

ADVANTAGES AND DISADVANTAGES OF TRADITIONAL AND AGILE METHODS IN SOFTWARE DEVELOPMENT PROJECTS – CASE STUDY

Adam GUMIŃSKI^{1*}, Katarzyna DOHN², Enioluwa OLOYEDE³

¹ Silesian University of Technology; adam.guminski@polsl.pl, ORCID: 0000-0001-8250-7101

² Silesian University of Technology; katarzyna.dohn@polsl.pl, ORCID: 0000-0002-4178-1347

³ Silesian University of Technology; eo300167@student.polsl.pl, ORCID: 0009-0000-2002-3291

* Correspondence author

Purpose: The article deals with the comparison of traditional and agile software development methodologies to evaluate their effectiveness in project management. The study aims to identify the challenges of communication and trust that occur in project management and how these challenges can be addressed. The literature review discusses various traditional and agile software development methodologies, their limitations, and the importance of project management, communication, and trust.

Design/methodology/approach: The study compares the effectiveness of agile project management and waterfall project management Software Development Life Cycle (SDLC). The objectives of the research were achieved by a questionnaire survey and interviews with software development teams. The results highlight the challenges of communication and trust in project management, and the study concludes that agile methodologies are more effective in project management as they provide more opportunities for communication and collaboration among team members.

Findings: The study recommends that teams working on software development should implement agile approaches to improve their practices regarding project management. Agile approaches offer an increased number of chances for communication, cooperation, and feedback, all of which can contribute to the successful completion of a project. In addition, the findings of the study suggest that future research should center on the development of solutions to address the issues of communication and trust in project management.

Research limitations/implications: The research limitations are due to fewer sample size of the data and the study is based on self-reported data which may not be entirely accurate. The study also has a limited scope, and the results may not be generalizable to other industries or project types.

Practical implications: The research proved that agile methods are challenging in IT project teams. Still, communication and trust in project management are of great importance to ensure effective development of IT projects.

Originality/value: The value of the research is the identification of the challenges of communication and trust that occur in project management and how these challenges can be addressed. The recommendations elaborated within the study to support the effectiveness of projects' development are dedicated to companies in an IT sector.

Keywords: Scrum, Waterfall, software, agile, project management.

Category of the paper: research paper, case study.

1. Introduction

In recent times, there have been notable changes in the software development industry, leading organizations to re-evaluate their approaches to project management. In the domain of software development project management, two prominent methodologies, namely traditional and agile, have surfaced as the primary contenders. In the pursuit of delivering software of superior quality within the constraints of budget and time, it is crucial to comprehend the inherent pros and cons of each methodology.

The article offers a comprehensive case study that analyses the complexities of conventional and agile approaches in software development initiatives. The objective is to offer a comprehension of advantages and constraints linked to each approach by scrutinizing practical situations and consequences. The aforementioned assessment is aimed at the support of software developers, project managers, and stakeholders in arriving at well-informed decisions that are in accordance with the distinctive needs of their respective projects.

This case study examines the attributes of the conventional waterfall approach and the agile methodology, analysing their effects on project planning, implementation, flexibility, client contentment, and overall project achievement. The conventional waterfall methodology adheres to a sequential and linear approach, while the agile methodology prioritises iterative development and ongoing collaboration.

Through a critical evaluation of the benefits and drawbacks of each methodology, the paper is aimed at explaining their impact on key project variables such as project schedules, cost-benefit ratios, risk mitigation, and stakeholders' involvement. By conducting a comprehensive examination, the aim is to furnish practical observations obtained from practical encounters, endowing readers with valuable expertise to proficiently traverse the software development terrain.

The paper provides a comprehensive case study that analyses the merits and demerits of conventional and agile approaches in software development endeavours, ultimately leading to a persuasive conclusion. Through an examination of various methodologies in practical situations, the paper is aimed at equipping stakeholders with the requisite information to make well-informed choices. The capacity to implement a suitable project management methodology can significantly impact the attainment of favourable software development outcomes in a constantly changing industry.

