LONDON
METROPOLITAN
UNIVERSITY

CS5002 - Software Engineering

Assessing the Benefits and Drawbacks of Different Software Development Life Cycle (SDLC) Models: A Comprehensive Review

Group Coursework Assignment.

Brandon Eddy - id : 20017685

Manolis Diamandakis - id : 20023410
Jaxon Stewart - id : 20012387

Dan-Marius Bradea - id : 20044497

Coursework no: 001
Instructor : Ramzi Djemai
8 February 2023

Table of Contents

1.0 - Introduction

Software Development Life Cycle (SDLC) is a systematic approach to software development that provides a structured framework for designing, building, and maintaining software systems. The purpose of SDLC is to ensure that software systems are delivered on time, within budget, and with the desired level of quality. There are various models of SDLC, each with its unique strengths and weaknesses, including traditional approaches like the Waterfall model and more modern approaches like Agile methodology.

This research aims to compare and analyse the various SDLC models to determine their limitations, strengths, and benefits. Through a comprehensive examination of the existing literature, this research will provide insights into the most effective SDLC model for delivering high-quality software systems.

Throughout this paper we will analyse key papers which analyse SDLC, from these papers we aim to gain a greater understanding of the research that has been done around the SDLC. In particular, we want to know the variety of methodologies that exist, what are the strengths and weaknesses of the different types. Lastly, where it is best to apply each of them to enable developers to create the highest quality software, in the most efficient manner.

2.0 - Papers

2.1 - Software Development Life Cycle (SDLC) Analytical Comparison And Survey On Traditional And Agile Methodology

In the ever-evolving world of software development, the choice of software development life cycle *(SDLC)* model can greatly impact the success of a project. This research aims to compare and evaluate the traditional SDLC models with the more recent Agile methodology, in terms of their effectiveness, limitations, and suitability for different projects. Upon conducting my research, I have found the various advantages and disadvantages of the various methodologies and this research aims to fill the current gap when deciding on what methodology would work best for projects. The common pattern I have found is that Agile is the most used methodology, particularly Scrum. *(Figure 1)*

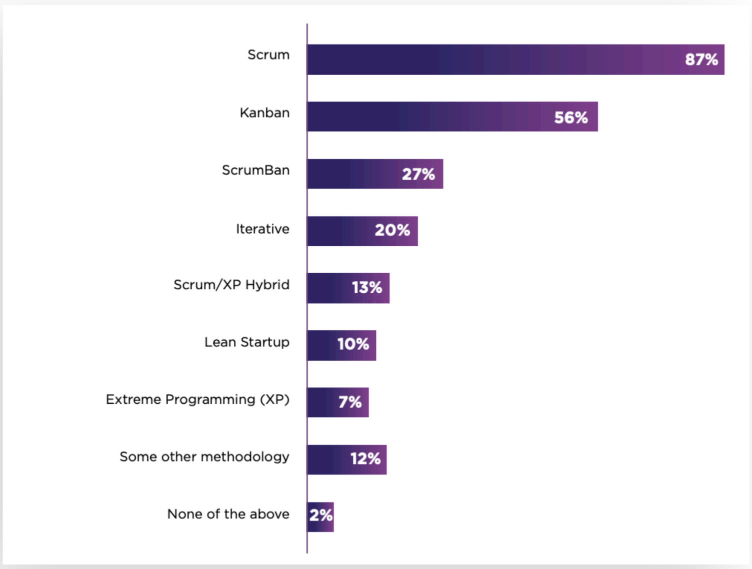
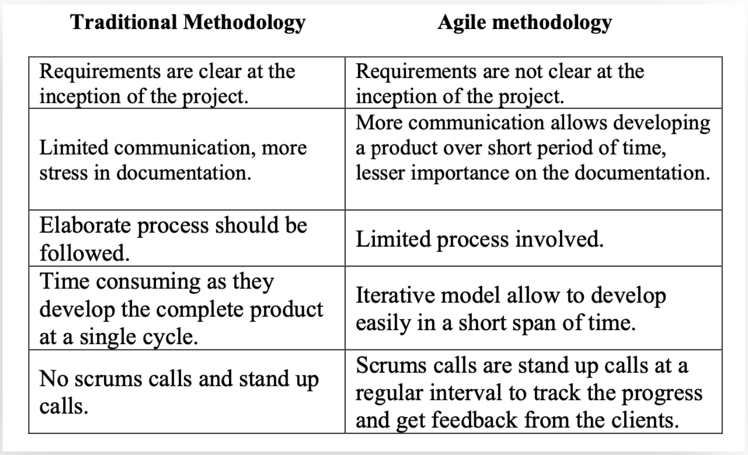


