is to uncover some universal qualities and attributes of effective organizational systems, as well as how to apply them effectively to a specific group of companies in a given situation and context with established contingency factors [Otley, 1980]. The contingency theory is used to study effective leadership, effective incentive systems, adaptable organizational structure, adaptive management techniques, project management, and strategic management accounting [Simon, 2007]. Positive theory empirical research results in an impact assessment of current contingency factors on presently used approaches, whilst normative theory empirical research results in the search for an offering of a sensible approach, as well as advancements in organization and management methods. Contingency theory empirical research focuses on discovering both external and internal contingency elements that influence performance, as well as developing appropriate management methods [Paterek, 2017].

As I have already mentioned in the paper, the research methodology is based upon identifying some internal and external factors based on the literature which is shown in Table 4 Internal contingency factors identified including Size of the team, Project Time Frame, Project result Criticality, Organizational/Corporate Culture, Team member rotation (Stability), Project Uncertainty, Project Activity dependencies and Team member Skills to follow and adapt a methodology

Leadership style, attractive compensation systems, flexible organizational structure, adaptive management techniques, project management, and strategic managerial accounting are all studied using the contingency theory [Simon, 2007]. Positive theory empirical research leads to an examination of current contingency variables' impact on currently employed approaches, whereas normative theory empirical research leads to the search for an offering of a practical strategy, as well as developments in organizational and management methods. The goal of empirical contingency theory study is to identify both external and internal contingency aspects that affect performance, as well as to design appropriate management approaches [Paterek, 2017].

Internal Factors	External Factors
Size of the team	Digital Transformation
Project Time Frame	Globalization
Project result Criticality	Onshoring and Offshoring
Organizational/Corporate Culture	
Team member rotation(Stability)	
Project Uncertainty	
Project Activity dependencies	
Team member Skills to follow and adapt a methodology	

Table 4. Internal and External contingency theories for Agile transformation referenced from Paterek (2017) and Jovanovic et al., (2017)

The technique of identifying the project's contingency factors would imply that the organization would assess the project's contingency factors before beginning the transformation. The desired transformation method can be chosen based on the specified project context (factors listed in Table 4). (Plan driven, hybrid or Agile). The chosen technique then serves as the Agile transformation's goal. Table 5 and 6 lists the factors that should be considered when determining the best strategy for a concrete project. Multiple research investigations were used to develop the eight criteria for the internal contingency factors and three external contingency factors. The recommended list of factors could be used to assess project contingencies. As a transformation objective, an appropriate approach will be chosen based on the evaluation of contingency factors.

For example, with reference to table 5, if it can be determined that a project's delivery time frame is short (factor 2), team member rotation is low (factor 5) and the number of modifications is large - more than 50 per month (factor 6), the "Agile technique" should be applied to that project. Ideally, the organization will be able to assess all eleven elements listed in Table 5, but if this is not possible, the more factors assessed, the better. Hybrid methods are a combination of Agile and plan-driven approaches, with just some Agile principles and practices being used in the existing business, depending on the desired goal. If the outcome of the contingency evaluation is that the organization should change to Agile, the "Agile technique" will be chosen and applied.

Internal Contingency Factors	Agile Method	Plan Driven method
Size of the team	Small teams with less than 40 members	Big team with more than 40 members
Project Time Frame	Short delivery time frame	Easily reachable delivery date
Project result Criticality	Low Criticality(financial)	Highly critical(Impact on human life)
Organizational/CorporateCulture	Accepting and encouraging the flexibility of procedures	Accepting and encouraging the stability of procedures
Team member rotation(Stability)	Lower level of changes	Higher level of changes
Project Uncertainty	Big number of changes	Small number of changes
Project Activity dependencies	Most of the activities in parallel	Most of activities in sequence
Team member Skills to follow and adapt a methodology	Majority of members with high expertise	Majority of members with low expertise

Table 5: Internal Contingency factors to evaluate the project management methodology in IS