2. The characteristics of differences and similarities of waterfall and agile methodology in an IT sector

2.1. Traditional/Waterfall Model

The traditional model, commonly called the waterfall model, is a static paradigm that takes a linear, sequential approach to systems development, finishing one task before beginning the next. The waterfall method divides projects into the following activities: requirement analysis, design, coding, and testing. According to Pressman, these include the following activities: communication (including project start-up and requirements gathering), planning (estimating, scheduling, and tracking), modelling (analysis and design), construction (code and testing), and deployment (delivery, support, and feedback) (Pressman, 2005). The phases of the model presented by Pfleeger and Atlee are requirement analysis, system design, program design, coding, unit and integration owner/user of the proposed system, and so forth (Pfleeger, Atlee, 2006).

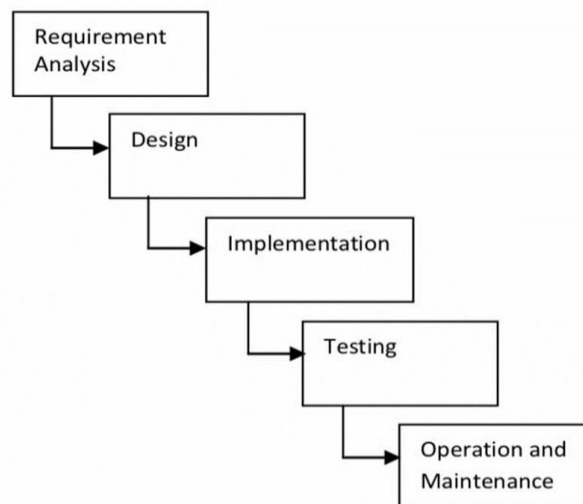


Figure 1. Waterfall Model Phases.

Source: Pfleeger, Atlee, 2006.

According to Othman et al., the waterfall model entails that one phase must be completed before the next can begin, with a high risk that errors from the previous phase will be transmitted to the next, as verification occurs at the end or near the end of software development.

Modifications cannot be made using a waterfall development process, which is a drawback of the conventional method. Furthermore, there is no overlap between stages in the typical waterfall approach to development. The model can support iteration, but it only does so indirectly (Pressman, 2005). A phase cannot be repeatedly reviewed to look for flaws after it has been completed. Since the phase cannot be reopened, no improvements are possible. This methodology is particularly helpful for developing structured systems because it is strictly forbidden to change the program once it has been coded. Additionally, the waterfall approach typically separates processes from data, so if data need to be changed, the code must also be

adjusted (known as software coupling). This renders software unusable and makes system upgrades difficult because all processes need to be changed in order to make any adjustments, which can be time-consuming and expensive.

There have been new improvements made to the waterfall model that are meant to address the shortcomings of the traditional waterfall model. The result of these adjustments is what McConnell calls "Modified Waterfalls," an improved version of Rapid Development models. This is according to research. Requirement analysis, design, implementation (or coding), testing, and maintenance are all still present in the updated version of the waterfall methodology. The overlap of phases makes the software engineering process adaptable. This ensures that any problems with the software system are addressed early on in the development process, saving money on post-implementation modifications. As a result, the modified waterfall model is now widely used in many industries, particularly manufacturing and construction, for both the management of information systems and the execution of projects.

The waterfall model has several advantages, including structuring the work, simplifying follow-up, clarifying objectives, improving communication, and traceability. The model also creates a clear and precise structure, enables smooth information transfer, simplifies project management, makes it easy to set goals, helps examine each stage's output, stabilizes a project, and prioritizes project timelines. However, the model has some disadvantages, such as inflexibility, limited customers' involvement, late feedback and testing, high risk of project failure, long delivery cycles, lack of visibility and control, limited room for experimentation, and difficulty in managing complex projects.

2.2. Agile Methods

Agile development is based on the concept of incremental and iterative development, in which the stages of a development life cycle are reviewed repeatedly. Software is enhanced iteratively by identifying solutions based on user feedback (Szalvay, 2004).

Agile methodologies require a shift from command-and-control management to leadership-and-collaboration. The organizational form that facilitates this shift needs the right blend of autonomy and cooperation to achieve the advantages of synergy while providing flexibility and responsiveness.

