Quaestum

DOI: https://doi.org/10.22167/2675-441X-20210561
Quaestum 2021; 2 e26750561
ISSN: 2675-441X

Article

Product management and the construction of the required software: the case of the Commercial Agreements module

Gestão do produto e construção do software necessário: o caso do módulo Acordos Comerciais

Livyson Saymon Leão Azevêdo[1*]

Received: oct 26, 2020
Accepted: dec 22, 2020

[1*]Universidade de São Paulo – Postgraduate student of the MBA in Project Management – ESALQ. Dr. Alfredo Ellis Street, n 210. APT 103 – Bela Vista, 013220-050 - São Paulo, SP, Brazil. E-mail: livyson@gmail.com

Abstract: This article describes an applied research conducted in a technology company. The project addresses the adoption of product management within the context of software engineering, analyzing changes in environment and work dynamics resulting from this adoption. The goal was to understand how the management of software products in the Agreements Module is transforming the software engineering of a technology company, working on the uncertainties pertinent to the product at the implementation phase, avoiding the mismatch of information that ends up causing the non-delivery of the products, failing customer requirements, and creating unrealistic schedules and budgets. The purpose of this work was to contribute to the area of technology by producing content for a branch not yet explored in academia, as it is experiencing great growth in the labor market. The cohort of this research was ten professionals from Agreements squad: five developers, one design analyst, one quality analyst, one product manager, one agile coach and one coordinator. Through surveys applied to the end user and to the team involved at the development, it was possible to assess the level of satisfaction within the process and the product delivered. Throughout the article, it is possible to observe the importance of product management in the production of software; as when implemented in a planned and integrated way, it can bring profits and avoid rework.

Keywords: innovation; software engineering; technology management; web product.

Resumo: Este artigo descreve uma pesquisa aplicada realizada em uma empresa de tecnologia. O projeto aborda a adoção da gestão de produtos no contexto da engenharia de software, analisando as mudanças no ambiente e na dinâmica de trabalho decorrentes dessa adoção. O objetivo foi entender como a gestão de produtos de software no Módulo de Contratos está transformando a engenharia de software de uma empresa de tecnologia, trabalhando as incertezas pertinentes ao produto na fase de implementação, evitando o descasamento de informações que acaba gerando o não entrega dos produtos, falha nos requisitos do cliente e criação de cronogramas e orçamentos irrealistas. O objetivo deste trabalho foi contribuir com a área de tecnologia, produzindo conteúdo para um ramo ainda não explorado na academia, por se encontrar em grande crescimento no mercado de trabalho. O coorte desta pesquisa foi de dez profissionais do plantel de Agreements: cinco desenvolvedores, um analista de design, um analista de qualidade, um gerente de produto, um coach ágil e um coordenador. Por meio de pesquisas aplicadas ao usuário final e à equipe envolvida no desenvolvimento, foi possível avaliar o nível de satisfação com o processo e com o produto entregue. Ao longo do artigo, é possível observar a importância da gestão de produtos na produção de software; pois quando implementado de forma planejada e integrada, pode trazer lucros e evitar retrabalho.

Palavras-chave: inovação; Engenharia de software; gerenciamento de tecnologia; produto da web.

Introduction

Since 1940, when systems were still programmed in binary code, developers have faced difficulties about implementing solutions that meet the real needs of the time. Still in large computers (mainframes), the obstacles to the solution of complex issues such as those pertaining to ballistics – which were of fundamental importance during the period of World War II – generated a high cost for the American government, since maintaining all the machinery required a large amount of valves and energy. At that historical moment, the need to understand what the American army wanted was already noticeable: many of the achievements of scientists who worked on breaking codes were based on monitoring the movements of opposing troops.

In the 1960s, traditional computers managed to advance to a new stage where multitasking was performed by a single computer. In the following decade, a single computer could already be accessed by multiple users. The new equipment demanded less and less human interference, gained a graphical interface and the ability to create programmed routines. With the arrival of the Uniplexed Computing and Information Service (UNIX) operating system, the adoption of technologies by universities and companies has become essential. According to Bezerra (2017), in the 70s, the Unix system was the most used, adopted by the government, universities and private companies. Still according to the author, in the same decade, emerged ARPANet (the embryo of the future internet), Altair and Mark8, (the first computers for home use, called microcomputers).

