

Digital Product Management: Ensuring Value for a Digital Transformation

CS4991 Capstone Report, 2022

Anderson Proescholdbell
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
akp4ujp@virginia.edu

ABSTRACT

A large hospitality company sought to rejuvenate its business by modernizing its antiquated processes with a digital product. Within a software development team covering multiple disciplines, I worked alongside a senior product manager to facilitate the agile development process and ensure that the product would be valuable to consumers. As a product manager, my role existed at the intersection of desirability, feasibility, and viability. Thus, I had to define what users really desired and make sure it was both viable from a business perspective and feasible from an engineering standpoint, within the constraints of working on a team. I worked with the team to successfully develop increments of the digital product and release them into production. The expectations of stakeholders were fulfilled as exemplified by the client communicating being highly satisfied with the end product, user testing showing their customers were also pleased with the result, and the internal team completing agile retrospectives and viewing the project positively. Further work will be required in the future to iterate on and improve the software as well as fix bugs and pay off technical debts.

1 INTRODUCTION

In 2016 the total revenue of the software market in the United States alone is estimated to have been \$211 billion [1]. Just five years later, the total revenue in 2021 is estimated to have been \$284 billion [1]. The software industry in the US alone in a five-year span has grown by an estimated 135%. This shows the substantial influx of new digital products that are being developed and released. Each of these products seeks to differentiate itself from existing products by providing value that other products do not. With so

much money being pumped into this relatively new industry, it is no surprise that many companies are looking to software to help grow their business.

One subset of these products are novel ideas that have no analog correspondents. Another subset is products which attempt to modernize existing products or processes. The project I worked on falls into this second group. My project sought to replace the process of booking reservations over the phone with a software experience. With the original process driving revenue for so many years, the new product had expectations that it had to surpass. A substantial investment was made to recreate and more importantly reimagine the process through the lens of software. Further investment was contingent on the success of the new software product defined by reducing the need for phone operators, increasing customer retention through the reservation flow, and consequently increasing total revenue.

2 RELATED WORKS

Rigby, et. al. (2016) discuss the growing adoption of Agile methodologies within the software industry since the publication of the Agile manifesto in 2001. The management of this project was administered under Agile methodologies, specifically following Kanban principles. Lei et. al. (2017) provided an analysis of how such Agile methodologies could discernably improve project efficiency. Specific processes and strategies within Agile methodologies pertain majorly to the product manager of the team. These processes are exhibited through various adaptations of The Product Strategy Cycle as outlined by Pichler (2022).

Last, it is important to discuss the technologies used to bring the project to life. While a multitude

of different applications and tools were used to develop this project's software components, the two main technologies utilized for this project were Node JavaScript and React as a framework. These two technologies have been explored extensively by Tilkov and Vinoski (2010) and Gackenhaimer (2015), respectively.

3 PROCESS DESIGN

This experience revolved around the matter of delivering a product to a client. My mentor and I acted as the product managers of this project. We were involved from conception through completion. The engagement with the client can be defined as four mostly-distinct phases.

3.1 Understanding The Business Problem

3.1.1 Conversations.

Perhaps the most important phase of an engagement with a client is understanding the problem they wish to solve. If our team, which was being paid to create an exemplary solution for the client, did not fully understand the problem, there would be no basis upon which to create a superb solution. Consequently, this phase entailed numerous detailed discussions with the client. The conversations took place remotely over Microsoft Teams since the client company was based in another region of the US.

3.1.2 Problem Categories.

The client had identified recognized(?) realized(?) found(?) that their system of managing reservations was holding their company back. With the existing system, the only option customers had to create reservations for property was to phone. There are three main issues with this approach.

3.1.3 Problem 1.

Talking over the phone is a major inconvenience for many people. Consequently, it was likely that the company was losing reservations or facing high rates of attrition due to the unappealing phone process being the only option to place reservations.

