

The Case for Agile Methodologies against Traditional Ones in Financial Software Projects

V. P. Munteanu and P. Dragos

ABSTRACT

Agile methodologies have emerged to challenge traditional ones and overcome their limitations. Increasingly, software development organizations are scaling agile practices in order to meet the requirements of the quickly changing and regularly developing business environment. The main objectives of this study are to investigate the main differences between the traditional project management methodologies and agile methodologies, as well as to identify some key advantages and challenges of applying agile development in a financial software environment. The findings clearly show that using Agile methodologies in a financial software development environment increases the efficiency and transparency of the development process, as well as the stakeholders' satisfaction, but the project managers must know how to adapt different Agile frameworks to the given context of their organization.

Keywords: Agile methodologies, Traditional methodologies, Project Management, Stakeholders satisfaction, Software development.

Submitted : February 7, 2021

Published : February 16, 2021

ISSN: 2507-1076

DOI: 10.24018/ejbmr.2021.6.1.741

V. P. Munteanu

Department of Management, Faculty of Economics and Business Administration, West University of Timisoara, Romania.
(e-mail: valentin.munteanu@e-uvt.ro)

P. Dragos*

Department of Management, Faculty of Economics and Business Administration, West University of Timisoara, Romania.
(e-mail: paul_dragos2005@yahoo.com)

**Corresponding Author*

I. INTRODUCTION

Project management has the aim to deliver projects on time, respecting the initial requirements and the agreed quality, within the planned budget. The achievement of project management is perceived as planning the project accurately at the beginning and then executing the project according to this plan. Execution of the projects requires monitoring progress.

The tools, techniques, and methods used in traditional project management have been applied to software projects for years. However, in the field of software engineering, there is an inevitable factor of change that makes the majority of tools, methods, and techniques unusable as they are [1]. Initially, the strategy was to minimize or to prevent the changes by creating a better plan, better analysis. The products as well as processes have been the target for stabilization without producing desired effects [2]–[5] but it was never enough to prevent or avoid change [6]–[8]. Research studies show that software specialists spend about 30–50% of their time on rework rather than on work done right in the first time [7], [9]. In traditional fields such as construction and mining, rework is neither very common and acceptable nor physically possible, particularly after certain milestones. Once a task is completed, it is assumed that there will be no further related work. Software industry has been searching for different solutions for the substantial change issue. Starting from 1980s prototyping methods and iterative lifecycle models are developed [10] and later from, agile approaches are in focus to manage changes organizations. Nowadays, agile is used in a significant number of software projects. Agile approaches embrace the change and use it as an opportunity but do not have the variety of the tools that

exist for plan driven project management specifically for depicting the status quantitatively and establishing future estimates. To provide the best of both worlds, traditional project management methods, tools, and techniques could be adapted or replaced by more effective ones for software engineering projects [1].

The challenge now facing many financial firms is the necessity to adapt more quickly to shifts in the corporate environment and also to have faster processes for brand and project management [11]. The financial companies like banks are constantly examining the development of automated systems with agile approaches to cope with fintech organizations. The old generation software development models slowed down the processes of launching new products and services in large financial institutions [12]. How to speed up and improve the software development process and compete with fintech companies with agile models, has risen the main question for financial institutions in the last few years. However, there is no pure and perfect AGILE software development model suitable for every financial institution [13]. The model should be adapted to take into account the constraints and regulations that heavily supervised financial institutions are following, and which makes these different from fintech companies. The question is, how financial institutions can learn from the techniques used in fintech companies and find the best suitable model to compete with smaller organizations in the speed of software development [1].

The philosophy of agile methods is derived from the Agile Manifesto [14] which focuses on small team size, people and their interactions, customer collaboration, early product delivery, and efficient and effective responses to change of the

customers' demands [14]. There are several agile frameworks e.g., extreme programming (XP), crystal methods, scrum, adaptive software development (ASD), feature driven development (FDD), and dynamic system development methods (DSDMs) [15]–[17] which have been developed to implement the Agile Manifesto [18].

II. MATERIALS AND METHODS

To achieve the purpose of the study, a literature review was conducted, through which the differences between traditional and agile methods in financial software projects were assessed. Based on these findings, a project manager would be able to properly select the right methodology for a given project and organization to ensure the stakeholders satisfaction and, ultimately, the success of the project.

First of all, to understand the differences, some of these method features are briefly discussed as follows. Traditional methods are most suited to organizations that have linear structure, while agile methods are suitable to organizations that have an iterative type of structure. Traditional method is used for projects that are large-scale, while agile method is used for those projects that are usually small and medium in size. Also, there is less client involvement in traditional methods, whereas in using agile methods there is high user involvement. Furthermore, user requirements are clearly defined before coding or implementation, while in an agile method the requirement from the client is like an interactive and engaging type. Also, the restarting cost of traditional methods is very high as compared to very low cost involved in agile methods. This clearly shows the agile method is more flexible, effective and reliable for any project involved.

