Principles of Agility used in Modern Software Development

Sandro Schweiss

IU International College

Agile Project Management DLBCSAPM01

Tutor Inga Schlömer

Table of Contents

I. List of Abbreviations

Abbreviation	Meaning
RUP	Rational Unified Project

2. Introduction

2.1 Project Overview

Project management plays a crucial role in today's complex business environment. Effective project management ensures that projects are executed efficiently, on time, and within budget, leading to successful outcomes and the achievement of organizational goals. It provides a structured framework for planning, organizing, and controlling resources, tasks, and activities throughout the project lifecycle. Project management methods, such as Agile, Scrum, and Lean, provide teams with proven strategies and tools to manage projects effectively, foster collaboration, mitigate risks, and optimize resource allocation. By utilizing these methods, organizations can enhance project visibility, improve communication, and increase stakeholder satisfaction. Ultimately, project management and its methods enable businesses to maximize efficiency, minimize errors, and deliver high-quality results, thereby driving growth and competitive advantage in today's fast-paced and demanding marketplace (Institute, 2021).

Therefore, this project report explores the transition from a plan-driven approach to agile project management in a software development agency. The agency specializes in building custom applications for clients and has successfully delivered numerous projects using a traditional plan-driven methodology. However, the agency recognizes the need for greater adaptability and responsiveness in the ever-evolving software development industry. To address this, the agency has chosen a specific project, a client's request for a mobile application to streamline their internal communication processes, as a case study to demonstrate the benefits of adopting agile principles.

2.2 Project Approach

Shifting a project and subsequently an organization from the traditional plan-driven methodology to an agile methodology can be challenging and complex. Many different structures, options, and procedures must be taken into consideration. The goal is to ensure a successful implementation of the agile methodology in the project. The first step, therefore, was to review related literature. In this approach, relevant information from multiple literature sources was extracted. The current state of the art was explored, and possible solutions were outlined.

2.3 Project Methodology

After evaluating the material and use cases, the actual project methodology was chosen. As previously elaborated, many different components must be considered when transitioning towards an agile project management method. But not only is it vital to determine what to consider but also how to implement it. There exist many different types of agile frameworks. Changing the framework can alter the outcome of the procedure significantly. Therefore, no particular Agile framework was chosen. Instead, there is a focus on the overall values and principles of Agile according to the Agile Manifesto (“Principles behind the Agile Manifesto,” 2020).

2.4 Project Planning and Organization

After determining the project methodology, it was necessary to organize and gather information about the company and project and plan accordingly. Understanding the business process, the risks, the impact, and the organization as a whole is necessary for any project to be successful. Therefore, there were various data collected from the project, which is implemented in the Agile method, but also from the organization. This ensures that the right methodology is chosen and if it is even a good idea to shift to another project management method.

2.5 Project Report Structure

This project report is structured to provide a comprehensive analysis of the transition from a plan-driven approach to agile project management. It will include an overview of the principles of agility used in modern software development, an explanation of the current plan-driven approach for software development in the agency, the importance of adopting an agile approach in software development, an exploration of the Agile Manifesto and its translation to software development, and a comparison between agile and plan-driven project management. The report will conclude with insights into the suitability of the agile approach for the future of the organization and recommendations for successful implementation.

3. Principles of Agility used in Modern Software Development

3.1 Overview

3.1.1 Objective of the Project Report

The organization I am associated with is a software development agency that specializes in building custom applications for various clients. With a team of skilled developers and project managers, the agency has successfully delivered numerous projects over the years. However, the prevalent plan-driven approach has become increasingly restrictive, hindering adaptability and responsiveness in an ever-evolving industry. One of the many current projects has been chosen as a case study to demonstrate the transition from the plan-driven approach to other project management methods. The project in question is a client's request for a mobile application to streamline their internal communication processes. Under the previous plan-driven methodology, the project would have followed a predefined sequence of steps, with a detailed project plan and fixed deliverables. The team would have spent a considerable amount of time upfront gathering requirements, creating comprehensive documentation, and defining a rigid timeline for execution. In the agile approach, however, the organization recognizes the importance of flexibility, collaboration, and continuous improvement. By adopting agile principles, the team aims to create an environment where they can respond to changing requirements and client feedback more effectively. This change creates a fundamental shift in software development practices. Therefore, the aim of this report is to utilize the agile methodology, to test, which of the two approaches is more suited for the future of the organization.