Figure 1 - Most popular methodologies 2022. (Digital.ai, 2022)

The study compares the two approaches in terms of their effect on the quality assurance process in software development projects. A case study approach was used to evaluate the quality assurance practices of software development projects that utilized either the Agile methodology or the traditional Waterfall SDLC. The study found that while both approaches have their advantages and disadvantages, the Agile methodology was more effective in ensuring quality in software development projects. *(Dora and Dubey, 2013)*

Agile Software Development is an adaptive and flexible approach to projects, with a high degree of satisfaction among team members. Iterations are short to provide faster feedback from the client to the project team. Thus, giving you better results while being able to adapt from one methodology to another on a single project if needed. Furthermore, testing commences in parallel to development which also leads to fewer errors. *(Dora and Dubey, 2013)*

In conclusion, the comparison of traditional and agile SDLC models has revealed several trade-offs *(Figure 2)*, with both approaches having unique advantages and disadvantages. The best approach for a particular software development project will depend on a variety of factors, including the specific needs of the project, the development team, and the organisation.



Traditional Methodology	Agile methodology
Requirements are clear at the inception of the project.	Requirements are not clear at the inception of the project.
Limited communication, more stress in documentation.	More communication allows developing a product over short period of time, lesser importance on the documentation.
Elaborate process should be followed.	Limited process involved.
Time consuming as they develop the complete product at a single cycle.	Iterative model allow to develop easily in a short span of time.
No scrums calls and stand up calls.	Scrums calls are stand up calls at a regular interval to track the progress and get feedback from the clients.

Figure 2 - Some differences between Traditional and Agile methodology (Dora and Dubey, 2013)

2.2 - A Spiral Model Of Software Development And Enhancement & A Comparison Between The Traditional Waterfall Model And Agile Methodologies In Software Development

Over time, software development methodologies have evolved, each with its own set of strengths and weaknesses. The Waterfall Model, Agile Methodologies, and Spiral Model are the three most popular approaches discussed in this article.

The Waterfall Model (Shatat, 2015) is a linear sequential approach that works best for projects with well-defined requirements and little room for change. It has a well-defined and structured process, but it lacks the adaptability to accommodate changes as the project progresses.

Agile Methodologies (Shatat, 2015) are iterative and adaptive approaches that work well for projects with constantly changing requirements. The approach prioritises delivering a working product as soon as possible and allows for continuous feedback and improvement. However, a lack of structure and a clearly defined process can lead to inconsistent results.

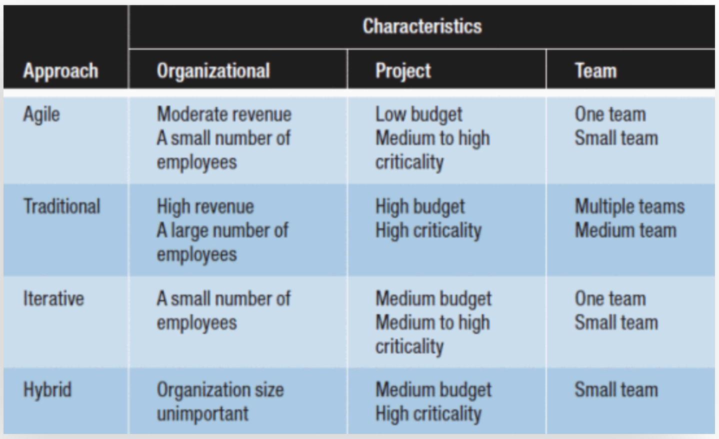
The Spiral Model (Boehm, 1988) is a structured and risk-driven approach to development that allows for changes to be made throughout the process. The model is appropriate for complex projects, but it necessitates a significant time and resource investment for risk management.

In conclusion, the choice between the Waterfall Model, Agile Methodologies, and Spiral Model is determined by the project's specific needs and requirements. According to surveys, Agile Methodologies are the most widely used approach right now; however, the best approach depends on the specific needs of the project. It is critical to weigh the benefits and drawbacks of each approach and select the one that best meets the project's requirements.