This research paper is also attempting to identify and evaluate comparison between traditional and agile methods and their related advantages and disadvantages with relevant secondary resources from journals and articles. Different types of Agile methodologies were analyzed and evaluated using some case studies. The adaptation of agile methods in an organization were also analyzed and discussed with their uses and importance with application in the banking sector. Also, Agile in software testing was analyzed and discussed. Moreover, secondary resources were used to evaluate the different stakeholders' satisfaction when using Agile methods.

III. RESULTS

A. The Traditional Methodologies

Traditionally, financial firms have relied on software approaches like the waterfall method [19]. Due to its vulnerability, even a minor mistake in this sector can cause the loss of millions, especially in money transfer [20]. However, many companies consider that waterfall models are appropriate for them.

There are a few traditionally used banking models software for testing, which are given below:

- Waterfall model;
- Iterative model;
- Spiral model.

The waterfall model is a software model, commonly used in the banking and finance industry. It is a popular choice

when it comes to software development. This model goes through different development stages, as it is divided into various phases like requirement, analysis, design, testing, deployment, maintenance [21]. You cannot go to the next step unless the earlier stage is completed. Each phase collects the information from the previous stage. Each of the stages has some end product, and the stages are strictly documented. The process goes on in a waterfall model. That is why this model is called the waterfall model. The waterfall is a great model for the banking and finance industry. It has been a standard approach for the firms, and it has helped many organizations to deliver successful projects. However, it is relatively slow, as it needs more time to assess the requirements [21]. It can cost more time in a lengthy task and that is why organizations are re-evaluating this method.

The Iterative model is the implementation of (SDLC) Software Development Life Cycle [21]. It emphasizes on initial, simplified implementation, which gradually becomes complicated. Whenever the iterative model is discussed, the incremental model must be addressed as it is connected with it. The incremental alterations are made during the implementation of the new iteration. In the iterative model, a project is divided into small parts, and these parts are exposed to various iterations of the waterfall model [21]. At the end of the iteration, a new module is done, or the previously made iteration is enhanced. This module is joined in the software architecture, and that's how the system is tested altogether. In this model, the stages are repeated in each iteration, and they can be reduced, unlike waterfall. The stages of the iterative model are planning, design implementing, testing, and evaluation. These stages go in cyclic form. Leading banks are moving towards digitalization. Thus, they should be working on the building blocks of the digital initiatives, and one of the most common building blocks is the iteration model. That is how these banks and firms are executing their digital strategies, the iteration model being an important model for the banking and finance industry.

The spiral model is a significant software development life cycle model. It is a famous choice among development software, due to its unique features. It is a combined model of waterfall and iterative [21]. It is cyclic and linear at the same time. It is helpful for risk handling, it looks like a spiral with a lot of circles and the number of circles or loops in this model is unknown, as it varies from model to model. Each of the circles is a phase of the software development process. The number of stages can vary, it is up to the project manager, and she can increase or decrease the number of stages depending on the project's risks. It means, the project manager determines the number of stages, the project manager playing a crucial role in the project.

The spiral model is used for ATMs in the banking sector. However, it is used in combination with the waterfall model. This model is used for eradicating the shortcoming of being time-efficient and cost-efficient. The spiral model emphasizes improving concepts to avoid the modification on a higher level in the late stages of development because of early errors in the initial stages of development. That is why the spiral model has its importance in the banking and finance industry [21].

B. The Agile Methodologies

Agile means swift or multipurpose. This approach is very common nowadays, especially in big firms [22]. In this method, the software develops in incremental, swift cycles. The interactions between customers and developers are emphasized on the processes and tools. This method focuses rather on responding to modification than on general planning [23].

According to the Agile model, every project should be handled differently, and there should be a modification in the existing methods, so they can be suitable for the project. The agile process is popular due to its unique features, like being flexible and adjustable. The agile process model is a software development approach that is based on iterative development [24]. Like the iterative model, the agile method splits the project into smaller parts, and it also involves long term planning. These small parts are given in iterations. Each of these iterations lasts for about three weeks. Each iteration contains cross-functional teams working altogether in different areas. These areas include requirement, design, construction/iteration, testing, deployment, and feedback [24]. At the end of each iteration, a product is being shown to the customers and other vital stakeholders.

The agile method gives the leverage to make a change at any point in the project to meet the project's demands. Other than that, incremental testing reduces the chances of risks. However, continuous client interaction can add extra time to all the people involved, stakeholders, clients, test teams, and development teams. The main features of Agile are compared to the features of the traditional methods in Table 1.