3.1.4 Problem 2.

Placing reservations over the phone is tricky for both the customer and the operator. Conveying necessary information to a customer vocally takes more time compared to conveying information visually. In addition, the phone operator must have

the required information accessible which is not instantaneous. Additionally, a customer might require time to deliberate their reservation decision. A customer doing so has no way of reentering the phone system reservation process in the same position they left off and must restart. Furthermore, giving credit card or banking information over the phone is inconvenient and insecure.

3.1.5 Problem 3.

The company must maintain a substantial workforce of phone operators trained for both using systems and understanding complex business logic involved for the hospitality company.

3.1.6 Problem Conclusion.

The considerable inconveniences and expenses derived from the three aforementioned categories were the driving issue the company was facing and required help with.

3.2 Defining Solution, Scope, & Technical Implementation

3.2.1 Solution.

Once we knew the problem our client faced, we had to design a solution. Based on our previous conversations with client stakeholders and their request for the creation of a website, it was fairly straightforward to agree with them that a website would be the digital transformation they required.

3.2.2 Scope.

We then referred to all functions the client's phone process currently possessed to prioritize our development. We created an MVP, or minimum viable product, based on current functionality and proposed additions. This process involved prolonged deliberation from my mentor and me, with consultation from a solutions architect. Budget and manpower constraints were also considered.

3.2.3 Technical Implementation.

Part of determining scope relied heavily on consulting with software engineers. We had to determine what technology we were going to use and how long we estimated it would take in order to complete our scope plan. Upon detailed discussions with engineering, we decided upon a website built upon the React framework coded in

JavaScript [7, 6]. Both are widely used for web development and can expect feature improvements and security fixes for the foreseeable future.

3.2.4 Higher Definition.

Once we had a somewhat detailed scope plan, we had to provide more detail in all areas. While the goal of Agile development is to not over-plan, for our project to be a success we needed to have enough planned out to be able to hit the ground running for development [3]. Creating this higher definition scope plan required the practice of requirement elicitation [5]. At a high level, requirement elicitation is facilitating a discussion with a stakeholder to more fully determine what a feature should look like. For instance, if we were planning on developing a way for users to authenticate themselves before entering a reservation flow, we would approach the client with questions such as: “How do customers currently verify themselves?” and “What data do we currently have on customers with respect to authentication?” This allowed us to flesh out features in greater detail.

3.3 Facilitating Development

The methodology of Agile development our project chose to employ was Kanban, which involves normal Agile rituals, but forgoes assigning story points to development tickets. The rituals and principles my mentor and I conducted or instituted for this project are as follows.

3.3.1 Standups.

Standups, as practiced on our project were a time every morning for the whole team to align. We would align on who was working on what, what had been completed, and who needed help. Having these regular meetings served to maintain team clarity and understanding of the project’s standings and was a good time to ask questions or make announcements.

3.3.2 Ensemble Programming.

Ensemble Programming is the technique in which software engineers work in groups rather than individually. On our project, we would often have teams consisting of two software engineers, one software test engineer, and sometimes also a product designer. The thought behind this is that developing in a group allows all individuals to

contribute their unique talents as well as create a more streamlined thought process for the product. In addition, this reduces knowledge transfer overhead between parties. Since multiple roles are all working together, they do not have to pass off work to each other for next steps and communicate extensively.

3.3.3 Constant Iteration.

Constant iteration is a description of always working on new releases. The way this manifested in my project was by our team aiming to push out new features as often as possible. If we did our jobs well, an end user would see small changes take place so gradually that they would not even notice the product changing right beneath them.

3.3.4 Regular Feedback.

While not an official principle of Agile, getting regular feedback from the client was critical for maintaining proper course and creating the right product. We were lucky enough to have stakeholders join our standups every morning, ready to critique our work so that we could make changes and improve it. Constantly getting feedback meant that we worked more leanly and did not waste development time on unwanted aspects.

