This coursework is a 2000-3000 word essay researching the reasons for, and evolution of, agile program and project management methodologies.

Introduction

In the current age many projects are undertaken using Agile methodologies. Their popularity has increased throughout the history of technological developments where it used to be that set out rules and plans had to be followed strictly – the waterfall method. Agile methodologies are able to provide faster results – due to the ability to quickly produce working prototypes and better manage risks – if there is a problem with the development process it will cost less as adjustments can be made incrementally.

History of agile

Agile methodologies are relatively new: The Agile Manifesto launched in 2001, where “17 technologists drafter the Agile Manifesto” (Sacolick, 2022). They wrote about the principles of agile project management with 4 major principles to guide developing teams on. These being: “Individuals and interactions over processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation. Responding to change over following a plan.” (Sacolick, 2022)

Although this has only formerly been introduced in 2001, there are roots of Agile methods dating further back in software engineering. Iterative and incremental developments have been in use since 1957. People who didn’t like the waterfall method approach suggested ideas such as involving the customer, working software over documentation and response to change which were Agile ideas that are used today. (Abbas, Gravell and Wills, 2008).

Agile had emerged as an alternative to the Waterfall model, which was the traditional way to manage and plan projects. It required a large amount of time and resources, which often lead to delays and cost overruns because issues in with the development would cost a lot of time and money to solve. On the other hand, Agile focuses on incremental development which provides faster results in working software, greater flexibility to change project requirements based on response from customers.

Scrum

Another form of Agile development is Scrum, which actually came 8 years before the Agile Manifesto, but it is considered to be part of agile because it is based on the same incremental and iterative approach that agile is based on. What is SCRUM? It is an agile software development method that can be used to control, manage, and develop complex software using iterative and incremental practices. It was first introduced to the public at a conference in 1996. It is a simple practical process which captures key agile characteristics and practices including: “self-managing, cross-functional, self-directed, self organizing and collocated teams of eight or fewer team members develop software in sprints” ( Cho and Joey, 2010, page 26 ). Sprints being fixed term development periods, where each sprint done delivers changed or additions to the program that can be tested, reviewed, and given feedback on. Once the sprint objective is set no additional work can be added unless it is done by the actual development team. The leader of the group, called the Scrum master manages the teams that are developing and ensures they are successful in the outcomes of each sprint. Another aspect where Scrum is similar to Agile is that it “allows requirements, architecture and design to emerge over the course of the project”. (Cho and Joey, 2010, page 26)

Teamwork is essential in order for Agile project management to work. It is important that everyone involved is aware of the project goals, tasks and objective. With these in mind they need to be able to work together to achieve them. This requires effective communication and collaboration between all parties included: developers, managers, clients etc. Alongside this having defined roles and expectations for the project is a good way to not overlap work and make sure everyone stays on track. To maximize benefits, leaders need to focus on frequent inspection of work and adaptation to any issues in the workflow. This means that teams should be continuously reviewing progress and making any necessary changes to stay on track. Regular meetings for reviews and sprint planning can help keep things in check and making sure everyone is up to date and on track. However, many companies deal with loss of time and money to unproductive meetings. Many meetings nowadays aren’t good, it isn’t because they are lengthy or boring, but because they are devoid of a proper business objective ( Magnet, 2011 ). With this in mind, I should also mention that these meetings need to be focused and productive.

Advantages and disadvantages of SCRUM

With many advantages, it is important to know that are also some drawbacks which should be considered. Agile can be difficult to manage. This is because of the frequent changes, agile requires a lot of management and coordination to ensure that tasks are completed on time and customer requirements are met. Some teams may even find it difficult to work in an Agile project. They may not be able to adopt the needed skills and tasks that are fundamental to Agile, they would prefer working with a detailed plan in front of them rather than them taking risks and implementing own ideas. In addition, it can become unpredictable and risky. Without a comprehensive plan and detailed designs, it can be difficult to predict how long a project will take or how much it will cost. This can be a big problem for business without an unlimited budget and time frame.