TABLE I: COMPARISON BETWEEN AGILE AND TRADITIONAL SOFTWARE DEVELOPMENT METHODOLOGIES [25]

	Agile Methods	Traditional Methods
Approach	Adaptive	Predictive
Success Measurement	Business Value	Conformation to Plan
Project Size	Small	Large
Management Style	Decentralized	Autocratic
Perspective to Change	Change	Change
Culture	Adaptability Leadership-Collaboration	Sustainability Command-Control
Documentation	Low	Heavy
Emphasis	People-Oriented	Process-Oriented
Cycles	Numerous	Limited
Domain	Unpredictable/ Exploratory	Predictable
Upfront Planning	Minimal	Comprehensive
Team Size	Small/Creative	Large
Approach	Adaptive	Predictive
Success Measurement	Business Value	Conformation to Plan
Project Size	Small	Large
Management Style	Decentralized	Autocratic
Perspective to Change	Change	Change
	Adaptability	Sustainability

The product has to go through different improvement processes, which involve simple minimal functionality. The process starts by gathering information for the requirements of the project. In this stage, the business opportunities are identified, and planning occurs for the project, based on the collected data. After the requirement stage, the design stage takes place; in this stage, the information is gathered, and the

stakeholders identify requirements. Once the design stage is completed, the team is ready to start the work. The teams, which include designers and developers, start working on the project. Their objective is to organize a functional product. The product has to go through different improvement processes, which involve simple minimal functionality.

In the testing stage, the product is examined by the team, and the team seeks bugs at work. Once testing is completed, deployment occurs; in the deployment stage, the team launches a product for the user's environment. After the product is being launched, this stage takes place. In this stage, the team receives feedback about the product performance.

C. Types of Agile Frameworks

The Agile method's famous choices include (RUP) rational unified process, Scrum, Extreme programming (XP), and Adaptive software development.

Scrum is an AGILE model, used as a development software. It is perhaps the most popular agile model. It emphasizes on task management in team-based development conditions. The basic concept of this model is that developers and teams should work together. In simple words, the philosophy of this model is teamwork. Scrum uses the iterative approach to develop software that focuses on the team; thus, experienced employees on small teams proved to be the most successful, due to self-management and self-organization [26].

In this method, the team breaks the long-term goals into short term goals and executes them by applying fixed-length iterations or sprints to develop software. These sprints or iterations last for two to four weeks, and they are led with planning and previous sprint assessment. In Scrum, no modifications are allowed after the sprint activities are defined. The meeting is a crucial part of Scrum; in each sprint, sessions take place to examine the progress and get feedback. This method endorses modifications and development, which increase the values of complex projects. Scrum includes the structure of the commonly used software development with the iterative practices of modern AGILE [26].

Scrum is lightweight, which means Scrum consumes less time and increases productivity. It includes daily meetings and timely project deliveries. It helps the team to be more efficient and dynamic. It will lead the organization at the highest level of division. The hierarchy will be eradicated by Scrum. The authority and accountability of teams will be increased. This can increase customer satisfaction because of focusing on the customer. These are the reasons why Scrum is considered as the best Agile model.

RUP stands for a Rational Unified Process; It is also a software development process that combines linear and iterative structures. It is divided into four stages: Inception, Elaboration, Construction, and Transition [27]. Every stage is done in many iterations except Inception. All other development process stages like requirement, design, testing, etc. are done corresponding with these four stages of RUP. Although the intensity is different, it helps build stable and flexible solutions. However, this software is not the same as other Agile models like SCRUM and XP, it is not as efficient as other models of AGILE. Customer involvement, iterations and their intensity vary depending on the project requirements.

RUP development software delivers a structure for

companies to develop software programs. It gives them a plan for the development process and reduces the resources from being wasted and it increases productivity.

The process of RUP starts from the first stage of Inception. In this phase, the idea of the project is proposed. The team identifies the requirements of the project and determines whether to pursue the project or not. The second stage is Elaboration; in this stage, the team developers evaluate the architecture and requirements of the project. Developers also consider all the applications for the software and evaluate the cost for the development. In the third stage, which is Construction, the project is being developed and completed. On the other hand, the software is designed, written, and being tested. Finally, in the last stage, the software is launched for the public, final touches, and updates are being made based on users' responses [27].

RUP is being used in the banking sector incorporated with AUP (Unified Agile process). By this process, the project can be quickly achieved within the deadline and budget set. A bank in Greece introduced a medium-sized project named integrated desktop (I.D.) utilizing AUP. AUP is a big step in the I.T., and this can only be possible through RUP. Companies who found AUP to be productive must have the structure of the organization's RUP and AGILE approaches [27]. This proves the worthiness of RUP in the banking and finance sector.