With the accession to new computers by several enterprises, a search for information decentralization began in the 1980s, which would generate a boost for the 1990s, known as the decade of technological integration, where several business models needed to be renewed and reintegrated. The same technology that came to companies to facilitate information exchange and optimize processes required a high investment in technological resources, customized software and employee training. Oliveira and Santos (1998) reinforce that there was a cheaper and easier access to equipment, consequently, there was an increase in the amount of information available through these channels, since the new technologies of computers and telecommunications along with the use of databases in line helped the transformations that took place in social structures.

With all these advances, the equipment, in addition to being cheap, has become easy to manipulate. Attentive to this scenario, several companies emerged to insert technological solutions in the most diverse branches: health, education, financial, legal, entertainment, etc.

The new companies needed to serve their customers, helping them to understand what they wanted. The software factories (companies that have technological apparatus, processes and human resources focused on the delivery of computer systems), as they learned about technology, also tried to understand how the market where they operated worked. Thus, there were many projects canceled, softwares discontinued or poorly built, and customers frustrated. It is pointed out by Leopoldino and Borensteins (2011) that the uncertainties pertinent to the development of software end up affecting the results of the projects, since there are several reports of projects that are out of specifications, are delivered after the deadline, exceeded the estimated costs or were simply canceled, consuming a large amount of resources without obtaining the desired results.

The new era of computing, which began in the 2000s with the advent of mobile peripherals (cell phones, tablets, smartwatches) demanding professionals specialized in artificial intelligence, made companies invest in new ways to manage the needs of its customers, thus creating what we now know as Product Management.

Product Management arrives with the objective of ensuring that the customer's needs are really met, ensuring that the best practices necessary for the construction of the right software are applied. The right software is one that meets the real needs presented by users. Product Management starts to treat software as a product, and for that reason, it is involved in its entire life cycle: from the beginning (introduction) to its end (decline). Unlike the requirements engineer, the activities of the product manager are performed during the development process and not just at the beginning of the requirements writing.

Torres (2015) describes Software Product Management as responsible for all aspects of a product throughout its life cycle. Still according to the author, one of the main functions of this area is to connect the company's strategy to customers' problems and needs, generating a solution through technology.

For this reason, in order to understand how the study subject organization (a Brazilian multinational company that offers digital solutions for process management in hospitals, clinics, manufacturers and distributors of medicines) acted in the process of developing software, meeting the needs of customers, reducing costs and innovating, this research is based on the following question: how did the Product Management of the Agreements Module act in its construction process?

The understanding of Product Management Software in the Agreements Module is transforming the company's software engineering, working on the uncertainties pertinent to the product in the implementation phase, avoiding the mismatch of information that ends up causing the non-delivery of customer requirements and creating unrealistic schedules and budgets. The goal is to contribute to the area of technology by producing content for a branch that is not yet explored in academia and is experiencing great growth in the labor market.

Among the specific goals are: to understand how the knowledge in the area of Product Management was applied to the development of the product Agreements; identify changes in requirements gathering with the adhesion of the Product area; point out the main characteristics of the product manager; verify how the Management of Software Products is helping the company to innovate, reducing costs; and check the customer satisfaction measurement process.

Material and methods

It was noticeable the lack of content aimed to the Software Product area, at the same time that there was a notorious growth, although slow, of authors writing about the area, according to Torres (2015). In the last 10 years, with the change in the mindset of large software companies, with the rapid opening of countless technology startups and increased competition in several sectors, software development processes have joined the product management in order to optimize processes in technology companies, according to Fridschtein (2005).

Therefore, this applied research had a qualitative approach, since it used observations, questionnaires and unstructured interviews. Thus, this work had as procedures the case study and the field research in a software developing company.

The chosen company was a Brazilian multinational that provides technological solutions for process management in hospitals and clinics, connecting these first to distributors and manufacturers of medicines, medical materials and supplies. The company, which is located in the city of São Paulo, was the pioneer and, until today, is the market leader in the supply of this type of software. Agreements was analyzed, which is a product immersed in Regene, a purchasing management solution platform for private institutions, where buyers (hospitals and clinics) can request budgets, quotes and establish purchase contracts with suppliers, which, in turn, offer proposals for their products available in the catalog. The platform use provides a clear reduction in costs and optimization of operational routines, serving as a communication interface so that buyers have access to the entire industry and its main distributors. Agreements, on the other hand, is a module of this platform and allows contracts between suppliers and buyers to be established, defining conditions for the supply of medical materials, medicines and other categories of supplies for the health segment. It is necessary to understand that, before the development of this module on the Regene platform, the company already made this feature available on an old platform, whose development was delayed due to outdated technologies. Therefore, its users who needed this module were unable to migrate to the new platform.