The waterfall method

The waterfall method was published in 1970 by Winston W. Royce and has been used far and wide in software development. The waterfall method is based on well-designed plans for different parts of the software. Large-scale software “requires a base programming paradigm or model” (McCormick, 2012). In the software development process, programming models are used to plan various development stages for the application. This is the waterfall method. There are 6 different phases in the waterfall method:

1. Requirements – involves understanding what is needed for the design and functions. Unless you have clear set of requirements you cannot proceed with the projects.
2. Analysis – Analysis of software and hardware needed for project. From deciding what language to use to database system which program will run on.
3. Design – Algorithm or flowchart of software or software code to be written in next stage is created now. It relies on previous two stages for it to work.
4. Coding – Based on what was designed in the sage before, in this stage actual coding is carried out. “A proper execution of the previous stages ensures a smooth and easier implementation of this stage “ (McCormick, 2012)
5. Testing – With coding done, testing the code is necessary. Check for any flaws and if the software was created to the provided requirements.
6. Maintenance – Providing the client with some support regarding the software and if the client requests new or changes to the software the development process must start again.

Although it is costly to change both to time and budget, it still remains “one of the most commonly used methodologies”. (McCormick, 2012)

Advantages and Disadvantages of Waterfall

There is clear structure compared to other methodologies. Waterfall focuses on clearly laid out steps. The structure is simple, as mentioned above. Since the development team has to go through each step before moving to the next one, if there are any issues they can be brought up and solved immediately before continuing.

You are able to determine the end goal early using waterfall. Committing to the goal at the very beginning means teams can avoid straying away from the plan. Unlike SCRUM which divides projects up into individual sprints, Waterfall keeps the focus on the end goal at times. Waterfall eliminates the risk of the team getting stalled as they worked towards that goal.

One of the biggest disadvantages is that it makes changes difficult. Because you have to follow the steps and complete them fully before going forward, once they are complete they cannot be changed without a considerable amount of work and time being put in and with work becoming pointless which can impact the project timeline.

It excludes the client from the development phase, which in the end can cause issues with the development stage. Unlike the Agile methodologies where the client is kept in the loop and can give feedback on the software, if there are any changes they want to certain features they can give them and give more feedback on how the software should run as a whole. Waterfall excluding this means that if the client changes their mind or would like a certain feature to work a different way or change to another feature completely the whole waterfall process might have to start again. This will cost a lot of time and money, and in this regard makes agile methodologies more suitable for software that might require sudden changes.

Waterfall delays testing until after completion. With testing being so late in the development stages means that information on how the program works cannot be taken until almost the very end of the stages so again changes/issues in the program will cause delays, leading to decreased customer satisfaction.

(LucidChart, 2017)

Extreme Programming (XP)

It is a development methodology that was introduced in 1996 by Kent Beck as a lightweight agile framework. (productplan.com, n.d.) which focuses a lot on creating high-quality programs tailored to customers requirements in a quick time period. It bases itself on frequent releases of useable software in short bursts of development periods. It develops, tests and releases software incrementally, essentially creating ‘checkpoints’ in the development process. XP is considered to be an agile methodology, which makes sense due to its iterative nature. It is a good methodology to follow for teams that need to quickly release working software and respond to sudden changes in customer requirements.

Same as other methodologies there are certain rules to follow, and XP is no different. It helps remind development teams about how and what needs to be done.

There should be some planning done, where the first requirements are drawn out, what functionality is required, and what features have priority over others. These should provide enough detail for teams to estimate how long it will take to implement the required features and some sort of project timeline can also be created.

Management of XP teams is crucial and for it to be successful it needs to meet certain standards. Some of these include: daily meetings, a workspace where team members work together, determining length of each iteration, reassigning people/tasks as needed to avoid interruptions, adjusting XP if it is not running smoothly.

In XP you start with the simplest design that can possibly work, then add complexity when it is required and always refactor code to keep the design simple and have clean code.

Finally there are the rules of coding and testing, with coding needing to always have the customer available to keep in touch with all stages of the project, writing code according the set out standards everybody has agreed on, making it easier for the whole team to read, understand and refactor. Practicing pair programming, Integrating and committing code many times a day to avoid different developments happening. With testing, all code is tested as is created and new test methods are created when defects are found, to ensure all bugs are found and fixed before software is released. (Lucidchart, n.d.)

Advantages and Disadvantages

One of the main advantages of XP is that it allow companies to save time and costs when compared to the waterfall methodology. This is because XP focuses on “timely deliveries of final products” (Kukhnavets, 2018)

XP’s process is visible, you can see what gets done and when. Developers will commit to the code what they solve and everyone can see their progress.

A disadvantage is that it may not be the best option if developers are not working in the same workspace as you need high levels of reliable communication and connection for XP to work effectively.