3.1.2 What is Plan-Driven Project Management?

Plan-driven project management is a systematic approach to organizing, directing, and measuring the work involved in a given project. It is a method where all activities are planned in advance, and progress is measured against the predetermined plan. This approach is often used for small, well-defined projects with a limited scope and few variables. It focuses on repeatability and predictability, as well as on thorough documentation and verification. Plan-driven methodologies originated from ancient civil engineering projects and have since evolved to manage diverse projects across industries, including software development. These methodologies, also known as traditional or classic, emphasize thorough planning and assume that all aspects of the project can be anticipated and accounted for in the plan. There are many types of plan-driven management methods, the most notable are the waterfall model, the spiral model, and the Rational Unified Process (RUP). These methods divide projects into distinct phases and documents. Each phase is then reviewed and approved before the project moves on to the next phase, which allows thorough planning at each step of the process (Kerzner, 2022).

3.1.3 What is Agile Project Management?

Agile project management emphasizes collaboration, adaptability, and continuous improvement as it is a flexible and iterative approach to project management. In agile project management, projects are broken down into smaller iterations called sprints, typically ranging from one to four weeks. Cross-functional teams work together closely, fostering collaboration, open communication, and shared responsibility. The focus is on delivering incremental value to customers at the end of each sprint, allowing for early feedback and the opportunity to adapt the project's direction based on customer needs. Key practices in agile project management include daily stand-up meetings for team synchronization, backlog management to prioritize and track work items, and regular retrospectives to reflect on lessons learned and improve team processes. The emphasis is on delivering working software or tangible results quickly and frequently, enabling faster feedback loops and the ability to respond to changes or new insights. Agile project management embraces change as a natural and expected part of the project lifecycle. The iterative nature of the approach allows for flexibility, enabling teams to adapt their plans and priorities based on evolving requirements and customer feedback. Continuous improvement is encouraged, with a focus on learning from experiences and applying those learnings to enhance future iterations (“Principles behind the Agile Manifesto,” 2020).

3.2 The Need for an Agile Approach

3.2.1 The Current Plan-Driven Approach for Software Development

As elaborated in chapter 3.1.1 the software development agencies current approach for developing software is a plan-driven approach. The software to be developed is a mobile application to streamline internal communication processes. The way the project would have been structured under a plan-driven approach is as follows.

The plan-driven approach starts with a comprehensive project plan that outlines the project objectives, timeline, resources, and milestones. The project team follows this plan diligently, adhering to the predetermined schedule and executing tasks according to the pre-established sequence. The focus is on following a linear and sequential process, where each phase is completed before moving on to the next. Extensive documentation plays a crucial role here. Detailed requirement specifications, functional designs, and technical specifications are created to provide a clear roadmap for development. These documents serve as reference points for the development team, ensuring that they have a well-defined understanding of the project's goals and requirements. Additionally, quality assurance and verification processes are integrated into the plan-driven approach. Testing activities, including unit testing, integration testing, and system testing, are performed to ensure that the developed software meets the predetermined quality standards. Comprehensive documentation of test plans, test cases, and test results is maintained throughout the testing phase (Kerzner, 2022).

3.2.2 The Importance of Agile in Software Development

The plan-driven approach is best applicable to small projects, which have clearly defined goals and boundaries. Applied to other projects, however, for example, in the software space, the plan-driven approach suffers from several downsides, since the emphasis is on upfront planning and documentation. This leads to challenges in adapting to changing requirements and customer needs. In software development, requirements often change as stakeholders gain more insights and as the market landscape shifts. The rigid nature of the plan-driven approach makes it difficult to incorporate these changes without revisiting the entire plan, potentially causing delays and disruptions, and limiting its ability to handle evolving requirements. Another downside is the lack of customer collaboration and feedback. In a plan-driven approach, customer input is typically limited to the initial requirements-gathering phase, and there may be minimal interaction throughout the development process. This can lead to a mismatch between the delivered product and customer expectations, as there is little opportunity for regular feedback and validation. Furthermore, the plan-driven approach often lacks flexibility in resource allocation. Once the project plan is set, resources are allocated based on the predetermined timeline and requirements. However, as the project progresses, it may become evident that certain resources need to be reallocated or additional resources are required, making it challenging to perform these adjustments efficiently. On the other hand, adopting an agile approach addresses these downsides by embracing change, customer collaboration, and flexibility. Agile methodologies promote iterative development cycles, where requirements are continuously refined and reprioritized. This allows for the incorporation of new insights and changing market conditions without disrupting the overall project progress. Agile methodologies also encourage close collaboration between the development team and the customer. Through regular feedback loops and frequent demonstrations of working software, the customer's needs and preferences can be understood and incorporated into subsequent iterations. This results in a higher likelihood of delivering a product that aligns with customer expectations. Additionally, agile practices enable flexible resource allocation. The cross-functional teams in agile development can self-organize and adapt to changing needs. This allows for optimal utilization of resources and ensures that the right skills are available at the right time (Cohn, 2010).