2.3 - Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?

"Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?", is a survey which asked 2000 professionals in different sectors, what software development methodologies they used. It also asked these professionals questions about the nature of the company they worked for, from this information the report can infer conclusions about how different methodologies lend themselves well (and badly) to different projects based on: the size of the projects, the revenue of the companies, the number of teams involved, the size of the teams involved, the length of the projects and project criticality.

Based on this data the paper concludes that the different approaches vary in three main characteristics, Organisational(No of employees and business revenue), Project (Budget and project criticality) and Team (No. of teams and team size). Based on the data it compiles its conclusion in *(Figure 3)* which shows that more lightweight approaches(such as Agile) lend themselves well to smaller organisations with smaller budgets and teams while more heavyweight (Traditional) methodologies lend themselves well to larger organisations with higher budgets and larger teams.



Approach	Characteristics		
	Organizational	Project	Team
Agile	Moderate revenue A small number of employees	Low budget Medium to high criticality	One team Small team
Traditional	High revenue A large number of employees	High budget High criticality	Multiple teams Medium team
Iterative	A small number of employees	Medium budget Medium to high criticality	One team Small team
Hybrid	Organization size unimportant	Medium budget High criticality	Small team

Figure 3 - Table indicating how different approaches vary and the characteristics of the projects they are informing

2.4 - An Excursion to Software Development Life Cycle Models: An Old To Ever-Growing Models

Every Software Development Life Cycle Model has its own application in which it is more effective. These can be compared through a series of features that each model has in similarity or difference to another. For example, the Waterfall model features a linear sequential work flow and focus on complete development, while RAD features an incremental process and focus on quick development. In addition to the differences between the aforementioned traditional SDLC models, the topic also includes the distinct Agile Models.

Two well-known examples of contrasting Agile models can be Scrum, featuring a focus on delivery of an application according to the system developer, and the Dynamic System Development Method, focusing on delivery based on the user needs. In addition to these two groups of models, hybrid models are also used in the industry, in the event the mid-point between the different categories is required.

To conclude, the plethora of models within the Software Development Life Cycle can be picked, chosen and morphed based on the preferences of developers, management as well as the user/client side of a project. The chosen method will determine the path of the development life cycle and the features within, impacting the journey to the end result.

3.0 - Review of key papers

3.1 Brandon Eddy - *Software Development Life Cycle (SDLC) Analytical Comparison And Survey On Traditional And Agile Methodology*

- Analyses the strengths and weaknesses of traditional and agile SDLC methodologies.
- Compares the impact of each methodology on the software development process.
- Identifies best practices for selecting the appropriate SDLC model for specific projects.

3.2 Dan-Marius Bradea - *A Spiral Model Of Software Development And Enhancement & A Comparison Between The Traditional Waterfall Model And Agile Methodologies In Software Development*

- Waterfall Model: best for projects with defined requirements and limited scope for change *(Shatat, 2015)*.
- Agile Methodologies: adaptable and flexible, ideal for projects with changing requirements *(Shatat, 2015)*.
- Spiral Model: structured and risk-driven, allows for changes, but may not be appropriate for all projects due to complexity and required resources *(Boehm, 1988)*.

3.3 Manolis Diamandakis - *Choice of Software Development Methodologies: Do Organizational, Project, and Team Characteristics Matter?*

- A survey asking professionals about what software development methodologies they use
- About the nature of the organisations they work for, i.e. Revenue, No. Of Employees, the sector in which the organisation operates.
- Asks professionals about the nature of the projects they are working on. This includes the size of the teams, the budget of the project, the number of teams working on it and the criticality of the project to the organisation as a whole.
- Conclusions about what different methodologies are best suited for.

3.4 Jaxon Stewart - *An Excursion to Software Development Life Cycle Models: An Old To Ever-Growing Models*

- Applications of Software development life cycle models based on their individual differences.
- The difference between Agile and Traditional models, as well as the need for Hybrid methods.
- Choosing a specific SDLC model based on the imposed application and requirements.

4.0 - Summary

The methodology used in the literature review involved a collaborative effort among all group members. Each member was tasked with examining a specific aspect of the topic, providing their individual critiques and conclusions based on their research. The results of these individual assessments were then compiled to arrive at a comprehensive overview of the subject.