Size of the team- [Cockburn, 2001] argued that size is an issue in agile methodology. As a company expands in size, synchronizing interfaces becomes a major concern. After 20-40 individuals, agile with face-to-face communication breaks down and becomes more difficult and complex. [Lindvall et al., 2002]. Most papers demonstrating the success of agile life cycles cite real-world examples, although these examples are often from short or low-risk projects, often in smaller firms [e.g., Cockburn and Highsmith, 2001; Pikkarainen et al., 2008]. Small projects like the ones detailed in these articles allow teams to maintain infrastructure that is tailored to project interdependencies. The project team in one scenario analyzed by [Pikkarainen et al., 2008] comprised of only six people. Team members' duties in developing a security management system were mutually linked, necessitating regular informal communication between them. Because of the modest size of the team, agile tactics such as open office spaces and informal team meetings were successful in satisfying project requirements. As a result, agile development works best for small or medium-sized collocated teams [Boehm and Turner, 2003; Lindvall et al., 2004].

Large or complicated projects, as well as mature organizations, are more difficult to deploy agile approaches in. Projects necessitate a great deal of coordination since large, complex projects and organizations have so many interdependencies. As a result, in these circumstances, less expensive forms of coordination, such as standardization and planning, are easier to manage. [Barlow et al., 2011] In other words, most people working on major projects and organizations find it difficult to perform large-scale mutual adjustment coordination. Agile approaches are typically not preferred for large, complicated projects or established organizations because they rely on mutual adjustment.

Time frame- Agile development models stress "agility" and "adaptability" in development, as opposed to waterfall development models' "set-in-stone" approach. Instead of a single time-consuming and inflexible development schedule, agile models use many incremental development plans that aim to improve the output with each iteration. A time frame is the amount of time that a project is expected to be finished within. The Waterfall method can be employed when the time frame is lengthy enough, however rapid delivery projects should be handled in a "Agile" manner, according to [McCormick, 2012].

Project Result Criticality- [Cockburn, 2001; Boehm et al., 2003] have founded out that since the waterfall or the traditional methods works on proper documentation, Agile is Untested on safety-critical items; simplistic design may cause issues and documentation is lacking while plan driven methods are developed to handle extremely critical products are difficult to adapt to low-criticality products efficiently.

Organizational/Corporate culture- As mentioned by [Bohem et al., 2003] that the plan driven methodology in is procedure oriented and is stable with the procedures while agile is flexible, The agile approach can be difficult to embrace in a culture that depends heavily on legacy systems, or in a culture driven by control. [Zerfab, 2018]. According to [Arokodare et al., 2021], a flexible organizational structure, a strong organizational culture, leadership unity, and entrepreneurial management abilities are all required of an agile organization.

Team member rotation/Stability- Our suggestion to choose between agile or plan driven with respect to team member rotation is that to use agile in IT only when the rotation is low. Agile development approaches are built on "*intensely iterative procedures*" [Austin and Devin, 2009, p. 463], which means that teams rigorously analyze, design, and code in short intervals; meet with the customer to evaluate their work and seek feedback; and then repeat the cycle. This process differs significantly from plan driven methods, in which a lengthy design step precedes a lengthy coding phase, which is followed by a lengthy testing phase. [Barlow et al., 2011]. It can be problematic in an agile environment if the team is not solid with the members or if the team members change frequently before the project is completed, because agile does not follow procedures based on documentation, which may not be viable for the new team member.

Project Uncertainty- Unexpected changes in scope, errors in project plans, or team member competency can all lead to uncertainty in the software development process [Schmidt et al., 2001]. Mutual adjustment is a popular method for coordinating reciprocal interdependencies. It refers to ad hoc, informal contact that occurs outside of formal project planning and often continues throughout the project [Thompson, 1967]. Plan-driven methodologies assume that project interdependencies are generally sequential and can be managed through planning and evaluation. Many interdependencies in a software project's life cycle, on the other hand, are reciprocal in nature. As a result, some of the time and money invested in building detailed plans has been wasted, and some mutual adjustment is required. Agile methodologies, on the other hand, adopt a different approach. They consider the majority, if not all, project interdependencies to be reciprocal, and hence use mutual adjustment to coordinate all project interdependencies. In other words, they place less emphasis on formal, upfront planning and coordinate on-the-fly as needed, according to [Barlow et al., 2011]. Therefore, agile methodologies can be adapted when there the project demands uncertainty.