3.2.3 Managing Complex Projects with the Stacey Matrix

The Stacey Matrix is a conceptual model used to assess the complexity and uncertainty of a project. It helps project teams and stakeholders understand the nature of a project and select an appropriate project management approach. It consists of two dimensions: certainty and complexity. The certainty dimension represents the level of predictability or knowledge available about the project requirements, constraints, and outcomes. The complexity dimension represents the intricacy and interdependencies of the project, including the number of stakeholders, the level of technology involved, and the novelty of the project. The Stacey Matrix is therefore divided into four quadrants. The first is

“Simple”, for projects with a high level of certainty and low complexity. the requirements and outcomes are well-defined, and there is little ambiguity. Traditional, plan-driven approaches are suitable for simple projects since they can be executed using standard procedures and established practices. Then there is “Complicated”, for projects, with either low uncertainty but medium complexity or medium uncertainty and lower complexity. Although the requirements are known, the solutions may require expertise or specialized knowledge. The plan-driven approach is still applicable, but it may require more analysis, coordination, and technical expertise to handle the complexity effectively. The third quadrant is “Complex”, for projects that have both a medium level of uncertainty and a medium complexity. The requirements are not fully known, and there are multiple interdependencies and variables to consider. Agile approaches are well-suited for complex projects as they allow for iterative development, frequent feedback, and adaptation to changing circumstances. Through experimentation and collaboration, agile teams can explore and discover the best solutions for complex problems. At last, there is the section “Chaotic” for projects, that have high levels of uncertainty and high complexity. The requirements and outcomes are unclear and constantly changing. In such situations, a highly adaptive and emergent approach is necessary. The focus is on containing the chaos, stabilizing the situation, and gradually transitioning to a more predictable state (Project Management Institute & Agile Alliance, 2017).

In the case of the mobile application project, it is likely to fall under the category of “Complex”. Internal communication processes can involve multiple stakeholders, intricate workflows, and varying needs across different departments or teams. Additionally, technology advancements and evolving user expectations may introduce uncertainties that cannot be fully anticipated in the initial stages of the project. Considering the complexity of the project, an agile approach aligns well with the nature of the endeavor. Agile methodologies provide the flexibility and adaptability to navigate through the complexities and uncertainties that may arise during development.

3.3 The Agile Manifesto

3.3.1 The Values and Principles of the Agile Manifesto

Agile development is guided by four core values and twelve supporting principles as stated in the Agile Manifesto. The four values are as follows: First, “Individuals and interactions over processes and tools”. This value emphasizes the importance of prioritizing human collaboration and effective communication within a development team. It recognizes that the success of a project lies in the skills, creativity, and cooperation of individuals working together rather than relying solely on processes or tools. Second, “Working software over comprehensive documentation”. This value underscores the significance of delivering tangible and functional software to customers. While documentation is essential, the primary focus should be on producing working software that provides value and meets customer needs. Third, “Customer collaboration over contract negotiation”. This value

emphasizes the importance of active customer involvement throughout the development process. It encourages close collaboration and continuous feedback, allowing for better alignment with customer requirements and ensuring that the end product satisfies their needs. Fourth and last is “Responding to change over following a plan.” This value recognizes that change is inevitable and that the ability to adapt to changing circumstances is crucial for success. It prioritizes flexibility and encourages iterative development, allowing for frequent feedback and adjustments to deliver a better end product. The twelve principles of the Agile Manifesto provide further guidance for implementing the values effectively. These principles emphasize customer satisfaction, active stakeholder involvement, frequent delivery of working software, a collaboration between developers and business teams, and embracing change as a competitive advantage. The twelve principles are: satisfying the customer through early and continuous delivery, changing requirements should be welcomed, frequent delivery of working software, daily cooperation for the duration of the project, the project is built around motivated individuals, promoting face-to-face communication, the product is the primary measure of progress, continuous improvement, promoting simplicity in design and processes, fostering self-organizing teams and ensuring technical excellence. By embracing these values and principles, agile teams can foster a culture of collaboration, adaptability, and continuous improvement, leading to more efficient and successful software development projects (“Principles behind the Agile Manifesto,” 2020).