In agile development, the development life cycle is broken up into smaller portions that are referred to as "increments" or "iterations". Each of these "increments" or "iterations" touches on each of the conventional phases of development, as opposed to the implementation of a single massive process model that is used in conventional software development. The following four are among the four main agile factors, according to the Agile Manifesto:

1. Early customer participation
2. Iterative design
3. Self-organizing groups
4. Change adaptation

These six techniques - crystal approaches, dynamic software development method, feature-driven development, lean software development, scrum, and extreme programming - are currently recognized as agile development methods (Leau et al., 2012).

Some of the agile software development methodologies are: Scrum, Lean, Adaptive Software Development (ASD), Crystal and Extreme Programming (XP).

Agile methodology can be applied everywhere such as teaching, research and service.

Below, you find the list of agile practices:

- Product Backlog.
- Sprint Backlog.
- Daily Scrum.
- Scrum Master.
- Sprint Review.
- Sprint retrospective.

2.2.1. Scrum

Scrum is an agile methodology that uses iterative and incremental methods whose purpose is to help development teams to concentrate on established goals and minimize the work done on less important tasks. Scrum aims to keep simplicity in a complicated business environment.

The term comes from rugby, where it is a strategy to return a lost ball into the game by teamwork. Scrum does not provide implementation-level techniques; it focuses on the way the members a development team should interact to create a flexible, adaptive and productive system in a constantly changing environment. The method was presented in detail K. Schwaber and M. Beedle, Prentice Hall (2002). Scrum is based on two elements: team autonomy and adaptability. Team autonomy means that project leaders establish the tasks the team must perform, but in each iteration, the team is free to decide how to work, with the goal of increasing team productivity. To prevent the uncertainty caused by project complexity and unpredictability, Scrum does not suggest specific software development approaches, but rather strategies and instruments for the management for various phases.

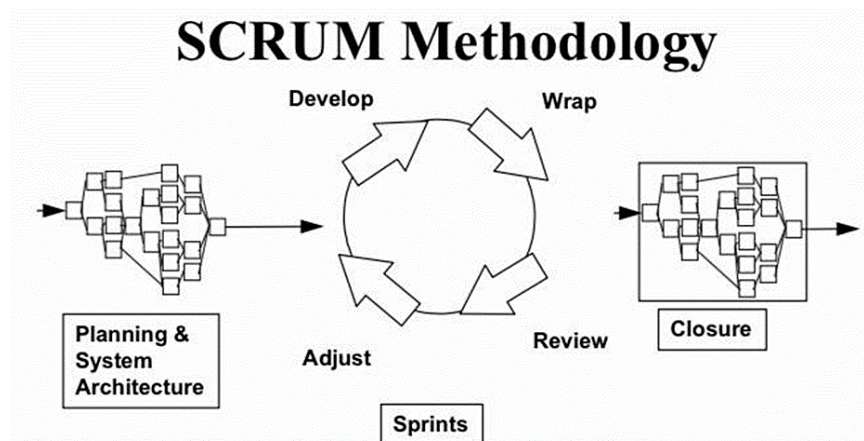


Figure 2. Scrum model.

Scrum is a very adaptable paradigm that can be used for any project in any industry. Both small and large projects can benefit from using it. The product owner, scrum team, and scrum master are all involved in scrum processes. It also includes sprints, the smallest time frame in which a team can assign tasks that must be completed in three to four weeks. The sprint backlog, which comprises all the requirements for the current sprint and any potential changes throughout development, determines the tasks for each sprint. The product backlog is regarded as the majority of requirements and is evaluated by the product owner. It is divided into sprints and then has a sprint planning structure that outlines various strategies for completing a sprint on schedule. Delivering a potentially organized and error-free product is the goal of each sprint (Abdulbaqi Badru, 2017).

2.2.2. Lean Model

Womack et al. (1990) introduced the lean concept based on many studies of the automotive industry conducted by researchers at the Massachusetts Institute of Technology (MIT) International Motor Vehicle Program (IMVP). This idea builds on TPDS and falls under the more general category of lean production. They concluded that there were differences between mass and Japanese lean producers in terms of strong leadership, teamwork, early communication and coordination across departments, and concurrent development procedures in addition to differences in the production processes. Lean manufacturing has been adopted by many businesses that create huge software-intensive systems, but in order to fully exploit this competitive advantage and squeeze out additional waste from the manufacturing processes, lean also needs to be applied to the PD processes, Liker, J.K., & Morgan, J.M (2006). Several businesses have begun to use some underlying LPD concepts and practices in order to improve the effectiveness of the PD processes. Continuous improvement (Kaizen), Kanban, concurrent engineering, inclusion of customers and suppliers, visual management, group work, and cross-functional teams are some of the techniques utilized to achieve the goal of LPD. However, implementing a small number of the techniques is insufficient to achieve LPD. According to Womack and Jones (2003), the entire enterprise needs to adopt the lean philosophy. Five categories are used to conceptualize lean thinking: value, value stream, flow,