To conduct this project, some tools, methodologies and good practices were used, among them: OKR, a methodology developed and popularly used in technology companies as a way to simplify the understanding of the organization's strategic objectives. Daher (2020) references this methodology as a proposal to define strategies and objectives for a short period so that they provide a reference to evaluate the performance of the implemented strategy. Also was used Scrum, the most popular agile project management framework. Layton and Morrow (2019) endorse that Scrum is a framework with the best results, demonstrably bringing up to 40% reduction in the time to launch any product on the market. Still according to the author, for greater assertiveness, Scrum adheres to some important ceremonies such as sprint planning, where the scope to be developed during a sprint (the work cycle, or the time the team has to complete the activities estimated during the sprint planning) is defined. To ensure the progress of activities, a daily ceremony is used, called a daily meeting, where all team members bring a summary of the progress of their activities. Kanban was used to facilitate the visual management of these activities, making their progress more transparent. Anderson (2010) describes that Kanban is becoming a popular way to visualize and limit the work in progress in the development of software and in the work of information technology. Another practice was design thinking, a technique of bringing together several people in order to approach problems and propose solutions through creative dynamics. According to Liedtka and Ogilvie (2019), design thinking is a systematic approach to problem solving. Roadmap was adopted too, a graphic map that represents on a timeline where the deliveries of activities in which the team is working are concentrated. Lombardo et al. (2017) describe that the roadmap is one of the most important and influential documents that an organization can develop to help it comply with the company's strategy. Another one was story mapping, a technique for describing the product to be developed in short stories. Patton (2014) shows that the story mapping tells the path the user takes to reach the goal while using the product. Finally, Backlog refinement was a ceremony to review the product's backlog in order to ensure that it contains the appropriate and duly prioritized items. According to Malhotra (2020), the backlog refinement makes the process more accurate and guarantees delivery with more quality.

Results and discussion

In this perspective, at first, after the formation of the software development team, emphasis was placed on the requirements demanded by the client, giving more clarity to the way the module should be created. In view of what has been discussed so far and taking into account what Vazquez and Simões (2016) point out, it can be said that requirements engineering is linked to all stages of software development.

Despite having been solidified for years in companies, requirements engineering has changed within the context of Product Management, leaving the scene the engineer responsible for raising requirements at the beginning of the project, giving place to the product team responsible for absorbing all the demands of the requirements engineer, also acting in other parallel activities, keeping such requirements up to date and, mainly, being responsible (i.e. owner) by the product throughout the construction phase.

The product leader (i.e. Product Director) met with directors and strategic clients to understand what needs they had for the Agreements product. Once expectations were defined, the needs were translated into objectives and key results (OKR) – methodology for managing the main objectives of a corporation in a period. According to Mello (2019), OKR is a tool that tends to guide companies in their business strategies, through the implementation and execution of objectives easily interpreted and adopted by the company's employees. Still according to the author, this methodology becomes interesting for modern companies because it is completely associated with the agility movement of software development, which aims to make the company's future vision more transparent.

In this case, the institution had the following demands: objective (O - objectives): increase the number of users migrated to the Regene platform; main results (KR - key results): migrate all users who use the already outdated Classic system to the new platform – an updated and more modern system that includes features from the old system revamped.

As can be seen, OKRs were objectives that guided the company, but they were not enough to determine what a development team should implement. Therefore, an initial conversation with the application users was important to understand their needs. Machado (2016) points out that, today, it is noticeable that the requirements elicitation process is as important as the documentation process, since it requires an ability to involve the development team, product managers and others interested in the product (i.e. stakeholders). Therefore, it was essential to recognize the problem that would be solved by product development. To address the efficiency of this practice in the research project, a form was sent to the participants of this initial conversation, questioning the efficiency of this dialogue, and asking what are the advantages and disadvantages of this process of extracting information from the customers.

The product manager needed to consolidate the information obtained in user stories–specification of the system user's daily life through a business language, which would serve as a reference for a functionality construction. Current companies that adopt agile methodologies do not adopt the creation of requirements as in the old ways, documenting and rarely updating them. Instead of it, a set of user stories about the product being developed is chosen. According to Rose (2017), the user story is a story told about the user's routine when using the product, and needs to answer the following questions: who is requesting it? What do they want? For what purpose? To assess how effective the adoption of this feature was in the development process, an unstructured interview was conducted with the system analyzers (developers).