3.3.2 The Translation of the Agile Manifesto to Software Development

The Agile Manifesto principles serve as a guide for implementing agile practices and connecting them to the core values of the Agile Manifesto. In the context of the mobile application project to streamline internal communication processes, these principles can provide valuable insights and direction. One principle that aligns with the project is "Deliver working software frequently, with a preference for shorter timescales." This principle emphasizes the importance of delivering tangible value to the customer in the form of functional software. In the context of the internal communication application, it means that the team should focus on delivering incremental features and updates in short iterations. This approach allows for regular feedback and ensures that the application remains aligned with evolving user needs and organizational requirements. Another relevant principle is "Business people and developers must work together daily throughout the project." For the internal communication application, this principle highlights the need for close collaboration between the development team and the stakeholders who understand the communication needs within the organization. By working together on a daily basis, the team can ensure that the application addresses specific points and provides effective solutions for streamlining internal communication processes. The principle of "Build projects around motivated individuals. "Give them the environment and support they need" is also applicable. In the context of the mobile application project, it emphasizes the

importance of empowering and supporting the development team. Providing them with the necessary resources, autonomy, and a collaborative work environment will enable them to make informed decisions, innovate, and drive the project's success. Furthermore, the principles of "Welcome changing requirements" and "Simplicity is essential" are valuable for the application. The project team should embrace flexibility to accommodate evolving user needs and focus on delivering a streamlined and user-friendly experience while avoiding unnecessary complexities that could hinder adoption and usability.

By adhering to these Agile Manifesto principles, the development team can apply agile practices such as iterative development, continuous feedback, close collaboration with stakeholders, and the ability to respond to changing requirements. This, in turn, enables the project to successfully deliver a mobile application that effectively streamlines internal communication processes and aligns with the values and principles of the Agile Manifesto (“Principles behind the Agile Manifesto,” 2020).

3.4 Comparison of Agile and Plan-Driven Project Management

3.4.1 The Key Differences between Agile and Plan-Driven Approaches

Agile and plan-driven development are two contrasting approaches to software development, differing in their underlying philosophies and methodologies. Agile development emphasizes flexibility, collaboration, and iterative delivery. It values adaptive planning, continuous feedback, and customer involvement throughout the project. Agile methodologies focus on delivering incremental value in short iterations and embracing change. They promote self-organizing teams and prioritize working software over comprehensive documentation. Agile development is well-suited for complex and uncertain projects where requirements may evolve over time. On the other hand, plan-driven development follows a more structured and predictive approach. It emphasizes detailed planning, documentation, and adherence to a predetermined plan. Plan-driven methodologies, such as Waterfall or the V-model, involve sequential phases and a thorough upfront analysis of requirements. This approach assumes that all project requirements can be accurately defined in advance. It suits projects with well-understood and stable requirements, where predictability and control are essential. The key differences lie in their response to change, level of customer involvement, and flexibility. Agile embraces change, welcomes evolving requirements, and encourages continuous feedback from customers. Plan-driven approaches, however, resist changes once the plan is established and typically involve limited customer collaboration during development. Agile provides flexibility through iterative development and short feedback loops, while plan-driven approaches follow a more rigid sequential process. Making Agile overall a more suitable project management method for software (Maximini, 2015).

3.4.2 The Transition from Plan-Driven to Agile Project Management

Transitioning from a plan-driven approach to an agile approach in an organization can pose various challenges, particularly in the context of a company, where there is already an established project management philosophy. The shift requires a significant mindset change, cultural transformation, and adjustments to existing processes and structures. One of the main challenges is resistance to change. Individuals accustomed to the plan-driven approach may be reluctant to embrace the agile mindset, which emphasizes flexibility and adaptability. Resistance can stem from a fear of losing control, uncertainty about new roles and responsibilities, or skepticism about the effectiveness of agile methods. Overcoming this challenge requires effective change management strategies, clear communication, and stakeholder involvement to address concerns.

