International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

Case Study: Comparing Different Software Engineering Models in Enhancing the Productivity of a Transportation Company

Krish Agarwal[1], Dhamini R Nijgal[2]

[1]Student, Department of Data Science, CHRIST (Deemed to be University) Pune, Lavasa
[2]Associate Professor, CHRIST (Deemed to be University) Pune, Lavasa

---

Abstract - This case study aims to investigate how various software engineering life cycle models can impact the productivity of a transportation company. It also examines how the transition from manual to automated processes can optimize the company's output in terms of cost, labour, and time. By exploring different software engineering models, this study seeks to identify the most effective approach for enhancing the transportation company's productivity. The study recognizes that while some software engineering models can lead to increased profitability, others may have limitations or flaws that need to be addressed. Ultimately, the goal of the study is to provide insights into the advantages and disadvantages of different software engineering models and their impact on the productivity of a transportation company.

***Key Words*:** Case study, Software Engineering life cycle model, Productivity, Optimizes, profit-maximizing tactics, Waterfall model, Incremental Model, Spiral Model, customer-end and driver-end application

1. INTRODUCTION

Businesses are always looking for methods to improve their production and efficiency in today's fast-paced world. This case study focuses on a transportation business that was having trouble meeting the rising expectations of its clients. The company throughout the years firmly believed in its manual procedure but factors like long-time waits, unprofessional behavior by drivers, ever rising traffic and customer dissatisfaction, etc. To overcome such challenges, the company approaches a software development company to find a viable digital solution. This case study's goals are to demonstrate the value of software engineering methodologies in improving business processes and to offer insights into the numerous software engineering models that may be used to address challenging business issues. This particular case study includes comparison and contrast between different software engineering models like Waterfall model, Incremental Model and Spiral Model, and analyzing their respective impacts on the productivity of the Transportation Company.

2. FOUNDATION

Let's establish a fictitious transportation company, DriveMate, founded in the year 2015, as the case study's starting point. It was a tiny firm that used manual processes at first, but as time went on and the company's client base increased, the manual processes were unable to handle the volume of business, manage the flow of money within the business for tax purposes, maintain the data regarding their drivers, etc. Increased customer dissatisfaction led to the loss of significant income-generating clients, which prompted the CEO of the company to contact a software development company, *NucleusTech: Software Solutions*, to help them digitize their work process and grow their clientele and revenue through data-driven decisions.

3. CHARACTERISTICS

NucleusTech: Software Solutions first gained knowledge of the company and its clientele. They carefully followed the processes of the software development paradigm, starting with requirement gathering. After gaining the essential business insight, the next step was deciding on the appearance and features of the application as well as writing down all the software (software languages) and hardware requirements. As mentioned in the problem statement, the company advanced with selecting a software engineering model that is suitable best for this case.

They narrowed their idea down to 2 faces, customer-end and driver-end applications.

Face 1: The customer can simply hit a button – "Ride" - and an available driver nearby will be assigned to their set location and drop them off to their desired location. The application will also offer basic information about the driver (contact details, name, rating), the route taken and the time of arrival, while keeping the traffic in account. Apart from an active trip, the application offers various other options like past trips, account information, application-based wallet for faster payments, etc.

Face 2: Any person with an existing driver's license can take up the post of a driver. Their credentials will be cross verified by the governmental database, and if it is approved, only then the driver will be taken into the company along with passing a systematic selection process. After the driver has been selected, they will be provided with the login credentials to the application. This side of the application provides the drivers a chance to select among the incoming pick-up requests from clients.

4. ANALYSIS

4.1 WATERFALL MODEL

The waterfall model is a breakdown of project activities into linear sequential phases, meaning they are passed down onto each other, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks [1].

The very first version of the customer-end application allowed the customers to book a ride at the provided location up to their desired set location. But what if a

customer changes their mind to either canceling or changing the final destination during a ride? The application doesn't support such a situation and to enable such a feature in the next version, the company will have to go through the entire process to fix the issue. If over time, bugs or necessary updates such as this arise very frequently, the maintenance cost behind the application will reach a high. But solving bugs in such a model increases the quality of the model.

4.2 INCREMENTAL MODEL

The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements [2].

As discussed in the above life cycle model, bugs constantly arise in an application but as the application proceeds with time and is in use, necessary advancements must be made. For example, in the driver-end application, the drivers should be allowed to take a second ride request as his/her active ride is about to end, but this should be kept in respect with the second customer, they shouldn't have to wait more than a reasonable and promised time. This gives both the company and the drivers to generate more revenue. These advancements/bug fixes come under a constant iterative approach, leading to increase in maintenance cost. But at the same time, the bugs can be rectified and fixed much faster than a waterfall model.

4.3 SPIRAL MODEL

The spiral model is a risk-driven software development process model. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping [3]. It contains majorly 4 phases: planning, risk analysis, engineering, and evaluation.
As discussed in the previous sections, bug fixes and necessary updates are crucial to maintain the quality and functionality of software. These issues should be addressed promptly to minimize their impact on the transportation company's operations. While such issues can be identified and tracked during the risk assessment phase of the software development life cycle, unforeseeable external factors like natural disasters, pandemics, and political instability can significantly impact a transportation company's productivity.

The recent global pandemic, for example, had a severe impact on the transportation industry, with companies experiencing a significant drop in demand and revenue. However, having a well-defined software engineering model in place can help transportation companies mitigate technical complications and adapt to new challenges more efficiently. By incorporating risk management strategies into the software development process, transportation companies can anticipate potential risks and develop effective contingency plans to minimize their impact.

In the case of the pandemic, companies that had implemented appropriate risk management measures were better equipped to manage the crisis. For example, companies that had invested in cloud-based technology and remote access solutions were able to maintain business continuity while adhering to social distancing guidelines. These companies were also able to quickly identify and address any technical issues that arose as a result of the sudden shift to remote work.

In conclusion, a sound software engineering model that emphasizes risk management can help transportation companies enhance their productivity and adapt to new challenges more effectively. While external factors like pandemics may be unpredictable, having a well-defined risk management strategy in place can help companies weather the storm and emerge stronger in the long run.[4]

5. Conclusion

In conclusion, after comparing different software engineering models, the incremental model emerged as the most suitable for enhancing the productivity of a transportation company. This model allowed for flexibility and adaptability throughout the development process, which was necessary for a dynamic industry such as transportation. Additionally, the incremental model allowed for continuous testing and feedback, resulting in better quality products and faster time to market. While the other models, such as the waterfall and spiral models, have their strengths, the incremental model proved to be the best fit for this particular case study. Ultimately, by adopting the incremental model, the transportation company was able to achieve its goals of increased productivity and improved customer satisfaction.

REFERENCES

1. Wikipedia, Waterfall model, Wikipedia, 2023
2. Wikipedia, Incremental build model, Wikipedia, 2023
3. Wikipedia, Incremental build model, Wikipedia, 2022
4. Mishra, A. & Dubey, D. (2013). A Comparative Study of Different Software: Development Life Cycle Models in Different Scenarios. International Journal of Advance Research in Computer Science and Management Studies. 1(5) 64-69