pull, and perfection. Value establishes the purpose that a product serves for a consumer and builds business operations from there. Each process step is described in a value stream, which classifies them according to the value added (e.g., value adding, necessary non-value adding and non-value adding steps). Processes are organized by flow to ensure that products move easily through the value-generating steps. Each customer calls the output from the prior phase using the pull method. Finally, perfection requires continuous process improvement to fulfill client expectations and has no flaws.

Studies on TPDS are often cited in literature that explicitly addresses LPD (Kennedy, Widener, 2008). A thorough study on LPD is presented by Liker, J.K., & Morgan, J.M (2006). They define the core and substance of LPD in a model called Lean Product Development System using the Sociotechnical Systems Theory (STS) (e.g. Miller, Rice, 1967) and the principles and practices of TPDS (LPDS). The foundation of LPDS is the idea that LPD is a philosophy being adopted throughout the entire enterprise rather than a few lean principles and practices being superficially applied to different areas of an organization.

The three main subsystems of the LPDS paradigm are process, talent, tools and technology. 13 concepts are used to define them.

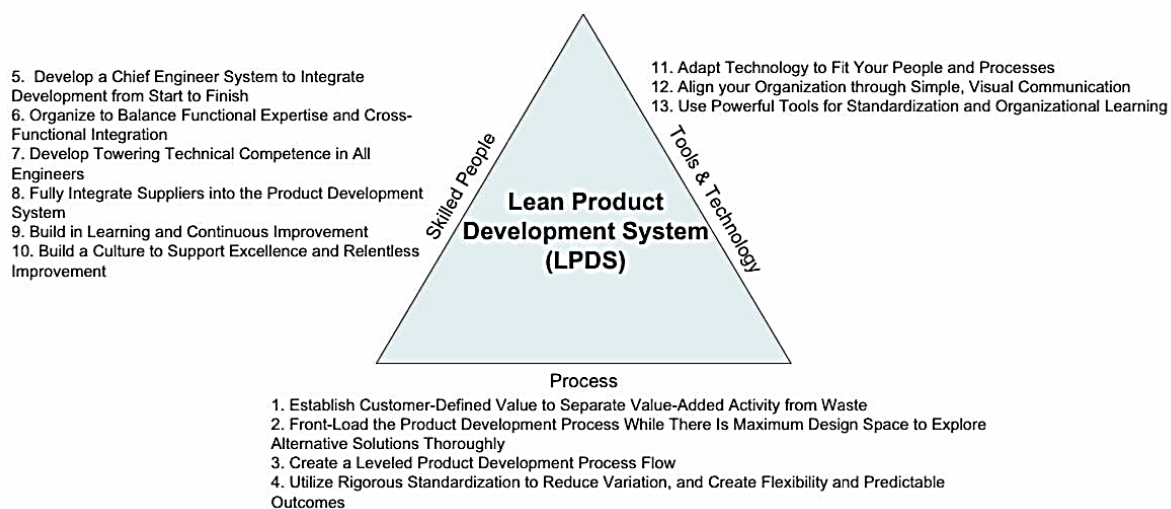


Figure 3. LPDS model.

Source: adapted from Liker, Morgan, 2006.

The advantages of the Agile approach include the following:

- It's adaptable and flexible.
- It encourages creative approaches.
- It involves low costs: Adopting the Scrum approach can be cost-effective for an organization, as it usually requires less documentation and control.
- It usually leads to better quality work.
- It improves customer satisfaction.
- It typically results in more satisfied employees.