XP or extreme programming is another Agile model. It is a software development model, and it targets to create quality software. It also concerns the life of the team. It is particularly popular in Agile models for suitable engineering practices for software development and it encourages modifications as it allows regular releases in short development sprints. The sprint lasts for one to two weeks [28]. This model enables the changes even after the iteration launch if the team is not working on a relevant software piece. That much flexibility makes things significantly complicated. It can impact things, especially the quality of the software. To resolve the problem, extreme programming uses other tools like pair programming, test-driven development, and test automation, continuous integration (CI), etc. [28]. It follows a set of values, unlike others. The values include simplicity, communication, consistent feedback, and respect. It has a unique feature like the small release, simple design of the software, and prescribed to follow coding standards, which makes it different from others. This model is particularly used in situations where clients are not sure of their requirements and they are continuously making changes in the project.

Extreme programming needs the developers to first understand the user's stories and describe a certain feature. Other than that, Extreme Programming includes practices like scheduling, and dividing work into iteration. Design with a relaxed and simple mind to be more creative, code, and test regularly to make the software safe and free from fault. Work on the feedback and upgrade according to the feedback to make sure the software meets the users' requirements. According to [29], there are 4 values of extreme programming:

- **Simplicity:** In Extreme Programming (XP), simplicity means the simplest way the work can be done. The reason to implement this is to minimize the wastage of resources and keep things simple. The design is kept as simple as

possible. Keeping the design simple allows the developers to maintain, support, and revise the design, ultimately leading to less wastage of the resources. In another way, it helps to deal with only the requirements and not focusing on anything else .

- **Communication:** Communication is a key feature in software development as it is highly dependent on teamwork, and thorough communication, the information can be transferred from one team member to the rest. Extreme programming emphasizes the proper communication: one on one discussion, whiteboard interaction, and other mechanisms.
- **Feedback:** Consistent feedback from the user is a key to success as it is known where to improve and where a revision is needed. By this process, it is easy to understand how users want the product to be. Feedback also promotes simple designs. When the development team launches something on the product, its design, implementation adjust according to the feedback.
- **Respect:** Respect is one of the values in this model because the team members need to respect each other to build communication. They provide feedback and accept feedback. Other than that, they should create harmony among them to build a relationship and keep it stronger.

As this model highly focuses on values related to collaboration, many European banks are adopting extreme programming to improve the delivery time and the quality of the projects.

D. The Advantages of Using Agile

The Agile method is very innovative compared to other models. This leads a company to innovation and to consistent work on the product results in developing the latest and effective solution that meets the customer needs and wants. For example, in the case of the increase in the use of smartphones, nowadays, people barely step into the branch for the transaction; they just use their phones. The 2018 U.S. Fintech market report described that nearly half of the respondents used mobile devices at a retail store [30]. Almost 48 percent of people use apps to pay their bills. Financial institutes using the Agile approach can benefit from changing the trends quicker and offer faster services, which will ultimately increase the customers. The Agile method proved to be very useful in challenging situations as the Agile development team can work in the cycle to find an innovative and productive solution, which can meet the new demands of the customer, quicker than other traditional software.

Agile method replicates the startup culture because it needs more than one Scrum team to work together to develop the product. This boosts creativity and functionality of team members, gaining a vital perspective from each other to increase innovation. On the other hand, traditional software has old ways of working, which focuses on the hierarchy system, which does not promote teamwork. Fintechs that use Agile methods have dedicated teams working for the development, research, product growth, etc., they collectively work to create innovative solutions and offer products with a short amount of time to launch. This allows the teams to work together, (otherwise they probably would not have collaborated), offering a new perspective to develop the product.

The Agile method considers an evolving industry like finance, being ready for the new changes and flexibility in the priority. There are several adjusting changes taking place worldwide, which means the old experienced companies and new entrants both should renovate and rework their work and services as needed. The Agile method reduces the time between new products and services demanded by customers, eliminating the reason to turn to a competitor. The Agile method gives the leverage of continuous delivery, which offers new innovative services for its consumers or upgrades the existing offerings to meet the dynamic changes in the market. The financial sector is one of the last sectors in the way of digitalization. Still, with many fintech startups across the world, giving the consumer easier and quick ways to manage their finances, this industry is soon going to witness significant changes in the future years. One of the many advantages where the Agile model comes into work is testing prototypes quicker.

E. The Downsides of Agile

Agile methods offer less predictability to developers. In order to deliver quality software, developers cannot measure the total amount of effort required to produce the product. This is an important factor at the start of the product development life cycle with regards to larger products. Teams which are new to the use of agile methodology fear this uncertainty. This causes work frustration, poor working practices, and often poor decision making among team members. However, on the other hand, the well-organized waterfall process makes it easy to measure the effort required, time needed to make the product, and cost of delivering the final software which is more efficient and reliable.