A cultural shift is another challenge that needs to be addressed. Agile principles emphasize collaboration, self-organization, and continuous improvement. In contrast, plan-driven approaches often rely on hierarchical decision-making and individual accountability. Transitioning to an agile approach requires fostering a culture of trust, empowerment, and open communication, where teams are encouraged to take ownership and collaborate across functions. This cultural shift can take time and effort, involving training, coaching, and creating a supportive environment that values experimentation and learning. Adjusting existing processes and structures is another significant challenge. Agile methodologies introduce new practices, ceremonies, and roles. This requires redefining project roles, establishing cross-functional teams, and implementing iterative development cycles. It may involve changes in project planning, tracking progress, and managing stakeholder expectations. Adapting the existing organizational processes and structures to accommodate agile practices can be complex and requires careful planning and alignment with business objectives. Furthermore, integrating agile practices with the existing project portfolio and organizational governance can be challenging. Agile projects typically involve shorter planning cycles, dynamic prioritization, and frequent feedback loops. This contrasts with traditional management and governance frameworks that are often based on long-term planning and control. Harmonizing these approaches and finding a balance between agility and organizational control is crucial. However, the long-term gain of the shift in project management philosophy is worth the risk. (Kerzner, 2022).

4. Conclusion

The following chapter concludes the project report; here, the project results are summarized and discussed.

In conclusion, the project report shows the significance of adopting an agile approach in modern software development. The key findings and insights indicate that the prevalent plan-driven approach can be restrictive, hindering adaptability and responsiveness in an ever-evolving industry. The project report on transitioning from a plan-driven approach to an agile methodology for a mobile application project showcases the benefits of flexibility, collaboration, and continuous improvement. The plan-driven approach, although suitable for small, well-defined projects, faces challenges in handling changing requirements and customer needs, lacks regular customer collaboration and feedback, and lacks flexibility in resource allocation. In contrast, the agile approach addresses these downsides by embracing change, fostering customer collaboration, and enabling flexible resource allocation. Agile methodologies, with their iterative development cycles, frequent feedback loops, and adaptability, allow teams to navigate complexities and uncertainties more effectively. The Agile Manifesto, with its core values and principles, provides guidance for implementing agile practices. The four core values prioritize individuals and interactions, working software, customer collaboration, and responding to change. The twelve supporting principles further emphasize customer satisfaction, active stakeholder involvement, continuous delivery of working software, a collaboration between developers and business teams, and continuous improvement (“Principles behind the Agile Manifesto,” 2020). For the mobile application project, the Agile Manifesto principles can be translated into specific actions. Delivering working software frequently, collaborating daily with stakeholders, building projects around motivated individuals, welcoming changing requirements, and promoting simplicity are all applicable to streamlining internal communication processes effectively. It is recommended that the organization embrace agile methodologies more widely in its software development practices. This can be achieved by providing training and support to teams in adopting agile practices, establishing cross-functional and self-organizing teams, and fostering a culture of collaboration and continuous improvement. Regular retrospectives should be conducted to reflect on lessons learned and enhance team processes. Additionally, the organization should ensure that there is participation from all stakeholders, including customers, to encourage active involvement and feedback throughout the development process.

V. Bibliography

Cohn, M. (2010). *Succeeding with agile: Software development using Scrum*. *The Addison-Wesley signature series A Mike Cohn signature book*. Upper Saddle River, NJ, Munich: Addison-Wesley.

Institute, P. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and The Standard for Project Management (ENGLISH)* (7th edition). Erscheinungsort nicht ermittelbar, Boston, MA: Project Management Institute; Safari. Retrieved from https://learning.oreilly.com/library/view/-/9781628256642/?ar

Kerzner, H. (2022). *Project management: A systems approach to planning, scheduling, and controlling* (Thirteenth edition). Hoboken, New Jersey: John Wiley & Sons, Inc.

Maximini, D. (2015). *The Scrum Culture*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-11827-7

Principles behind the Agile Manifesto (2020, June 9). Retrieved from https://agilemanifesto.org/iso/en/principles.html

Project Management Institute; Agile Alliance (2017). *Agile practice guide*. Newton Square, Pennsylvania: Project Management Institute Inc.