Dan focused on comparing three specific SDLC models to highlight the preferences at different levels of an organisation, including the developer, user, and management. Brandon analysed the strengths and weaknesses of both traditional and agile SDLC methodologies, identifying the best practices for each scenario. Jaxon evaluated the practical applications of different SDLC models, comparing traditional, agile, and hybrid methodologies. Manolis gathered real-world data from individuals in the software industry to determine the most commonly used models and the factors affecting their selection.

Furthermore, the results were analysed to arrive at conclusions on the best models for different projects, taking into account the nature of the various project. The examination of existing literature reveals that each model of SDLC has its own strengths and weaknesses and should be chosen accordingly. While highlighting the importance for software developers to carefully consider project requirements and stakeholder preferences when selecting the most appropriate SDLC model.

5.0 - Conclusion

In conclusion, based on the research, the research highlights that both traditional and agile SDLC methodologies have their own strengths and weaknesses, and the selection of a specific model ultimately depends on the nature of the project and the preferences of the stakeholders involved. The findings suggest that hybrid models, which incorporate elements from both traditional and agile methodologies, may offer the most flexible and efficient approach to software development.

The following research conducted by Dan focused on comparing three specific SLDC models to highlight the preferences at different levels. Brandon analysed the strengths and weaknesses of both traditional and agile SDLC methodologies and identifying the best practices for each scenario. Jaxon evaluated practical applications of different SDLC models, comparing traditional, agile, and hybrid methodologies. Lastly, Manos gathered real-world data from individuals in the software industry to determine the most commonly used models and the factors affecting their selection.

Possible avenues for future research could include an in-depth analysis of the impact of team size and composition, project complexity, and other contextual factors on the selection and effectiveness of different SDLC models.

Overall, the results of the studies highlight the importance of considering the project requirements and stakeholder preferences when selecting the most appropriate SDLC model for a given project.

Furthermore, this research provides valuable insights into the most effective methodologies for delivering high-quality software systems and contributes to the ongoing discussion on best practices in software development.

6.0 - Evidence of teamwork

First Meeting 12/01/23 - Library 10:30 am

Group convened in order to decide what topic would be the best fit for the four of us. We wanted a topic that would interest everyone while also being broad enough that it would yield a lot of research papers that we could use.

We have ended up deciding on researching Software Development Life Cycle Methodologies. We are going to aim to learn more about the differences between the different methodologies, how they are different, the strengths and weaknesses of them and how best to apply each of them.

Dividing up initial work: what do we want to do today?

We are going to start by doing initial research into the field so that we get a better understanding of what sort of information is out there. We agree that getting a large body of research in our chosen topic is going to make the rest of the work a lot easier.

We have discussed how to do the layout of the report and poster, and we have decided to all write our own short research reports on each paper we have found.

Once we have finished we will reference what we wrote and continue on and write the summary on the group report.

We will take the best paper from each member and write a little summary of our findings.

The intro and conclusion will discuss our assumptions and findings in the reports.

Second Meeting 31/01/23 - Library 11:30 am

Coming back from break we have shown each other the papers we have picked and the summaries we have written. We need to put together the group project now. Once we all have our papers written, we can move forward on this process. We just have had some mitigating circumstances come about with one of our members over the break, so we have been on a pause until today.

CHECKLIST

Finalise our findings for each papers

Gather our reference pages

Pick the best paper out of the 3 reports we have researched

Move into finishing group report

Next meeting next week we will put together the group report, and finish posters.

Meeting 2.5 01/02/23 - Library 13:30

We realised that each of us doing an individual report was pointless, after doing the research and asking Questions of Ramzi we realised it more appropriate to just move straight from research to poster design. Deadline is fast approaching so we decided that we would all try and finish our posters by next Wednesday. We would do this so that we would all have a high enough knowledge of our individual papers that when we next meet it would simply be a process of putting all of our information together into the group report and we would hopefully finish the group report at our next meeting.

CHECKLIST

- Complete any research that has not yet been completed
- Complete Posters in time for next meeting

Third Meeting 08/02/23

At this meeting we have all successfully finished our posters and are ready to begin work on the group report. To start with we each wrote a summary of our chosen papers. These summaries were to make up the Papers section of our group report. We then divided up the remainder of the sections to make sure that each of us were doing a fair amount of work.