Furthermore, with agile methods it requires more time and energy of everyone involved. There is a need to have consistent involvement of testers, developers and customers and they must constantly interact with each other. This requires frequent direct conversations, as they are the best form of communication and all teams should be involved and must have good cooperation. If they don't cooperate with each other then quality products cannot be produced without their input and suggestion. This method requires daily users that should be available for on-time testing and sign off on each stage of software development so developers can assign them as completed before they can focus on creating new features. This method ensures that the final product meets user expectations and demands, but its drawback lies in a process which is difficult and time-consuming for an organization or company [31].

Agile method requires greater demands on product developers and customers. Although it is an interactive, engaging and rewarding system for the company, it stresses the need for a big commitment for everyone involved for the entire project to ensure organizational success. The drawback of using this method is that if clients are not skilled and have no knowledge about software, they must undergo training to support in product development. If clients fail to participate in development, it will impact final software quality and success. It also will also negatively affect the company's reputation and image on the development company.

Agile method also lacks in providing necessary documentation for a company. Documentation is less detailed

and not clarified for the company and this can lead to the company not being successful. This means that when new members join the team, they do not know the details about certain features or products or how they need to be performed. This builds confusion, misunderstandings, and difficulties among team members since they are not sure the purpose of documentation and the proposed audience.

Moreover, this method may cause a project to fall apart from achieving its success. This could happen when there is less customer involvement in regards to feedback and communication from them, resulting in a developer might pay attention on the wrong side of development due to this lacking in features [31].

F. Adaptation of Agile Methods in an Organization

The adaptation of agile methods is defined as a process of modifying agile methods to align them with the needs of different projects and organizational environments which means customer needs must be addressed beforehand applying chances to the software being developed. These methods are customized according to the specific needs of the client by changing the agile practices with the help of regular involvement of a cross-functional department within an organization. For example, if a client needs a banking app that specifically records their banking transaction, then a company should develop a software that satisfies this need. According to [32], adaptation of method enables the bank to obtain relevant, real-time data and response which enables them to stay ahead in regards to satisfying customer needs.

The possible reason for adaptation includes product quality, efficiency and effectiveness. To succeed in adaptation, an organization team must be a genuine working group that has their time and space devoted towards change, instead of being someone who meets once in a while. This means there should be regular involvement of team members to bring about change in the methods. Moreover, different members of the organization should be embodied those who have significant power and influence and have energy and ambition to adapt. A company can also improve the flow of communication across team members. In this way developers are authorized from the start of a project so they can speak directly with the customer or workmate; and better understand the requirement of the project that needs to be done. In the end, they will have the best knowledge to use to get the work done so they should be in close communication with those demanding the work [33].

Furthermore, banks could foster adaptation by focusing more on customers. They should collaborate and connect with customers and build trust among them as part of defining their strategic needs. It helps customers to connect with banks that clearly communicate their brand values with devoted commitments, while they do not go after those companies which customers feel as inauthentic. Banks should also embrace design thinking which means they should apply product design prior to design architecture for customer products. Customer observation is another way to adapt in an organization that can help to understand their specific needs, which enables the bank to replace the old product process with new improved customer-oriented features designed specifically for their needs.

G. Case Studies

Customers are attracted to products which offer accessibility and innovation. Mobile customers and mobile payments are growing, and people are choosing digital banking over other steps: digital banks like Monzo, N26, Atom, and Revolt [33]. These banks offer innovative services, which are far from the services provided by street banks. In a 2018 Capgemini report, it was reported that only 38 percent of the respondents would praise traditional software compared to the 55 percent respondent who would recommend fintech providers. The old players in the banking and finance sector need to adopt the Agile method as it helps the firms to meet the new and complex demands of the customers with innovative solutions. Seventy percent of the people said Agile methods and models take less time to complete the projects. The Agile approach permits smooth transitions and allows big companies to compete with the digital firms at the same level playing field.

JP Morgan launched a website, which was developed with the help of the Agile method. Gavin Michael, who is the head of digital for consumers and community banking, gave credit to the Agile method for the development and success of the process. The model proved to be successful in the banking industry because banks are developing technology, which needs to quickly adjust the project's changes according to customers [35]. The secret of Agile's success is that the design cycle is much smaller, and usually, the developers can quickly get the input from real users.

In 2014, a bank DBS in Singapore asked its employees to think like a fintech organization and prepare the digital capabilities. They come with a solution that does not waste time figuring out digital capabilities. This strategy worked perfectly for them. The net profit of the bank was 2.7 percent from 2014 when the transformation started in 2017. After that, in the next nine months, the profits showed a significant boost, from 2.7 percent to almost 36 percent. This idea was given by the bank CEO who spent many years in CitiBank in Asia before becoming the CEO of this bank. While other big firms and banks in Europe and America failed to find the solution to this problem, a bank from Asia with only 394 billion dollar assets (as in 2018) saw the appropriate solution [36].