Project activity dependencies- Agile development approaches are built on "*intensely iterative procedures*" [Austin and Devin, 2009, p. 463], which means that teams rigorously analyze, design, and code in short intervals; meet with the customer to evaluate their work and seek feedback; and then repeat the cycle. This process differs significantly from traditional methods, in which a lengthy design phase is followed by a lengthy coding phase, which is followed by a lengthy testing phase. [Barlow et al., 2011]

Team member skills to follow and adapt a methodology- The main difference between agile and traditional development paradigms is that agile emphasizes reacting well to changing project needs [Austin and Devin, 2009]. Plan-based approaches proponents argue that investing time on design is vital to developing a strong system that meets specifications, and that losing flexibility is a necessary trade-off, according to [Barlow et al., 2011]. On the other hand, proponents of agile methods argue that a system developed using an iterative method can still meet all requirements and adhere to proper design standards because regular iterations allow for repeated refinement and a final product that is closer to the actual desires of customers. Agile methodologies also foster a focus on the customer's needs. Because agile approaches are adaptable, customers' desired modifications in plans or wants can often be accommodated in the next iteration of development. Practitioners claim that agile methodologies help them to stay focused on the customer's needs and shifting wishes throughout the project [Cockburn and Highsmith, 2001]. As a result, we conclude that agile approaches are feasible for IT project teams with sufficient expertise to react to change.

External Contingency factors	Agile method	Plan driven method
Digital Transformation	When the Time and budget is limited. It is helpful when there is a need of innovation at any stage of the project life cycle.	No limitations on time and budget. When the transformation is a prerequisite at the beginning of the project
Globalization	When the team is highly effective and efficiently communicate as agility relies heavily on close cooperation and shared understanding	When there is a need of sharing written documents to communicate within the team or the product owners
Onshoring and Offshoring	When the subcontractor is flexible with the pricing estimates, have a good track record and can be trusted	Plan on bidding the contract, the subcontractors have a process, with milestones and deliverables, in sufficient detail to determine a cost estimate.

Table 6: External Contingency factors to evaluate the project management methodology in IS

Digital Transformation- The digital transformation has brought with it new challenges as well as opportunities to improve project management methods [Denning, 2016b]. Software project firms are facing organizational changes connected to the introduction of new and modern project management methodologies to cope with increased market rivalry, changing consumer expectations, a dynamic marketplace, and to improve their organizational business agility [Gurd and Ifandoudas, 2014; Cegarra et al., 2016]. [Corejova et al., 2020] conducted a case study to determine if agile or plan-driven methodologies are more effective for digital transformation in an IT firm. According to the findings, firms that use agile approach have a total cost that is 64% cheaper than those who use the waterfall style. From the start, an agile model necessitates the involvement of system experts. To keep costs low, team turnover is higher. The agile model has a 34.8 percent lower cost per day than the waterfall model. He also validated that processing requirements in agile-driven projects takes less time than in waterfall-model initiatives. In comparison to the traditional paradigm, the project's overall time was reduced by about 45 percent. Time and money were saved by reducing the number of hours spent on project coordination.

Globalization- Through the continual rise of globalization in the digital era, customer companies are confronted with new market dynamics. [Burchardt et al., 2019]. Globalization, and the resulting requirement to be

highly linked, has resulted in an increase in the use of information technology in business [Guaragni et al., 2016]. He stated in his article that because there are few and extensive iterations in the traditional product development process, it is process-oriented and primarily linear. The project team is led through command-and-control by a centralized instance. Because formal channels are heavily used for communication, knowledge is exchanged explicitly rather than implicitly, making the project more bureaucratic. While Traditional methodologies is responsible for delivering what has been formally agreed upon in contracts with stakeholders, Agile is responsible for identifying the optimal solution to meet user needs. Agile methods do this by creating prototypes in short iterations and gathering user feedback by providing working product increments rather than technical documentation. One of the key drivers of Agile Project Development is the creation of user value.