Another disadvantage is that good design is important in software applications as it helps selling it but XP doesn’t focus on that but rather on coding. Documentation is not always great and sometimes is lacking, leading to occurrence of similar bugs in the code in the future as fixes may not be recorded. (Kukhnavets, 2018)

The Spiral methodology

Spiral, introduced in 1986 is a technical approach methodology who’s focus is on planning, risk analysis, development and evaluation. It is similar to the waterfall methodology as the development process go through each phase, however multiple times to ensure the software is prepared for deployment. The Spiral methodology focuses a lot on risk analysis and always goes through multiple solution possibilities before actually implementing one.

After the first iteration of the spiral is completed, the client gives feedback which is acted upon in future iterations. Spiral methodology is suitable for medium/large scale projects. (Burtsecu et al. 2014 page 43)

There is a lot of conflicting opinions on whether spiral is and Agile methodology or not. What it is, in my opinion, is a methodology closely similar to the waterfall methodology as it involves going through each phase at one time, however it brings iterative approach to it. Working on the bigger problem by breaking it down into smaller tasks and receiving feedback on each iteration of the development and keeping it in mind for the next spiral cycles.

Advantages and disadvantages

Software is being produced early in the development cycle.

Risk handling is an advantage of Spiral, as it takes place at every phase.

It is suitable for risky projects where business may be unstable, this is due to strong documentation, flexibility in requirements where these can be easily changed.

Some disadvantages include: It is costly, therefore it is not suitable for small projects. The process is complex, much more than other methods. It is dependable on Risk Analysis and requires expertise. End of project may not be known early as requirements may change therefore final product will change.

Conclusion

In conclusion, Agile project and program management methodologies have evolved significantly since their inception in the 1940s. Agile provides faster results and better risk management, as well as greater flexibility in response to changing requirements, making it the preferred approach for many businesses. Various frameworks and methodologies have also been developed, allowing Agile to be tailored to specific projects.

Words: 2368

References

Sacolick, I. (2022). *A brief history of the agile methodology*. [online] InfoWorld. Available at: <https://www.infoworld.com/article/3655646/a-brief-history-of-the-agile-methodology.html#:~:text=Agile%20was%20formally%20launched%20in>.

‌

Abbas, N., Gravell, A.M. and Wills, G.B. (2008). Historical Roots of Agile Methods: Where Did ‘Agile Thinking’ Come From?. *Lecture Notes in Business Information Processing*, pp.94–103. doi: <https://doi.org/10.1007/978-3-540-68255-4_10>.

‌

Cho, J. and Joey (2010). *An Exploratory Study on Issues and Challenges of Agile Software An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum Development with Scrum Recommended Citation Recommended Citation*. [online] Available at: <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1595&context=etd>.

‌

Magnet, P. (2021). *Meetings Are Killing Software Development*. [online] Medium. Available at: https://tipsnguts.medium.com/meetings-are-killing-software-development-1746c94cd362 [Accessed 27 Feb. 2023].

Mccormick, M. (2012). *Waterfall vs. Agile Methodology*. [online] Available at: <http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf>.

‌

LucidChart (2017). *The Pros and Cons of Waterfall Methodology*. [online] Lucidchart.com. Available at: <https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology>.

www.productplan.com. (n.d.). *eXtreme Programming (XP)*. [online] Available at: <https://www.productplan.com/glossary/extreme-programming/#:~:text=What> [Accessed 27 Feb. 2023].

‌Lucidchart. (n.d.). *Extreme Programming vs. Agile Methodologies*. [online] Available at: <https://www.lucidchart.com/blog/extreme-programming-vs-agile?page=1> [Accessed 27 Feb. 2023].

‌

Kukhnavets P. (2018). *Hygger*. [online] Hygger. Available at: <https://hygger.io/blog/disadvantages-and-advantages-of-extreme-programming/>.

‌

Burtescu, E., Cooper, J., Dardala, M., Dusmanescu, Fotache, M., Garlasu, D., Guran, M., Ticiano, L., Jordão, C., Kahanwal, Konstantas, D., Hitesh, K., Sharma, Nithchi, S., Paraschiv, C., Popescu, D., Gheorghe Sabau, Shah, N. and Wrembel, R. (2014). Database Systems Journal BOARD. *Database Systems Journal*, [online] (3), p.1. Available at: <http://dbjournal.ro/archive/17/17.pdf#page=38>.