When functionalities were in the process of quality analysis, software tests and validation, it was analyzed if the test scenarios were impacted by the adoption of user stories and what were the facilities and difficulties that quality analysts found. For this analysis, this research gathered data collected through the participant observation method, sharing the daily work of these professionals. According to Correia (1999), the participant observation technique consists of having direct and frequent contact with the researcher, within their cultural context, with his performance being a research instrument. Still according to the author, within these premises and restrictions of technique application, it is possible to eliminate subjective deformations so that there is an understanding of facts and interactions among subjects under observation in their context.

Throughout the product construction process, ceremonies that took place to define deliveries of the week were observed: planning week (i.e. sprint planning), where activities (i.e. tasks) developed during the week were planned, refined and elaborated; daily follow-up meeting (i.e. daily sprint), ceremony in which all members of the development team presented what they had done the previous day, talked about the activity in which they were working and if there was any impediment in the development of their functions; delivery review meeting (i.e. sprint reviews), ceremony to review the week's deliveries, when all the features delivered during the week were presented to the entire development team, the product manager and the other interested parties; and, finally, the retrospective meeting (i.e. sprint retrospective), ceremony in which the whole team came together to reflect on the sprint (period of time to complete a set of planned tasks) which was ending, with the objective of identifying strengths that could continue to be reproduced in following sprints, and also gaps so that an improvement plan could be created and executed in the next one. According to Iarandine (2019), there are many ways to perform the typical ceremonies of agile methodologies, most born from the Scrum framework. Still in accordance with the author, there is no need to be precise in the first versions of these ceremonies, since the team is learning to deal with them. The important issue is that the participants are able, little by little, to see the value of these meetings, so that, even without the presence of the Agile Coach (professional responsible for ensuring the agile process) they are able to prioritize and conduct the meetings.

Finally, when the first features of the Agreements Module were implemented and validated by the Quality Team, it was incumbent, before starting the implementation, to perform a usability test with some strategic users, who were invited by the company to use the module, in order to provide their impressions (i.e. feedbacks) regarding what was implemented up to that moment. Norlin and Winters (2002) reinforce that usability tests increase the value of product, as well as affirm the integrity of that organization.

New technological scenario

For many years, the requirements engineer was, within the development team, responsible for the communication interface with the customer. This professional was responsible for talking to the client and interpreting his needs for the development team. One of the problems with this dynamic is that the requirements engineer usually only communicates with the client and tries, through oral communication, to transcribe what the client needs, even when the client himself does not know exactly what he needs. As the methodology adopted is sequential and the customer's needs may change since the market is dynamic, once the requirements are defined, it is no longer possible to go back to a possible phase of change. The adoption of this model makes the requirements, necessarily, clear at the beginning of the project, and as it is already well known in the market, it is common for knowledge of requirements to be acquired as the software is being implemented. Sganderla et al. (2016) describe the cascade model as a model where each phase should only be started after the completion of the previous one, where it is necessary to approve at least one or more documents on the completed phase. Despite the mature completion of each phase, the authors also reinforce that the costs of producing and approving documents make iterations costly and end up involving significant rework, as the project only goes through this cycle once. Guided by this reasoning, it is possible to understand that it is a model that should be used in projects where the requirements were well understood.

Due to the changes the market began to present in the new technological age begun in the 2000s, when the systems needed to be delivered more quickly, it became difficult to continue using the Cascade Model. From this need agile methodologies are born, whose objective is to allow software development teams to better adapt to the changes that may occur in the project. Within this context, it is possible to deliver the functionalities of a project, present them to the client and ensure that possible inconsistencies will be reviewed and changed quickly. Eloi (2016) argues that agile methodologies are ideal for projects where requirements are frequently changed or undergo many changes, and therefore, projects that use such methodologies end up being supported by small development teams.

The company

The company was created in the city of São Paulo – SP in the year 2000, where it grew up developing technological solutions that automate the daily processes of hospitals, clinics, hospital networks, health plans, manufacturers and distributors of medications. Gradually, the company was also adapting its products to serve other health markets, reaching countries such as Argentina, Colombia, Spain and Mexico.