Offshoring and Outsourcing- Outsourcing and offshoring software development work to subcontractors is frequently based on contracts that detail exactly what the subcontractor is expected to do. In circumstances when subcontractors must compete for the contract, subcontracted tasks must be well-defined. A subcontractor will often establish a plan that includes a procedure, milestones, and deliverables, in sufficient depth to calculate a cost estimate, while preparing a bid. The process may be iterative and incremental, but in order to compete, the subcontractor may need to make the process predictable by specifying the number of iterations and deliverables for each iteration. [Turk et al., 2014]

According to Turk, it's feasible to write a contract that gives a subcontractor considerable leeway in how they build the product while staying under budget and time limits. This is conceivable if the subcontractor has a proven track record and can be trusted by the contracting business to develop a product that meets the contracting company's requirements. Since agile projects are dynamic and changes with the client requirement, it might be difficult to deal with the employees with no proper documentation of the process and requirements.

The agile approaches, Plan driven and Hybrid, are shown in Figure 6 to be dependent on the external and internal contingency theories given in Table 4. To employ hybrid techniques, project managers must carefully consider the contingency considerations and determine whether the project is appropriate for a hybrid approach. For instance, If the team size is high, the project criticality is low, and there is a requirement for innovation at several stages, the hybrid approach may be appropriate.

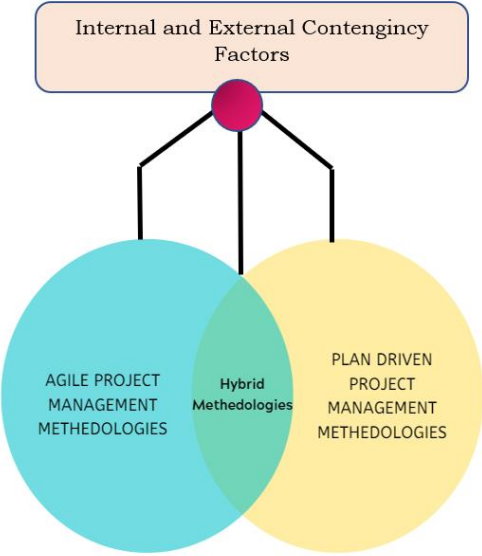


Fig 6: Theoretical model Source: Author

Conclusion

When it comes to agile development, there are a few things that developers should keep in mind. They should start by making incremental modifications to the requirement, project plan system, and generating artifacts to enable agility. They should also solicit feedback from all stakeholders and participants to ensure that the initiative meets their needs. In addition, only those tasks that provide value to the business processes supported by the system should be completed, and procedures and artifacts that do not offer long-term value to the working software system should be discarded. [Carayannis,2002]

As we have seen in the paper, there is no one-size-fits-all project management methodology for organizations. At the start of the project, the nature and characteristics of the project, as well as the best suited methodology, must be thoroughly examined.

Although it is difficult for project managers to ignore the agile wave, scholars have described how traditional project management should be considered for certain projects in their papers. Although agile methods have been shown to be superior in many aspects such as robustness and flexibility, literature suggests that they are not ideal when the project size is large. On the other hand, there are numerous research papers and findings that demonstrate how agile methods can help accelerate successful project development.

As previously stated, there are several agile methods, and organizations must carefully select which agile method to use for the project. Again, the project's characteristics would lead managers to select agile methodologies. There are also a number of hybrid methodologies that are gaining traction. Hybrid methods can be a combination of traditional and agile methodologies, or they can be a combination of agile methodologies themselves.