Brandon was in charge of compiling all our individual work into one succinct piece of work which was the group report Brandon also compiled all of the papers to make up the References section. Manos and Jaxon were in charge of writing minutes and the Introduction. Dan was in charge of the conclusion. We all wrote our own short summaries in bullet points to make up the summaries section of our paper.

We finished the large majority of the work today leaving only dregs to finish in the following days. The group report has been compiled and every section filled in. We only needed to alter small details and make minor changes over the coming few days such as adding in our WhatsApp conversations for proof of teamwork

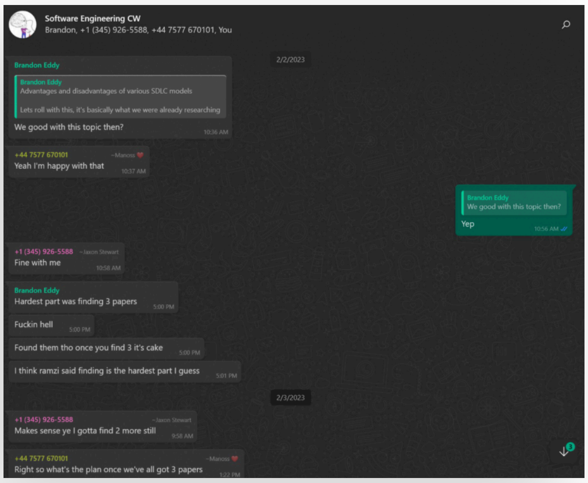
CHECKLIST

Complete individual summaries of papers in the form of a short text and a series of bullet points.

Discuss what the nature of the introduction and conclusion should be so that individuals can write them

Compile all our individual texts into one succinct group report.

Group Discussions


Software Engineering CW
Brandon, +1 (345) 926-5588, +44 7577 670101, You

2/2/2023

Brandon Eddy
> Brandon Eddy
> Advantages and disadvantages of various SDLC models
>
> Lets roll with this, it's basically what we were already researching

We good with this topic then? 10:36 AM

+44 7577 670101 ~Manoss ❤
Yeah I'm happy with that 10:37 AM

> Brandon Eddy
> We good with this topic then?

Yep 10:56 AM

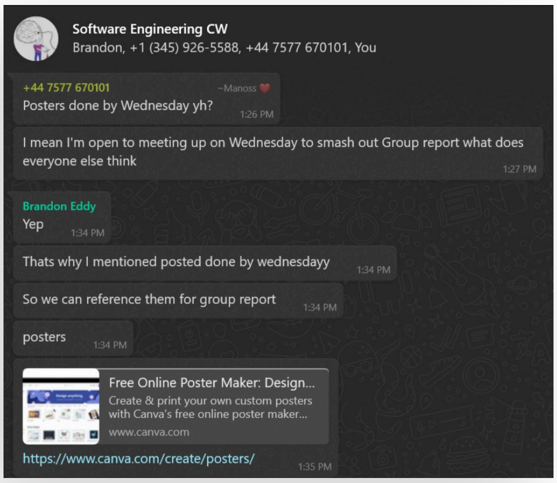
+1 (345) 926-5588 ~Jaxon Stewart
Fine with me 10:58 AM

Brandon Eddy
Hardest part was finding 3 papers 5:00 PM
Fuckin hell 5:00 PM
Found them tho once you find 3 it's cake 5:00 PM
I think ramzi said finding is the hardest part I guess 5:01 PM

2/3/2023

+1 (345) 926-5588 ~Jaxon Stewart
Makes sense ye I gotta find 2 more still 9:58 AM

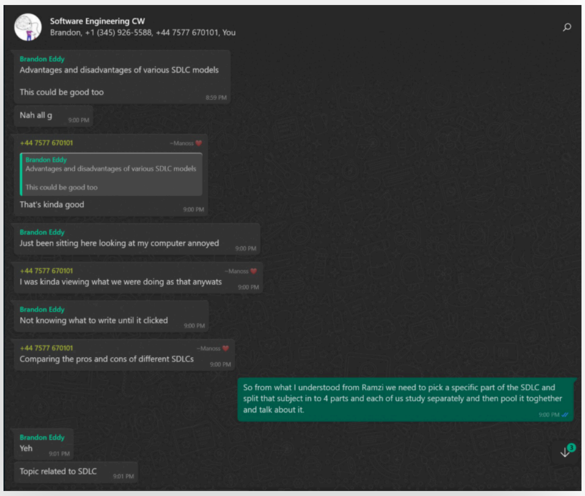
+44 7577 670101 ~Manoss ❤
Right so what's the plan once we've all got 3 papers 1:22 PM