Already consolidated as a Brazilian multinational, pioneer and leader in the market in which it operates, under the direction of its founder Mauricio de Lázzari, the organization in 2018 joined a sovereign investment fund, one of the largest in Singapore, which ended up helping to company in the expansion of its business and the allocation of investments in technologies, with only R$ 30 million invested in 2019 for the acquisition of equipment, formation of new technology squads, improvement of platforms and development of new technologies.

Visionary, in the same year, the multinational acquired the Startup Temasek Holdings, with the objective of investing in internet of things (IoT).

Classic: a revenue-generating platform, but in the process of being discontinued

Classic was one of the first products of company, developed in the early 2000s with the aim of leveraging the name of the institution and putting the company in contention for this connecting market between buyers and suppliers. Today, it is where more than 80% of the company's users are registered, in addition to being the most used platform in the segment.

Of the various difficulties encountered by health systems, the poor management of medication purchases and stocks is the most prominent. Thus, Classic proposed to solve it. Seventeen hundred hospitals and more than ten thousand suppliers use the platform for their transactions, mobilizing more than twelve billion reais per year.

Through the hospital supplies marketplace it is possible for hospitals no longer need to budget purchases of medicines, medical supplies and other supplies individually with each different supplier, which previously generated exhaustive, manual, limited and less effective communication work. Using Classic, hospitals are able to open quotes that are available to thousands of suppliers registered on the platform, allowing hospitals to cover a greater number of suppliers and, consequently, increasing the chances of closing deals at better market prices. Suppliers, in turn, end up having a larger catalog of customers who are registered in the system.

In addition to this flagship, the marketplace has an extensive list of important functionalities, which gradually became part of the transaction process of these hospitals and suppliers, such as, for example, contract purchases, intelligent reports for easy management of processes involving supplies, renegotiation of item quotation, permission management and electronic approval limits, etc.

Even with this vast catalog, the platform has technological limitations which impede the growth of the organization. The entire platform and its infrastructure and hosting ecosystem were built at a rapid growth rate, using technologies that are already out of date and do not integrate with new systems developed by the company. Any development of new functionalities or specific change in the platform ends up demanding more time due to the high learning curve both in understanding business rules and in the code implemented.

After some consultancy and in-depth studies, the company decided to no longer invest in new features within Classic, making the decision to develop a new marketplace using new technology and more modern integrations, and gradually migrate users as the functionalities they use in Classic are developed on the new Regene platform.

Despite the decision to discontinue, the company has an exclusive development and support team for customers who are still using the old platform, providing improvements in existing features, error corrections and specialized service.

Regene: modern, adherent, satisfactory, but still under construction

Regene is the institution’s new marketplace, the result of the decision to discontinue Classic. Gradually, the new platform has been gaining functionalities featured on the old platform, which allows the migration of users who end up having no more reasons to not migrate. Today, the big challenge is migration, since some features are not implemented in the new platform, preventing users from following their workflows normally.

The system was developed using some more modern technologies, such as Ruby, a language closer to human language, with easy to understand semantics and flexible construction, facilitating its customization. With several reusable components, it reduces the development time and the number of implementation decisions through the adoption of “gems” - package managers that provide programs, functionalities and libraries that can be inserted and integrated within systems which use Ruby programming language. A stable language allows greater security, either through code review, or through the automation of unit tests and integration.

The entire interface part (i.e. front-end) is developed using html, css, javascript and React, which means that the platform can undergo changes without major impacts and quickly, since the user’s behavior can be constantly changed .

The new organization investments are being focused on the implementation of these functionalities through the extension of its staff, reformulation of the technology area, the hiring of more specialist professionals and the formation of new development teams. 2019 was a year of migration, whose expansion was a target for 2020, in addition to some other strategic goals. For these clients to be migrated, the implementation of the product Agreements is of fundamental importance.

The product Agreements

Agreements is a module responsible for ensuring that once medicines, medical materials or supplies price quote is agreed, buyer and supplier can have these items available for the agreed price for a long period of time, signing a kind of digital contract, where it is also possible to manage the quantity of items purchased and order, partially or completely, the contracted stock.

Using this module the buyer, in addition to closing the digital contract with his concordant suppliers (those with the best price proposals), is able to manage for his establishments the stock of his requested items, not necessarily receiving them all at once and needing to stock them, as orders can be generated as these items become necessary.

Among other possibilities are: the renegotiation of agreements already contracted; communication via platform to renegotiate agreements rejected; management of agreements still in the negotiation process, current ones, with items available for requisition and closed ones.