In this paper, I propose a theoretical model based on research in which project managers or decision makers can select which methodologies to use in the organization. As we can see, there has been very little research done to allow project managers to decide which methodology would be best for the project. As a result, I have proposed a model here that includes internal and external contingency factors. The factors are chosen from the literature based on the client's needs and the project's characteristics. If all of the characteristics of a project support agile, agile must be chosen, and vice versa. If some characteristics point to agile and others to traditional, managers should carefully analyze and select a hybrid approach that is suitable.

However, there are some limitations to the study. First, the research can be expanded to include case studies in which project managers and business analysts are interviewed to learn about their perspectives on agile methodologies and how they are used in their organizations. This would result in a better understanding of the organization's attitude toward embracing new methods. Second, future research can be conducted to focus more on hybrid project management methodologies, and which are best suited for organizations.

Bibliography

1. Agerfalk, P.J., Fitzgerald, B., Slaughter, S.A.: Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research* 20(3) (2009) 317–328
2. Anand, R. V., & Dinakaran, M. (2016). Popular agile methods in software development: Review and analysis. *International Journal of Applied Engineering Research*, 11(5), 3433-3437.
3. Anwer, F., Aftab, S., Waheed, U., & Muhammad, S. S. (2017). Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. *International journal of multidisciplinary sciences and engineering*, 8(2), 1-10.
4. Arbain, A. F., Ghani, I., & Jeong, S. R. (2014). A systematic literature review on secure software development using feature driven development (fdd) agile model. *Journal of Internet Computing and services*, 15(1), 13-27.
5. Arokodare, M. A., & Falana, B. R. (2021). Strategic Agility and the Global Pandemic: The Agile Organizational Structure, A Theoretical Review. *Information Management and Business Review*, 13(1 (I)), 16-27.
6. Austin, R. and L. Devin (2009) "Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design", *Information Systems Research* (20)3, pp. 462–477.
7. Barlow, J. B., Giboney, J., Keith, M. J., Wilson, D., Schuetzler, R. M., Lowry, P. B., & Vance, A. (2011). Overview and guidance on agile development in large organizations. *Communications of the Association for Information Systems*, 29(2), 25-44.
8. Batra, D., Xia, W., VanderMeer, D., and Dutta, K. (2010) "Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry," *Communications of the Association for Information Systems*, Vol. 27, Article 21, pp. 379-394
9. Begel, A., & Nagappan, N. (2007, September). Usage and perceptions of agile software development in an industrial context: An exploratory study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (pp. 255-264). IEEE.
10. Bhavsar, K., Shah, V., & Gopalan, S. (2020). Scrumbanfall: an agile integration of scrum and kanban with waterfall in software engineering. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9(4), 2075-2084.
11. Boehm, B. Get ready for agile methods, with care. *Computer* (Jan. 2002), 64–69
12. Boehm, B., Turner, R.: *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison Wesley, Reading (2003)
13. Boehm, B., and Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE software*, 22(5), 30-39.
14. Burchardt, C., & Maisch, B. (2019). Digitalization needs a cultural change—examples of applying Agility and Open Innovation to drive the digital transformation. *Procedia Cirp*, 84, 112-117.
15. Cao, L., K. Mohan, and B. Ramesh. (2004) "How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects", *Proceedings of the 37th Hawaii International Conference on System Sciences*, Waikoloa, Hawaii, pp. 1–10.
16. Carayannis, E.G. *Agile Project Management for IT Project*. Greenwood Press / Quorum Books. 2002
17. Cegarra-Navarro, J.-G., Soto-Acosta, P. and Wensley, A.K.P. (2016) "Structured knowledge processes and firm performance: The role of organizational agility", *Journal of Business Research*, Vol. 69, No. 5, pp. 1544-1549.
18. Charvat, J. (2003). *Project Management Methodologies: Selecting, Implementing, and Supporting Methodologies and Processes for Projects*. Hoboken, NJ: John Wiley & Sons, Inc.
19. Cheema, A. & Shahid, A.A. (2005). Customizing Project Management Methodology. 9th International Multitopic Conference, IEEE INMIC 2005, Karachi, 1-6
20. Chin, C.M.M. & Spowage, A.C. (2010). Defining & Classifying Project Management Methodologies. *PM World Today*, 12(5).