Despite the advantages that can be gained from using agile approaches, many businesses are hesitant to completely abandon their standard practices. Their reluctance stems from a number of sources, including the following:

- the agile methods drastically cut down on documentation and heavily rely on tacit knowledge;
- the agile methods have not been adequately tested for mission/safety-critical projects;
- the belief that the agile methods are not adequate for highly stable projects;
- the belief that the agile methods can be successful only with talented individuals who favour many degrees of freedom;
- the belief that the agile methods can only be successful with talented individuals who favour many degrees.

2.3. The comparative analysis of agile versus traditional methodologies

A software program or information system is made to carry out a specific set of tasks. This group of jobs that the system will carry out frequently produces results that are well-defined and involve complicated computing and processing. Therefore, it is a difficult and time-consuming task to oversee the entire development process in order to guarantee that the final product has a high level of integrity and resilience, as well as user approval. To accomplish the aforementioned "characteristics of a successful system," a systematic development approach that may place an emphasis on comprehending the scope and complexity of the whole development process is therefore crucial.

The majority of system developers currently use either traditional development or agile development as their SDLC approaches. The following analyses the comparison between agile and traditional methodologies:

1. **Primary objectives:** According to Boehm, predictability, repeatability, and optimization have been a key set of goals for more traditional plan-driven techniques. While Agility is more concerned with delivering quick value and adapting to change (Fowler, Highsmith, 2001).
2. **Size:** Compared to Agile initiatives, plan-driven approaches scale to huge projects better. However, a bureaucratic, plan-driven organization won't be very effective on minor initiatives if it takes an average of one person-month merely to have a project authorized and begun (Boehm, 2002).
3. **Relationships with customers:** Agile approaches function best when customers work closely with the development team and when their tacit knowledge is adequate for the entire program. This approach runs the risk of tacit knowledge gaps, which plan-driven approaches minimize by using documentation, architecture review boards, and independent expert project evaluations to make up for on-site client gaps (Boehm, 2002).

4. **Planning and oversight:** Formal project management, according to Unhelkar, is crucial to the effective conclusion of a software project. Careful planning, estimation, coordination, tracking, and control are necessary for project management aspects that the Waterfall model officially covered. Agility gives the planning stage more weight than the final paperwork (Boehm, 2002).
5. **Communication:** While the Waterfall model prefers explicit documented information, Agility encourages face-to-face communication (Casteren, 2017).
6. **Requirements:** Pure Agile practitioners do not explicitly apply formal, upfront, solution-independent requirements engineering. The formal, cumbersome Waterfall model will struggle to keep up with the requirements' constant change. On the other side, plan-driven approaches can maintain even million-line systems under budget and schedule if the design foresees and accommodates required changes (Unhelkar, 2016).
7. **Development:** Agility maximizes the amount of work that is not done by valuing functioning software over thorough documentation. While the Waterfall model strongly relies on software architecture because it is a necessary step in the development process.
8. **Test:** Beznosov and Kruchten claim that traditional assurance procedures include assessment, vulnerability testing, internal and external reviews, dynamic testing, static analysis, and design and architectural principles. These techniques are significantly more suited to architecture-focused, well-documented waterfall development. On the other side, agile methodologies lack the emphasis on architecture or documentation but instead promote internal design and code review and encourage developers to follow coding standards.
9. **Clientele:** Agility requires devoted, nearby, knowledgeable clients. The Waterfall model needs customers who are sufficiently knowledgeable and skilled.
10. **Developers:** Agile, knowledgeable, collocated, collaborative, friendly, talented, skillful, and communicative developers are required in agile projects (Boehm, 2002).

3. The brief characteristics of the selected company's activities

The name of the analyzed company is Youverify, a RegTech100 company. Youverify is a Nigerian-based technology company that specializes in identity verification and background checks. The company's services include identity verification, address verification, document verification, facial recognition, and background checks. These services are designed to assist organizations/businesses and individuals in onboarding customers, validating their identities, minimizing the risks associated with fraudulent activities, and providing innovative solutions to ensure trust in online transactions.