Software Engineering CW
Brandon, +1 (345) 926-5588, +44 7577 670101, You

+44 7577 670101 ~Manoss ❤
Posters done by Wednesday yh? 1:26 PM
I mean I'm open to meeting up on Wednesday to smash out Group report what does everyone else think 1:27 PM

Brandon Eddy
Yep 1:34 PM
Thats why I mentioned posted done by wednesdayy 1:34 PM
So we can reference them for group report 1:34 PM
posters 1:34 PM

Free Online Poster Maker: Design...
Create & print your own custom posters with Canva's free online poster maker...
www.canva.com
https://www.canva.com/create/posters/ 1:35 PM


Software Engineering CW
Brandon, +1 (345) 926-5588, +44 7577 670101, You

Brandon Eddy
Advantages and disadvantages of various SDLC models

This could be good too 8:59 PM
Nah all g 9:00 PM

+44 7577 670101 ~Manoss ❤
> Brandon Eddy
> Advantages and disadvantages of various SDLC models
>
> This could be good too

That's kinda good 9:00 PM

Brandon Eddy
Just been sitting here looking at my computer annoyed 9:00 PM

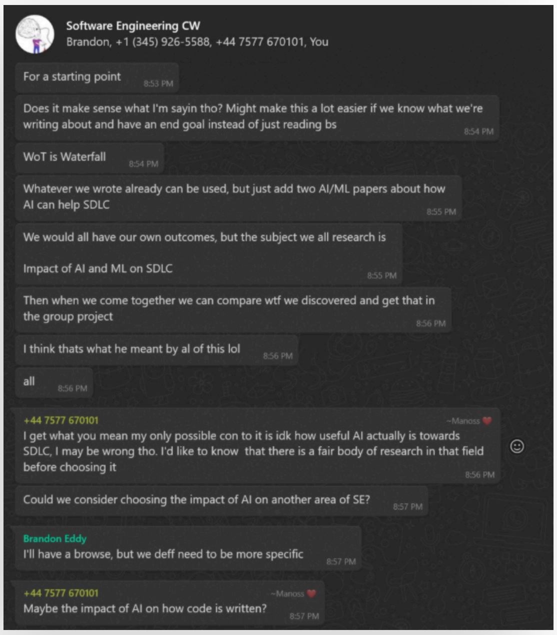
+44 7577 670101 ~Manoss ❤
I was kinda viewing what we were doing as that anywats 9:00 PM

Brandon Eddy
Not knowing what to write until it clicked 9:00 PM

+44 7577 670101 ~Manoss ❤
Comparing the pros and cons of different SDLCs 9:00 PM

So from what I understood from Ramzi we need to pick a specific part of the SDLC and split that subject in to 4 parts and each of us study separately and then pool it toghether and talk about it. 9:00 PM

Brandon Eddy
Yeh 9:01 PM
Topic related to SDLC 9:01 PM


Software Engineering CW
Brandon, +1 (345) 926-5588, +44 7577 670101, You

For a starting point 8:53 PM
Does it make sense what I'm sayin tho? Might make this a lot easier if we know what we're writing about and have an end goal instead of just reading bs 8:54 PM
WoT is Waterfall 8:54 PM
Whatever we wrote already can be used, but just add two AI/ML papers about how AI can help SDLC 8:55 PM
We would all have our own outcomes, but the subject we all research is