21. Corejova, T., Bielik, P., & Genzorova, T. (2020). Approaches to Project Management in the Process of Digital Transformation in the Company. *ENTRENOVA-ENTerprise REsearch InNOVAtion*, 6(1), 531-541.
22. Cunha, M. P. and Gomes, J. F. S. (2003) "Order and Disorder in Product Innovation Models," *Creativity and Innovation Management*, Vol. 12, No. 3, pp. 174-187.
23. Cockburn, A. & Highsmith, J. (2001). *Agile Software Development: The People Factor*. *Computer*, 34(11), 131–133
24. Coram, M., and Bohner, S. (2005). The impact of agile methods on software project management Paper presented at the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Greenbelt, MD USA
25. Datta, S.: *Metrics and Techniques to Guide Software Development*. PhD thesis, Florida State University College of Arts and Sciences (2009)
26. Denning, S. (2016a) "How to make the whole organization 'Agile'", *Strategy & Leadership*, 44 (4), pp. 10-17
27. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software* 85(6) (2012) 1213–1221
28. Dyba, T., Dings_yr, T.: Empirical studies of agile software development: A systematic review. *Information and software technology* 50(9) (2008) 833{859
29. Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development?. *IEEE software*, 26(5), 6-9.
30. Evangelou C, Karacapilidis N, Khaled OA. Interweaving knowledge management, argumentation and decision making in a collaborative setting: the KAD ontology model. *Int J Knowl Learn* 2005;1:130-145.
31. Frederick P. Brooks. 1987. No Silver Bullet Essence and Accidents of Software Engineering. *IEEE Computer* 20, 4 (1987), 10–19.
32. Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management—agilism versus traditional approaches. *Journal of Computer Information Systems*, 49(2), 10-17.
33. Fisher, J., 1995. Contingency-Based Research on Management Control Systems: Categorization by Level of Complexity. *Journal of Accounting Literature*, 14, pp. 24-53.
34. Fitsilis, P. (2008). Comparing PMBOK and Agile Project Management software development processes. In *Advances in Computer and Information Sciences and Engineering* (pp. 378-383). Springer, Dordrecht.
35. Georg Kalus and Marco Kuhrmann. 2013. Criteria for Software Process Tailoring: A Systematic Review. In *International Conference on Software and Systems Process (ICSSP)*. ACM, New York, NY, USA, 171–180.
36. Gren, L., Wong, A., & Kristoffersson, E. (2019). Choosing agile or plan-driven enterprise resource planning (ERP) implementations--A study on 21 implementations from 20 companies. *arXiv preprint arXiv:1906.05220*.
37. Guaragni, F., Schmidt, T., & Paetzold, K. (2016). Traditional and agile product development in a hyperconnected world: turning weaknesses into strengths. *Procedia CIRP*, 52, 62-67
38. Gurd, B. and Ifandoudas, P. (2014) "Moving towards agility: the contribution of a modified balanced scorecard system", *Measuring Business Excellence*, Vol. 18, NO. 2, pp. 1-13.
39. Hambrick, D.C., Lei, D., 1985. Toward an Empirical Prioritization of Contingency Variables for Business Strategy. *The Academy of Management Journal*, 28(4), pp. 763-788.
40. Hass, K. B. (2007) "The Blending of Traditional and Agile Project Management," *PM World Today*, Vol. 9, No. 5, pp. 1-8.
41. Hayat, F., Rehman, A. U., Arif, K. S., Wahab, K., & Abbas, M. (2019, July). The influence of agile methodology (Scrum) on software project management. In *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 145-149). IEEE..
42. Highsmith, J. (2003). *Agile project management: Principles and tools*. Agile project management. Cutter Consortium, 4(2).
43. Iriarte, C., & Bayona, S. (2020). IT projects success factors: a literature review. *International Journal of Information Systems and Project Management*, 8(2), 49-78.