H. Importance of Agile in Software Testing

Any financial error is always the critical one. If finances are involved in any application, guidelines have to be followed strictly. Both testers and developers of financial applications should have a good understanding of finance and complete domain knowledge as well [37].

According to [38], testing is a critical element in ensuring business delivery and it is important to mitigate the potential risks. One of the most important steps is to make a test plan [37]. The most important testing areas for a banking application are [38]: Continuous Testing; Omni-channel testing; Customer experience– testing; Analytics validations; Cyber Security QA. The banking applications require an End toEnd Testing methodology involving multiple Software Testing techniques to ensure [39]: total coverage of all banking workflows and Business Requirements; the functional aspect of the application; the security aspect of the application; Data Integrity; Concurrency; User Experience.

The typical stages involved in testing Banking Applications

are [39]: Requirement Gathering; Requirement Review; Business Scenario Preparations; Functional Testing; Database Testing; Security Testing; User Acceptance Testing. According to [37], the development or testing of any Banking Applications (High Performing, Reliable, Secure and Functional) is not teamwork but multi-functional teamwork in a healthy cooperated agile and technical environment. To help Testing, Development, Support team and the management to make the right decisions at the right time, good testing psychology, great communication skills and functional testing skills are a must for the software testers. In summary, the efficiency of the financial applications can be improved with proactive testing and risk management.

I. Importance of Agile in Assuring the Stakeholders Satisfaction

According to [37], banking applications are some of the most complex applications in today's software development and testing industry. The banking world is highly impacted by the constant changes brought in by the government in the form of banking regulations. The majority of Banking Projects are using: Agile/Scrum, RUP, and Continuous Integration methodologies, and Tools packages like Microsoft's VSTS and Rational Tools.

There are many points of contact with different interest groups when implementing a digitization project in the banking sector. All of them ultimately decide whether a project will be implemented at all, what scope it will have and how successful it will be in the long term. Therefore, it must be the aim of those responsible for the project to know the stakeholders and their interests at an early stage and to consider them in the various phases of implementation. It is essential for every digitization project in banks to identify the stakeholders, determine their needs and take them into account at an early stage in the digital solution approach [40].

Some fundamental stakeholders can be found in almost every project of digital transformation and they are represented in Fig. 1: customers, clerks, investors, audit, risk managers, IT team.



Fig 1. Fundamental stakeholders in banking software applications and in other financial software applications [40].

According to [40], today, the customer's expectations go beyond things that have become a matter of course, such as mobile user interfaces or 24/7 availability of information, even in the financial sector. Also, the transparency of processes, a good integration, the comparability of offers are important topics for the customers. The interests of the clerk are not very different from the customer's interests. By linking information across applications, the clerk wants to save time and avoid errors [40]. The investors, on the other hand, are particularly interested in how the digitization project will affect the profitability of the company. The questions about the expected increase in efficiency, the planned project period and the personnel required are obligatory [40]. Regarding the audit, as a guardian of regularity, it has the function of examining facts independently and objectively [40] and what is important for it are non-manipulable databases and reliable archiving. A reliable database as a basis for reliable results and an appropriate risk assessment is also of great importance for this interest group.

The success of the implementation team, another stakeholders group of a software financial project, is fundamental for a successful project. Whether they are credit experts, IT department heads, digitalization officers or clerks - all these people know the processes and legacy systems in detail and they also know their weak points. The fact that people who later work with the system are involved in the planning also increases acceptance and understanding of the system [41].

According to [37], Peer Business Analysts, Development leads, QA Engineers go through all the requirements so that neither new workflows nor existing business workflow is violated. They all make sure that the scenarios are implemented in such a way that all requirements from client side are covered. Business scenarios should include the main flow of the application. High-level scenarios are also called business scenarios. The Business Analyst reviews these Business Scenarios to make it sure that all of the Business Requirements are fulfilled.

IV. DISCUSSION

Nowadays, most of the firms are focusing on delivering a quality product to make their customers satisfied. For that purpose, they are stuck in traditional software. The world is digitizing, and it was becoming impossible to compete in the market with old standard software, especially with fintech organizations. Gradually, banks and other companies have started to orient themselves towards Agile approaches, and there is a lot of reason behind that. Traditional software, like the Waterfall methods, have been intensively compared to the Agile methodology. They have many negatives like testing can only begin after the process is completed, in a testing stage, you can't go back and edit anything, so save cost plus time. The projects get scrapped, costing expenses and time. The way to survive in the industry is to adopt the Agile method [22]. There are many advantages to the Agile approach.