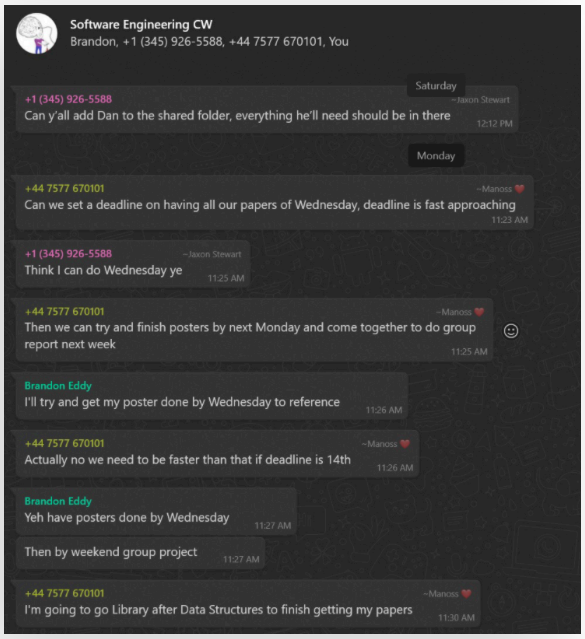
Impact of AI and ML on SDLC 8:55 PM
Then when we come together we can compare wtf we discovered and get that in the group project 8:56 PM
I think thats what he meant by al of this lol 8:56 PM
all 8:56 PM

+44 7577 670101 ~Manoss ❤
I get what you mean my only possible con to it is idk how useful AI actually is towards SDLC, I may be wrong tho. I'd like to know that there is a fair body of research in that field before choosing it 8:56 PM

Could we consider choosing the impact of AI on another area of SE? 8:57 PM

Brandon Eddy
I'll have a browse, but we deff need to be more specific 8:57 PM

+44 7577 670101 ~Manoss ❤
Maybe the impact of AI on how code is written? 8:57 PM


Software Engineering CW
Brandon, +1 (345) 926-5588, +44 7577 670101, You

Saturday

+1 (345) 926-5588 ~Jaxon Stewart
Can y'all add Dan to the shared folder, everything he'll need should be in there 12:12 PM

Monday

+44 7577 670101 ~Manoss ❤
Can we set a deadline on having all our papers of Wednesday, deadline is fast approaching 11:23 AM

+1 (345) 926-5588 ~Jaxon Stewart
Think I can do Wednesday ye 11:25 AM

+44 7577 670101 ~Manoss ❤
Then we can try and finish posters by next Monday and come together to do group report next week 11:25 AM

Brandon Eddy
I'll try and get my poster done by Wednesday to reference 11:26 AM

+44 7577 670101 ~Manoss ❤
Actually no we need to be faster than that if deadline is 14th 11:26 AM

Brandon Eddy
Yeh have posters done by Wednesday 11:27 AM
Then by weekend group project 11:27 AM

+44 7577 670101 ~Manoss ❤
I'm going to go Library after Data Structures to finish getting my papers 11:30 AM

7.0 - References

1. Boehm, B. W. (1988). A spiral model of software development and enhancement. Computer, 21(5), 61-72.
2. Digital.ai (2022) 16th Annual State of Agile Report, State of Agile. Available at: https://info.digital.ai/rs/981-LQX-968/images/AR-SA-2022-16th-Annual-State-Of-Agile-Report.pdf.
3. Dora, S. and Dubey, P. (2013) (PDF) SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) ANALYTICAL COMPARISON AND SURVEY ON TRADITIONAL AND AGILE METHODOLOGY, ResearchGate. Available at: https://www.researchgate.net/publication/319716548_SOFTWARE_DEVELOPMENT_LIFE_CYCLE_SDLC_ANALYTICAL_COMPARISON_AND_SURVEY_ON_TRADITIONAL_AND_AGILE_METHODOLOGY (Accessed: 1 February 2023).
4. Ihor (2021) The Agile Software Development Life Cycle: All You Need to Know, DistantJob - Remote Recruitment Agency. Available at: https://distantjob.com/blog/agile-software-development-life-cycle/.
5. Ruparelia, N. B. (2010) 'Software development lifecycle models', ACM SIGSOFT Software Engineering Notes, 35(3), p. 8. doi: https://doi.org/10.1145/1764810.1764814.
6. Shah, U.S. (2016). An Excursion to Software Development Life Cycle Models. ACM SIGSOFT Software Engineering Notes, 41(1), pp.1–6. doi:https://doi.org/10.1145/2853073.2853080.
7. Shah, U. S. (2016) 'An Excursion to Software Development Life Cycle Models', ACM SIGSOFT Software Engineering Notes, 41(1), pp. 1–6. doi: https://doi.org/10.1145/2853073.2853080.
8. Vijayasarathy, L. V. (2015) London Metropolitan University / All Locations, emu.londonmet.ac.uk. Edited by C. Butler20145. Available at: https://0-ieeexplore-ieee-org.emu.londonmet.ac.uk/document/7006383.