The Agreements team

The Agreements module consists of a working group called squad – a term created by the Spotify company to designate vertically oriented groups that are responsible for implementing specific products of the organization. The squad responsible for developing the Agreements module has ten people: five developers, a quality analyst, a design analyst, a product manager, an agile coach and a technology coordinator.

Product design

The initial step of creating a product is to build your vision of this object that has not yet been implemented. There are no judgments here, everyone involved can and should bring their vision of what will be implemented. At this stage it is necessary to validate the ideas with users, customers, directors, product manager and representatives of the technology team that will develop it. In this phase, the team experienced the following processes: definition of strategic objectives, design thinking, feedback from users and customer representatives, and vision of the product (i.e. kickoff).

Defining strategic objectives

The product design part was the most diligent part of the project, since the functionality already existed on the Classic platform and, in the board's conception, it would only need to exist on the new platform. It was then up to the executive directors to present to the product manager the importance that this implementation would bring to the organization by the number of users migrated, not getting involved in the way that this functionality would be implemented. The organization's goal was translated into numbers migrated to the new platform, and this was the guideline given to the product team, which in turn would unravel this goal on the design of the desired technological product.

Design thinking

The set of ideas and intuitions called design thinking consisted of bringing together several people to participate in the process of building a solution, having different angles and different panoramas for solving a problem. The approach made people experience ways of thinking through design, developing empathy by putting themselves in the place of the user affected by the problem to be treated and generating a feeling of belonging to the project by the people involved in the possible proposed solutions. According to Simon (1996), this way of thinking is completely practical and is associated with a more creative and collaborative action.

The first stage of the design thinking process was called empathy, where those involved in the solution started to listen to the customer's needs, understanding how, today, the existence of the module affects their routine and the business objectives they are part of. At this stage there was no room for judgment, nor for thinking that the user was already well known, since the idea was to present that situation as a novelty. The idea was just to listen to the usufructuary, see the points exposed by him and feel – in the sense of putting himself in the subject's place.

The second stage, called definition, consisted of synthesizing two points on small sheets of adhesive paper: who is the user and what was the problem he presented to us.

Having these two elements synthesized, the next phase was the idealization, where a group dynamic called brainstorming was carried out with the aim of making several people interested in participating in the construction of the project raise possible solutions. In this completely creative phase, all the conductors of the ceremony informed the participants that there would be no judgment, that no solution would present itself as right or wrong, since the objective was to use the imagination to provide as many solutions as possible. One point of improvement observed was the timing of this ceremony, which, as it was a very open phase – the satisfaction and fulfillment of the participants were notorious, since many adopted drawing, writing or even presentation techniques – many participants extended their presentation of the solution proposal.

Right after this brainstorming, the User Experience (UX) team started the stage that would most demand its analytical capacity, the prototyping stage, where the set of ideas that made the most business sense were grouped and transformed into prototypes.

Finally, with the prototypes created, the final step would be usability tests, where the prototypes were delivered to users who, in turn, were able to validate and vote for the best solution. Users were able to list the advantages and disadvantages of each one. At the end of the usability tests, with all the feedbacks provided by the users and with the analytical vision during the handling of the prototypes, the UX team already had a direction of which solution would be chosen.

Roadmap

The visual tool roadmap was adopted to manage the evolution of the product. Its construction was due to the prioritization of the main functionalities that were put in a sequential and simplistic visual model, however, clear enough to guide the team throughout the development process, as shown in Figure 1. The priority functionalities were distributed in the first construction phases and the others in sequence; thus, the product would already deliver value at the beginning of the project. During the course of the project, the roadmap was able to provide a status of where the team was, where they should be going and how to get there.