3.3.5 Discovery.

Discovery is just as the name implies; our team has discovered something new and we need to deal with it. Discovery in our project took place frequently when dealing with complex business logic. For instance, we might have been developing a reservation flow, thinking it works the same for all users, when suddenly a stakeholder casually mentions that only one type of user can enter this particular flow. That is an extreme example, but when dealing with clients and Agile methodology in general, changes are to be expected.

3.3.6 Retrospectives.

Another important ritual we followed was looking back retrospectively on how we worked over certain time periods. This meta-analysis allowed us to regularly adjust how we were working to increase our performance and overall satisfaction.

3.3 Delivery

The delivery of our product was largely incremental as described above. The way this would occur on our project was to have a development environment, user acceptance testing environment (UAT environment) and production environment. All work, of course, took place on the development environment. Once the work had been completed, it would be pushed to the UAT environment where both internal and client testers would ensure the quality of the code. From there, features would be incrementally released to the public per approval from the client.

4 RESULTS

The results for this project are more qualitative than quantitative. Once we delivered the final product and the contract with this client company ended, we no longer had access to figures such as user growth, effectiveness of the software, etc. However, we can conclude with a very high degree of confidence that the product met or exceeded expectations. Client stakeholders verbally expressed their content with the end product.

Furthermore, by all logic the product should succeed. This is a highly precedented path for a company to take to optimize their business as well as the fact that the phone system will still be available for customers who do not want to change their ways. Thus, as the product was released to the public some time ago and is still available on top of stakeholders directly expressing their content with it, the result of this project was a very successful product release.

5 CONCLUSION

The practices outlined are pertinent to all teams in software development. Software is only profitable if it is useful. Thus, no matter how technically sound a piece of software is, it is worthless if it is not of value to at least one individual or group. If the development of software is being explicitly paid for by an individual or group and they have given specific requirements for that software, it is perhaps even more crucial to confirm that the requirements will result in meaningful software.

The project described above that was undertaken by my internship team is a useful case to examine. The processes our team followed allowed for us to develop software we knew would be valuable to the

client and consequently please all involved stakeholders. Similar methods should be exercised by teams looking to create valuable software.

6 FUTURE WORK

To expand upon the listed practices, it would be beneficial to break them down into more concrete and detailed steps. While the methods my team used are discussed thoroughly, it could be difficult for a team to execute them without having prior experience. Additionally, conducting more research on the practices could determine if they have been optimally constructed or if there are better ways to accomplish the same goals.

REFERENCES

- [1] Statista. Software – United States | Statista Market forecast. Retrieved September 22, 2022 from <https://www.statista.com/outlook/tmo/software/united-states>
- [2] Agile Manigesto. Manifesto for Agile Software Development. Retrieved September 22, 2022 from <https://agilemanifesto.org/principles.html>
- [3] Rigby, D., Sutherland, J. and Takeuchi, H.. 2016. Embracing Agile: How to master the process that's transforming management . (May 2016). Retrieved September 22, 2022 from <https://hbr.org/2016/05/embracing-agile>
- [4] Lei, H., Ganjeizadeh, F., Jayachandran, P. K. and Ozcan, P. 2017. A statistical analysis of the effects of Scrum and Kanban on software development projects. Science Direct 43, (February 2017), 59-67. DOI: <https://doi.org/10.1016/j.rcim.2015.12.001>
- [5] Pichler, R. 2022. *Strategize: Product Strategy and Product Roadmap Practices for the Digital Age*. Pichler Consulting, Wendover, Buckinghamshire, United Kingdom.
- [6] Tilkov, S. and Vinoski, S. 2010. Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing 14, 6 (Dec 2010), 80-83. DOI: <https://doi.org/10.1109/MIC.2010.145>
- [7] Gackenheimer, C. 2015. What Is React?. Apress. DOI: https://doi.org/10.1007/978-1-4842-1245-5_1