Characteristics of the company's main activities are listed as follows:

1. **Know Your Customer (KYC) Solutions:** Youverify's solutions help businesses comply with KYC regulations by verifying the identities of their customers. They assist in preventing identity theft, fraud, and unauthorized access to services.
2. **Digital Onboarding:** Youverify offers solutions that facilitate smooth and secure digital onboarding processes for businesses. By automating identity verification, they help streamline the customer registration and account opening procedures.
3. **Fraud Prevention:** The company's services are designed to detect and prevent fraudulent activities. By analyzing data from multiple sources and conducting risk assessments, Youverify helps businesses identify suspicious behavior and mitigate the risk of fraud.
4. **Identity Verification:** Youverify specializes in providing identity verification services to businesses and individuals. They use advanced technologies like AI, machine learning, and computer vision to verify the authenticity of identification documents and validate the identity of individuals.
5. **Document Authentication:** Youverify utilizes optical character recognition (OCR) and document authenticity checks to validate the legitimacy of identification documents such as passports, driver's licenses, and national ID cards.
6. **Facial Recognition:** Their facial recognition technology enables businesses to compare live images or videos of individuals with their photos on government-issued ID documents. This helps confirm the person's identity during various online interactions.
7. **Risk Assessment:** Youverify performs risk assessments by analyzing data from different sources, including government databases and credit bureaus. This helps businesses evaluate the risk associated with individuals or transactions and make informed decisions.
8. **Industry Applications:** Youverify's services find applications in various industries such as banking, finance, e-commerce, ride-hailing, telecommunications, and online marketplaces. Their solutions help businesses enhance security, improve compliance, and build trust with their customers.

By leveraging advanced technologies such as artificial intelligence, machine learning, and data analytics, Youverify aims to enhance security and enable seamless user experiences in various industries, including financial services, e-commerce, sharing economy platforms, and recruitment.

The company has 32 employees who work in the engineering, sales, marketing, design, and legal departments. Key values of the company include customer focus, clear communication, transparency, and integrity. Notable clients of the analyzed company include Fidelity Bank, Wema Bank, Bolt, and Leadway Assurance.

4. The research analysis on applied project management methods in an analysed company

4.1. The form and source of data collection

The current study aimed to compare traditional (waterfall) and agile (scrum) methods of software development. The aim was to identify how these methods are used in software development and their performance. The respondents were asked the size of their project teams, the duration of past projects and other key questions related to the study.

The form was sent to 20 respondents to find out how familiar they were with the methods shown. A set of questions was chosen to see how the candidates used the agile method (Scrum) and traditional method (Waterfall) in different situations.

The study was about people in the field of software development. 90 percent of the participants had at least one year of work experience and 45 percent had more than four years of work experience while the most popular method used by the respondents is Scrum, with 70 percent of them using Scrum very well and 40 percent of the respondents are software engineers, 15 percent are quality assurance engineers, and the rest are either business analysts, project managers, product designers, or scrum experts.

4.2. Communication challenges in project management

In a survey conducted by Demir, 2007, communication was on the list, which isn't usually covered in other surveys. Smaller project groups were thought to have no trouble communicating. But the survey results show that there is almost no difference between project organizations with 1-10 people and those with 11-100 people in terms of ratios. Also, the results show that communication is hard no matter what kind of organization you work for. Compared to other types of organizations, it is a little bit higher in those that work with the government. Half of the projects in these groups had trouble with communication. It was a little lower than the other categories in small application projects (Demir, 2007).

In another survey conducted by Alzoubi and Gill, 2014, it was discovered that not enough attention has been paid to communication particularly with the customers. Only 14% of the studies that were chosen have talked about the challenge of communication (Alzoubi, Gill, 2014). In agile development, customers have to be involved in the process, and they can't be kept in the dark about project details (Korkala et al., 2010).

The final questionnaire consisted of four main sections: the technical background of respondents, new product development with scrum, new product development with the waterfall model, and their personal opinions. Questions were developed using 5-point Likert scales. In order to examine the use of software development methodologies, the Likert scale ranged from "never" to "always" and for performance, the range was from "much worse" to "significantly better".

In this current study, only 20 percent of the respondents mentioned customer satisfaction as an advantage of agile method (scrum). This relatively low percentage could be because of poor communication between the other teams and their customers.

4.3. Trust Challenges in Project Management

In a study by McHugh et al., (2012), it was found out that agile methods increased trust by making things clearer, accountable, communicative, open to feedback, and sharing of knowledge. Using agile practices made their projects more open and clearer, both within the team and throughout the organization. For example, during the iteration/sprint planning meeting, team members can see what tasks each person is responsible for and how much time they will take. "Everyone hears the news at the same time, not from someone else" (McHugh et al., 2012).