44. Introna, L. D. & Whitley, E. A. (1997). Against method–ism: Exploring the limits of method. *Information Technology & People*, 10(1), 31–45.
45. Jeffrey A. Livermore, “Factors that impact implementing an Agile Software Development Methodology”, pp. 82-85, IEEE 2007
46. Joslin, R. and Müller, R. (2015) “Relationships between a project management methodology and project success in different project governance contexts”, *International Journal of Project Management*, Vol. 33, No. 6, pp. 1377-1392.
47. Jovanović, M., Mesquida, A. L., Mas, A., & Lalić, B. (2017, October). Towards the development of a sequential framework for agile adoption. In *International Conference on Software Process Improvement and Capability Determination* (pp. 30-42). Springer, Cham.
48. Kalus, G., & Kuhrmann, M. (2013, May). Criteria for software process tailoring: a systematic review. In *Proceedings of the 2013 International Conference on Software and System Process* (pp. 171-180).
49. Karlström, D. and Runeson, P. (2005) "Combining Agile Methods with State-Gate Project Management," *IEEE Software*, Vol. 22, No. 3, pp. 43-49
50. Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of applied economics and business*, 3(3), 25-34.
51. Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Garousi, V., Felderer, M., ... & Prause, C. R. (2017, July). Hybrid software and system development in practice: waterfall, scrum, and beyond. In *Proceedings of the 2017 International Conference on Software and System Process* (pp. 30-39).
52. L. R. Vijayarathy and C. W. Butler, “Choice of software development methodologies: Do organizational, project, and team characteristics matter?” *IEEE Software*, vol. 33, no. 5, pp. 86–94, 2016.
53. Larman, C. (2004). *Agile and iterative development: A manager’s guide*. Boston, MA: Pearson Education.
54. Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. In *International Conference on Information and Network Technology* (Vol. 37, No. 1, pp. 162-167)
55. Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., ... & Zelkowitz, M. (2002, August). Empirical findings in agile methods. In *Conference on extreme programming and agile methods* (pp. 197-207). Springer, Berlin, Heidelberg.
56. Livermore, J. A. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. *J. Softw.*, 3(4), 31-36
57. Livermore, J. A. (2007, March). Factors that impact implementing an agile software development methodology. In *Proceedings 2007 IEEE SoutheastCon* (pp. 82-86). IEEE.
58. Masood, Z., & Farooq, S. (2017). The benefits and key challenges of agile project management under recent research opportunities. *International Research Journal of Management Sciences*, 5(1), 20-28.
59. McCormick, M. (2012). Waterfall vs. Agile methodology. *MPCS*,
60. Misra, S. C., Kumar, U., Kumar, V., & Grant, G. (2006, December). The organizational changes required and the challenges involved in adopting agile methodologies in traditional software development organizations. In *2006 1st International Conference on Digital Information Management* (pp. 25-28). IEEE.
61. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 72-78.
62. Otley, D.T., 1980. The contingency theory of management accounting: achievement and prognosis. *Accounting, Organizations and Society*, 5(4), pp. 413-428
63. Paterek, P. (2017). Agile transformation in project organization-issues, conditions and challenges.
64. Paterek, P. (2018, May). Agile transformation framework in software project organization. In *ICMLG 2018: 6th International Conference on Management, Leadership and Governance* (pp. 258-267).
65. Pikkarainen, M. et al. (2008) "The Impact of Agile Practices on Communication in Software Development", *Empirical Software Engineering* (13)3, pp. 303–337.
66. Pirjo Nakki, Kaisa Koskela, Minna Pikkarainen, “Practical model for user-driven innovation in agile software development”, *Proc. Of 17th International Conference on Concurrent Enterprising*, pp. 1-8, 2011