The Agile method saves cost and time as its implementation is intended to reduce the danger, uncertainty, and cost. The Agile methodology highly focuses on consistent feedback in the development of all the participants. The Agile

model allows the increase of team chemistry and team efficiency, which results in the team's productivity.

The Agile model provides benefits to the employees on all development processes, compared to other models. In an Agile method, there are models like Extreme Programming, which focuses on values and teamwork. This is the reason the Agile model is more productive for employees.

The Agile method provides business value, flexibility, adaptability. Project managers cannot predict the future; however, they can affect how quickly a team can adapt. One of the main features of the Agile method is the willingness to adjust to the transition. Since the project managers cannot see the future but can affect how the team reacts to the changes, the ability to adapt to the changes is a key feature of Agile project management.

The Agile model is very good at meeting the demands of the evolving sector. Normally, banks made a habit of using the waterfall model. This model perfectly meets the requirements; however, it lacks to meet the altered demand during the implementation phase of the project. It also lacks predicting the schedules. On the other hand, the Agile is particularly famous for regular significant changes. It meets all the strict regulatory requirements.

The Agile method helps to gain customer satisfaction, which is considered the most important thing for any company. Clients want maximum accuracy in their work. Agile professionals get quickly involved in their practices. Good customer relationship management (CRM) is very important to do. The factor of teamwork in the Agile model helps the developer to create worthiness for the business. This will open the door for a long-term stream of income.

Agile uses constant integration and consistent deployment. This means that all the people working on the project share their work on the central system allowing mistakes to be caught easily.

This saves time and saves the team from redoing their work. This also increases the (ROI) return on investment, as products and services are launched quicker.

REFERENCES

- [1] Rose, K. H. (2013). A guide to the project management body of knowledge (PMBOK® Guide)—Fifth Edition. *Project management journal*, 3(44), e1-e1.
- [2] Tarhan, A., & Demirors, O. (2011). Apply quantitative management now. *IEEE software*, 29(3), 77-85.
- [3] Uskarci, A., & Demirors, O. (2017). Do staged maturity models result in organization-wide continuous process improvement? *Insight from employees. Computer Standards & Interfaces*, 52, 25-40.
- [4] Tannöver, Ö. Ö., & Demirors, O. (2015). A process capability based assessment model for software workforce in emergent software organizations. *Computer Standards & Interfaces*, 37, 29-40.
- [5] Sulayman M, Mendes E. An extended systematic review of software process improvement in small and medium web companies. In 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011) 2011 Apr 11 (pp. 134-143). IET.
- [6] Cass AG, Sutton SM, Osterweil LJ. Formalizing rework in software processes. In *European Workshop on Software Process Technology 2003 Sep 1* (pp. 16-31). Springer, Berlin, Heidelberg.
- [7] Boehm BW, Papaccio PN. Understanding and controlling software costs. *IEEE transactions on software engineering*. 1988 Oct;14(10):1462-77.
- [8] Twentyman J. The crippling costs of IT project rework. *Inside Knowledge*. 2005 Jun 15;15.
- [9] Charette RN. Why software fails [software failure]. *IEEE spectrum*. 2005 Sep 6;42(9):42-9.