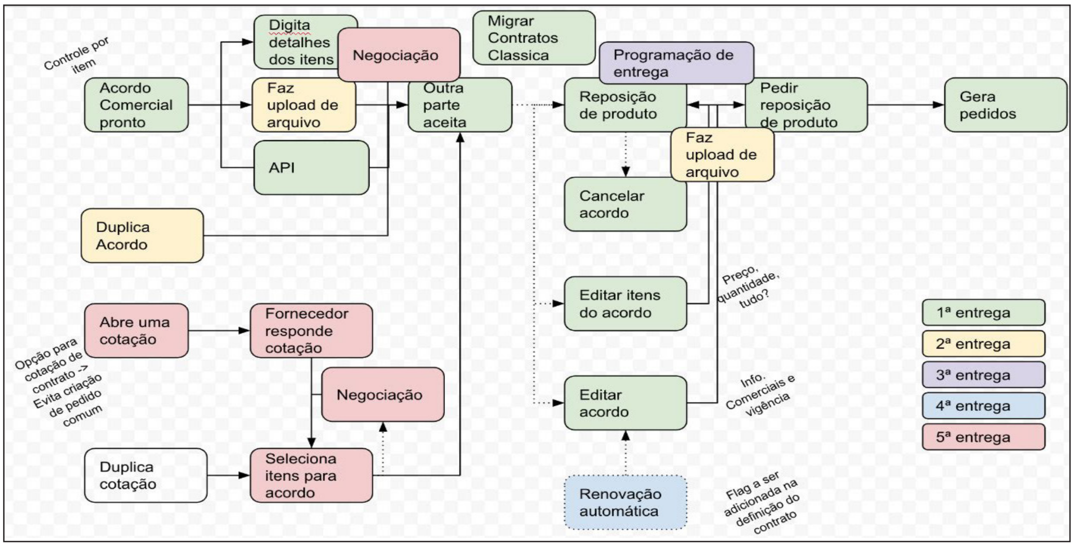


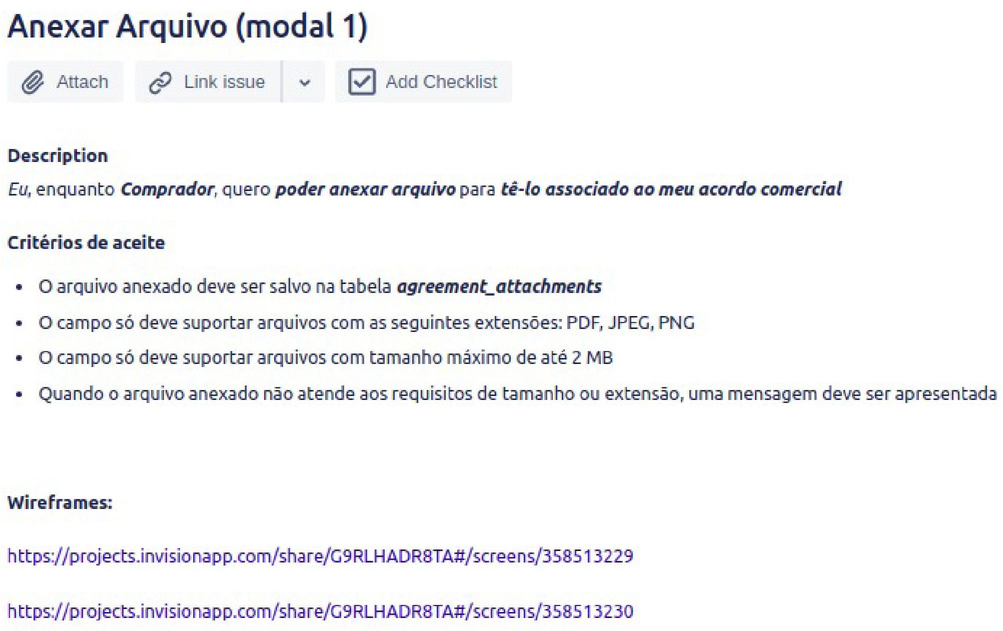
Figure 1. Product roadmap (Kickoff)
Source: Prepared by the author

With the chosen solution, it would now be up to the UX team together with the product team to present the proposal to the board and the solution development team. This presentation took place through a ceremony called Kickoff, that aimed to be the beginning meeting of the project, and in addition to introduce the members that make up the team, it defined objectives, material resources, restrictions, deadlines and schedules. Soon after this ceremony, the project could already be started.

Story Mapping

In the process of maturing the idea, the story mapping technique was used to create small stories that tell how users will use the functionality of the module to be developed. As you can see in Figure 2, these stories were written in a simple way and included the following sections: WHO, WHAT and WHY.

The “who” defines the user who will perform the action; “what” describes the user’s need; and the “why” transcribes the benefit to the user of implementing this functionality.