This is in consistency with the results from the survey. 30 percent of the respondents mentioned teamwork as an advantage of working with scrum method. This relatively high percentage has resulted because agile methods including scrum have been able to deal with the challenges of trust in project management.

4.4. Results Achieved

When respondents were asked about the advantages of the waterfall method, 25 percent of them said it was easier to use and probably the only advantage they know. In fact, another 25 percent of the respondents have never worked with the waterfall method. Out of the 8 software engineers among the correspondents, only 1 software engineer said Waterfall was easy to use.

It was discovered that agile software development life cycle processes are superior to the waterfall model. However, it is essential for the development team to select the SDLC that would work in the most effective manner for the project.

The results of the questionnaire survey and interviews highlight the challenges of communication and trust in project management. The study concludes that agile methodologies are more effective in project management as they provide more opportunities for communication and collaboration among team members.

4.5. Recommendations dedicated to analysed IT company

It was discovered that agile software development life cycle processes are superior to traditional SDLC. However agile software development life cycle also has a few drawbacks. So, it is essential for the development team to select the SDLC that would work in the most effective manner for the project.

The size of the development team, where it is located, the size and complexity of the software, the type of project, the business strategy, the engineering skills of the team, and any other relevant factors are some of the things that could be used by the team to determine what SDLC they want to use. Other factors may also be taken into consideration as necessary. It is essential for the team to take into consideration the distinctions between each SDLC, as well as its advantages and disadvantages, before settling on one. The team needs to conduct additional research on the company context, the requirements of the industry, and the business strategy before they will be able to compare the candidate SDLC to the selection criteria.

The Agile approaches offer an increased number of chances for communication, cooperation, and feedback, all of which can contribute to the successful completion of a project.

The study recommends that teams working on software development should implement agile approaches in order to improve their practices regarding project management and achieve higher effectiveness level of projects' development.

5. Conclusions

1. After analysing the results of the survey, it is clear that Scrum, a type of agile methodology, is becoming more and more popular in software development. People in professional organizations prefer to use Scrum, for different reasons. It was found that 95 percent of the people who took part in the study use agile methods, and about 74 percent of them use Scrum.
2. 30 percent of the respondents said that working with the scrum method made it easier to work as a team. This relatively high percentage is because agile methods like scrum have been able to deal with the problems of trust in project management.
3. Scrum also has a high success rate because the process of testing is done all the way through the product's lifecycle. It lets the product be constantly checked as it is being made. Agile also thinks that each member of the product team should do more and be more involved, so that people are not tied to their job descriptions. The team is also given the power to make decisions, and most of them have a clear idea of what they will do in their next sprint. Agile gives the opportunity to change requirements when it is needed.
4. When it comes to relationships with customers, only 25 percent of the respondents agreed that waterfall does a better job. Meanwhile, 75 percent agreed that scrum does better, and 40 percent even agreed that scrum does much better.
5. Agile methods increased trust by making things clearer, accountable, communicative, open to feedback, and sharing of knowledge. Using agile practices made their projects more open and clearer, both within the team and throughout the organization.

6. 30 percent of the respondents mentioned teamwork as an advantage of working with scrum method. This relatively high percentage has resulted because agile methods including scrum have been able to deal with the challenges of trust in project management.
7. The research has identified the challenges of communication and trust in project management and how these challenges can be addressed. Through a questionnaire survey and interviews with software development teams, the study has demonstrated that agile methodologies are more effective in project management as they provide more opportunities for communication and collaboration among team members.