- [10] Davis AM, Bersoff EH, Comer ER. A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering*. 1988 Oct;14(10):1453-61.
- [11] von Prince K. Paradigm-induced implicatures of TAM markers: The case of the Daakaka distal. In *Proceedings of Sinn und Bedeutung 2018* (Vol. 21, No. 2, pp. 969-984).
- [12] Chen, Z., Li, Y., Wu, Y. and Luo, J. (2017), "The transition from traditional banking to mobile internet finance: an organizational innovation perspective - a comparative study of Citibank and ICBC", *Financial Innovation*, Vol. 3 No. 1, p. 12.
- [13] Lee, J.-C. and Chen, C.-Y. (2019), "Investigating the environmental antecedents of organizations' intention to adopt agile software development", *Journal of Enterprise Information Management*, Vol. ahead-of-print, available at: <https://doi.org/10.1108/JEIM-06-2018-0119>.
- [14] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J., 2001. Manifesto for agile software development.
- [15] Vilela J, Castro J, Martins LE, Gorschek T. Integration between requirements engineering and safety analysis: A systematic literature review. *Journal of Systems and Software*. 2017 Mar 1;125:68-92.
- [16] Bowen S, Maurer F. Process support and knowledge management for virtual teams doing agile software development. In *Proceedings 26th Annual International Computer Software and Applications 2002 Aug 26* (pp. 1118-1120). IEEE.
- [17] Holz H, Maurer F. Knowledge management support for distributed agile software processes. In *International Workshop on Learning Software Organizations 2002 Aug 6* (pp. 60-80). Springer, Berlin, Heidelberg.
- [18] Shameem M, Kumar RR, Nadeem M, Khan AA. Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. *Applied Soft Computing*. 2020 May 1;90:106122.
- [19] Awad, M.A. (2005), "A Comparison between Agile and Traditional Software Development Methodologies, This report is submitted as partial fulfillment of the requirements for the Honours", The University of Western Australia.
- [20] Kumar, G. and Bhatia, P. (2014), "Comparative Analysis of Software Engineering Models from Traditional to Modern Methodologies", presented at the International Conference on Advanced Computing and Communication Technologies, ACCT, available at: <https://doi.org/10.1109/ACCT.2014.73>.
- [21] Liang, J. and Shekhar, S. (2018), "AGILE ADOPTION IN INVESTMENT BANKS".
- [22] Vinekar, V. and Huntley, C.L. (2010), "Agility versus Maturity: Is There Really a Trade-Off?", *Computer*, Vol. 43 No. 5, pp. 87–89.
- [23] Flora, H. (2014), "A Systematic Study on Agile Software Development Methodologies and Practices", *International Journal of Computer Science and Information Technologies*, Vol. 5, pp. 3626–3637.
- [24] Javanmard M, Alian M. Comparison between Agile and Traditional software development methodologies. *Fen Bilimleri Dergisi (CFD)*. 2015;36(3).
- [25] De Carvalho, B. V. and Mello, C. H. P. (2017). "Scrum agile product development method-literature review, analysis and classification". *Product: Management and Development*, 9(1), 39-49.
- [26] McManus, J. (2004), "Managing Stakeholders in Software Development Projects", Elsevier Butterworth-Heinemann, ISBN: 9780750664554.
- [27] Jureczko, M. (2008), "The Level of Agility in Testing Process in a Large Scale Financial Software Project".
- [28] Newkirk, J. (2002), "Introduction to agile processes and extreme programming", *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002* (pp. 695-696). IEEE.
- [29] SPGlobal (2020), 2018 US Fintech Market Report, SP Global, available at: <https://www.spglobal.com/marketintelligence/en/documents/2018-us-fintech-market-report.pdf> (accessed on August 2020).
- [30] McCarthy S, O'Raghallaigh P, Fitzgerald C, Adam F. Towards a framework for shared understanding and shared commitment in agile distributed ISD project teams. In *ECIS 2019, Proceedings of the 27th European Conference on Information Systems 2019* (pp. 1-15). AIS Electronic Library (AISeL).
- [31] Roses, L.K., Windmüller, A., Carmo, E.A. do, Roses, L.K., Windmüller, A. and Carmo, E.A. do. (2016), "Favorability Conditions in the Adoption of Agile Method Practices for Software Development in a Public Banking", *JISTEM - Journal of Information Systems and Technology Management*, Vol. 13 No. 3, pp. 439–458.
- [32] Newton N, Anslow C, Drechsler A. Information Security in Agile Software Development Projects: a Critical Success Factor Perspective.
- [33] Mbama, C. I., & Ezepue, P. O. (2018). "Digital banking, customer experience and bank financial performance". *International Journal of Bank Marketing*.
- [34] Crosman, P. (2016), "Agile Development Is Reshaping Tech at Banks Like Chase and BBVA", *American Banker*, available at: <https://www.americanbanker.com/news/agile-development-is-reshaping-tech-at-banks-like-chase-and-bbva> (accessed on August 2020).
- [35] Sia, S. K., Soh, C., & Weill, P. (2016). "How DBS Bank Pursued a Digital Business Strategy". *MIS Quarterly Executive*, 15(2).
- [36] Sharma, A. (2020), Why Software Testing Is Very Important For Banking Applications?, available at: <https://www.bugraptors.com/importance-of-qa-for-banking-applications/>.
- [37] Chintala, S. and Mohan, N. (2020), "The intricacies of software testing", available at: <https://bankingfrontiers.com/the-intricacies-of-software-testing/> (accessed 12 April 2020).
- [38] STH (2019), "The Implementation of a Banking Software for the Lending Sector in Practice", available at: <https://www.knowis.com/blog/implementation-of-a-banking-software-for-the-lending-sector-in-practice> (accessed 23 January 2020).
- [39] Murmann, S. (2018), "The Most Important Stakeholders in the Digital Transformation of Finance", available at: <https://www.knowis.com/blog/the-most-important-stakeholders-in-the-digital-transformation-of-finance> (accessed 12 April 2020).
- [40] Sternkopf, C. (2018), Software Testing Help, "The Implementation of A Banking Software for the Lending Sector in Practice", available at: <https://www.knowis.com/blog/implementation-of-a-banking-software-for-the-lending-sector-in-practice> (accessed in August 2020).