67. Qumer, A., Henderson-Sellers, B.: An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology* 50(4) (2008) 280–295
68. Royce, W.W. (1970) "Managing the Development of Large Software Systems: Concepts and Techniques", *Proceedings of the 9th International Conference on Software Engineering, Monterey, CA*, pp. 328–338.
69. Salameh, H. (2014). What, when, why, and how? A comparison between agile project management and traditional project management methods. *International Journal of Business and Management Review*, 2(5), 52-74.
70. Sauer, C., & Reich, B.H. (2009). Rethinking IT project management: Evidence of a new mindset and its implications. *International Journal of Project Management*, 27, 182– 193.
71. Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of management information systems*, 17(4), 5-36.
72. Serrador, P., Pinto, J.K.: Does Agile work? A quantitative analysis of agile project success. *International Journal of Project Management* 33(5) (July 2015) 1040–1051
73. Shaydulín, R., & Sybrandt, J. (2017). To agile, or not to agile: A comparison of software development methodologies. *arXiv preprint arXiv:1704.07469*.
74. Shenhar, A.J., Dvir, D., Lechler, T., Poli, M.: One size does not fit all - true for projects, true for frameworks. In: *PMI Res. Conference, September 2016*, pp. 99–106 (2002)
75. Sherehiy, B., Karwowski, W. and Layer, J.K. (2007) "A review of enterprise agility: Concepts, frameworks, and attributes", *International Journal of Industrial Ergonomics*, Vol. 37, No. 5, pp. 445–460.
76. Sidky, A.: A structured approach to adopting agile practices: The agile adoption framework. PhD thesis, Virginia Polytechnic Institute and State University (2007)
77. Simon, C., 2007. A Configuration Form of fit in Management Accounting Contingency Theory: An Empirical Investigation. *The Business Review*, 7(2), pp. 220-227
78. Sircar, S., Nerur, S.P., and Mahapatra, R. Revolution or Evolution? A Comparison of Object-Oriented and Structured Systems Development Methods. *MIS Quarterly* 25, 4 (Dec. 2001), 457–471.
79. Špundak, M. (2014). Mixed agile/traditional project management methodology–reality or illusion?. *Procedia-Social and Behavioral Sciences*, 119, 939-948.
80. Stoica, M., Ghilic-Micu, B., Mircea, M., & Uscatu, C. (2016). Analyzing agile development-from waterfall style to Scrumban. *Informatica Economica*, 20(4).
81. Syed-Abdullah, S., Holcombe, M., & Gheorge, M. (2006). The impact of an agile methodology on the well being of development teams. *Empirical Software Engineering*, 11(1), 143-167.
82. Tamanini, I., Pinheiro, P. R., Machado, T. C. S., & Albuquerque, A. B. (2015). Hybrid approaches of verbal decision analysis in the selection of project management approaches. *Procedia Computer Science*, 55, 1183-1192.
83. Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus waterfall project management: decision model for selecting the appropriate approach to a project. *Procedia Computer Science*, 181, 746-756.
84. Turk, D., France, R., & Rumpe, B. (2014). Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*.
85. Veiga, A. P. (2017). Project success in Agile development projects. *arXiv preprint arXiv:1711.06851*
86. Vinekar, V., Slinkman, C. W. and Nerur, S. (2006). Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information Systems Management*, 23(3), 31-42.
87. West, D., Gilpin, M., Grant, T., & Anderson, A. (2011). Water-scrum-fall is the reality of agile for most organizations today. *Forrester Research*, 26(2011), 1-17. Wheelwright, S.C., Clark, K.B.: *Creating project plans to focus product development*. *Harv. Bus. Rev.* 2, 1–15 (1992)
88. Wysocki, R. K. (2009) *Effective Project Management: Traditional, Adaptive, Extreme*. Indianapolis:Wiley Publications.
89. Zerfaß, A., Dühning, L., Berger, K., & Brockhaus, J. (2018). Fast and flexible: Corporate communications in agile organizations (No. 5). *Communication Insights*.
90. Zuo, A., Yang, J., & Chen, X. (2010, November). Research of agile software development based on formal methods. In *2010 International Conference on Multimedia Information Networking and Security* (pp. 762-766). IEEE.