References

1. Abdulbaqi Badru et al. (2017). *Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature*.
2. Alzoubi, Gill (2014). *Agile global software development communication challenges: A systematic review*.
3. Boehm, B. (2002). Get Ready for Agile Methods, with Care. *Computer*, 35, 64-69.
4. Casteren, W.V. (2017). *The Waterfall Model and the Agile Methodologies: A comparison by project characteristics*.
5. Cho, J.J. (2010). *An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum*. All Graduate Theses and Dissertations, 599.
6. Cockburn, A., Highsmith, J. (2001). *Agile software development 2: The people factor*. IEEE Computer.
7. Cohen, S.A. (2010). Software System Development Life Cycle Model for Improved Stakeholders' Communication and Collaboration. *Int. J. of Computers, Communications & Control*, Vol. V, No. 1, pp. 20-4.
8. Demir (2007). *Measurement of Software Project Management Effectiveness*.
9. Dima, A.M. (2006). From Waterfall to Agile software: Development models in the IT sector. Impacts on company management. *Journal of International Studies*, 2, 315-325.
10. Fowler, M., Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35.
11. García-Magariño, I., Gómez-Rodríguez, A., Gómez-Sanz, J., González-Moreno, J.C. (2009). *Ingenias-scrum development process for multi-agent development*. Berlin/Heidelberg: Springer, pp. 108-117.
12. Handriyantini, E. (2014). *Developing Computer-based Educational Game to support Cooperative Learning Strategy*.
13. Hayat, M., Qureshi, M. (2016). *Measuring the Effect of CMMI Quality Standard on Agile Scrum Model*. ArXiv preprint arXiv:1610.03180.

14. Highsmith, J., Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
15. Iftikhar, A., Ali, S.M. (2015). Software quality assurance a study based on Pakistan's software industry. *Pakistan Journal of Engineering, Technology & Science*, 1(2).
16. Kennedy, F.A., Widener, S.K. (2008). *A control framework: Insights from evidence on lean accounting*.
17. Key Scrum Benefits in Project Management, <https://www.workamajig.com/blog/scrum-methodology-guide/scrum-benefits>.
18. Korkala, M., Pikkarainen, M., Conboy, K. (2010). *Combining Agile and Traditional: Customer Communication in Distributed Environment*.
19. Leau et al. (2012). *Development Life Cycle AGILE vs Traditional Approaches*.
20. Liker, J.K., Morgan, J.M. (2006). *The Toyota Way in Services: The Case of Lean Product Development*.
21. *List of Scrum Advantages and Disadvantages*. Indeed.com. <https://www.indeed.com/career-advice/career-development/disadvantages-of-scrum>.
22. Malik, R.S., Ahmad, S.S., Hussain, M.T.H. (2019) *A Review of Agile Methodology in IT Projects*.
23. McHugh et al. (2012). Agile Practices: The Impact on Trust in Software Project Teams. *IEEE Software*, 29, 71-76. 10.1109/MS.2011.118.
24. Miller, E.J., Rice, A.K. (1967). *Systems of Organization: The Control of Task and Sentient Boundaries*. London: Tavistock.
25. Nerur, S., Radha, K.M., Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48, 5.
26. Pfleeger, A.L., Atlee, J.M. (2006). *Software Engineering. Theory and Practice*. Pearson.
27. Pressman, R.S. (2005). *Software Engineering: A Practitioner's Approach*. London: Palgrave Macmillan.
28. Schwaber, K. (1997) Scrum development process. In: *Business Object Design and Implementation* (pp. 117-134). London: Springer.
29. Schwaber, K., Beedle, M. (2001). *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall.
30. Schwaber, K., Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice Hall.
31. Stoica, M., Mircea, M., Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, 17, 64-76. 10.12948/issn14531305/17.4.2013.06.
32. Szalvay, V. (2004). *An Introduction to Agile Software*.
33. *The Waterfall Model: Advantages, disadvantages, and when you should use*, <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>.
34. Unhelkar, B. (2016). *The Art of Agile Practice*. CRC Press. Retrieved from: <https://www.perlego.com/book/1471504/the-art-of-agile-practice-a-composite-approach-for-projects-and-organizations-pdf>

35. Vijayasarathy, L.R. (2008). Agile Software Development: A survey of early adopters. *Journal of Information Technology Management, Vol. XIX, No. 2.*
36. Vresk, A., Pihir, I. Furjan, M.T. (2020). *Agile vs. Traditional Methods for Managing IT Projects.*
37. Womack, J., Jones, D., Roos, D. (1990). *The Machine That Changed the World: The Story of Lean Production, Toyota's Secret Weapon in the Global Car Wars That Is Now Revolutionizing World Industry.*
38. Womack, J.P., Jones (2003). *Lean Thinking.* New York: Free Press.