Anexar Arquivo (modal 1)

Description

Eu, enquanto ***Comprador***, quero ***poder anexar arquivo*** para ***tê-lo associado ao meu acordo comercial***

Critérios de aceite

- O arquivo anexado deve ser salvo na tabela ***agreement_attachments***
- O campo só deve suportar arquivos com as seguintes extensões: PDF, JPEG, PNG
- O campo só deve suportar arquivos com tamanho máximo de até 2 MB
- Quando o arquivo anexado não atende aos requisitos de tamanho ou extensão, uma mensagem deve ser apresentada

Wireframes:

https://projects.invisionapp.com/share/G9RLHADR8TA#/screens/358513229

https://projects.invisionapp.com/share/G9RLHADR8TA#/screens/358513230

Figure 2. Evidence of a user story
Source: Prepared by the author

When all the stories were created, we started to have a backlog of stories – a work list for the module to be completed – that were prioritized by the product manager.

Backlog refinement

After having all the stories prioritized and the development team having joined the product manager to validate the writing of these stories, this ceremony called backlog refinement took two hours and happened as follows: the product manager presented the stories; the team asked several questions, some were answered during the ceremony and others, for which the manager did not have the answer, were noted; several improvements in the writing of the stories were scored and technical descriptions were added to them by the more technical analysts. In the end, the backlog was more mature.

The role of the product manager in the software engineering process

The development team chose to adopt a hybrid software development model – uniting the concepts, good practices, ceremonies, roles and artifacts of the Scrum framework and the Kanban visual model to create a graphical representation of flows control, stories status and tasks under development.

The team’s sprint was defined to have one week, but later, due to the understanding of the complexity and gains for the team from the point of view of activities delivery, it was agreed that a sprint would have two weeks.

Every beginning of sprint, the development team, the agile coach and the product manager met at the planning ceremony to align what was being developed and delivered in that two-week time span. The presence of a product representative is of fundamental importance at this stage, as is he who will resolve any doubts that the team may have about the functionalities that will be implemented, in addition to renegotiating with the team possible changes that present as necessary at the time. During this one and a half hour ceremony, the entire team was shown to be participative, discussing solutions, translating functional aspects into technical requirements, renegotiating scope with the product manager, and directing activities according to the knowledge that each team member presented. The stories were dissolved in tasks, generating the possibility of more than one team member working on a single story, however, in different parts of it. Once the sprint was defined and it was already clear to everyone the objective of the next two weeks, the team started the development of the tasks.

During these weeks, all the ceremonies of the Scrum were respected. Sprint planning at the beginning of every sprint. Daily meeting being held every day to obtain reports on the individual activities of each employee. Sprint review at the end of it, presenting the features produced during the two weeks of work. Sprint retrospective fortnightly, helping the team to see the strengths, points to improve and creating action plans for the weaknesses raised.

In all these phases, in traditional models of software development, the requirements engineer or analyst (the professional responsible for viewing the product on models such as Classic or Cascade) does not act effectively in these meetings, being only the requirements artifact used to conduct these ceremonies, which can hinder the understanding of the team about what really needs to be accomplished. Soares (2004) reinforces that these methodologies make difficult to understand the requirements if a specialist professional is not directly involved in the entire development process, monitoring and updating the changes that may be necessary as the software is built. Agreements squad experienced a different reality, due to the product manager, a professional responsible for collecting requirements during product design, ensuring that requirements were questioned, changed, perfected and managed not only at the beginning of the development process, but throughout the module's development period. This professional was responsible for seeking more information that could help in understanding the features, solve doubts and negotiate changes with the engineering team.

Bug bash

After six months of work, several functionalities implemented and several changes made to the product code, the squad felt the need to guarantee a higher quality of the product that was being delivered. With so many functionalities delivered and being integrated gradually, coupled in an existing platform constantly updated (Regene) and communicating in an ecosystem that had a communication interface using outdated technologies (Classic) the feeling of the squad was of uncertainty about the quality of what was being delivered. To mitigate the possible impacts of product delivery, the bug bash ceremony was added to the process.

Bug bash is a software quality process that brings together professionals from different areas of the organization with the aim of using the product that is being developed. The Agreements 'bug bash' brought together professionals from different areas of the organization, who were strategically chosen, ensuring diverse and important views for the product. The objective of this process is to ensure that many people using the same application can point out different problems, improvements and solutions. The event was conducted in a gamification style, where people were divided into teams and the team that most encountered problems and improvements was rewarded.

During the entire meeting, it was possible to discuss functionalities, allowing the product manager to note difficulties inherent to usability, note suggestions for improvements and recognize problems in the product (i.e. bugs). These points were all recorded and subsequently prioritized by the team, since at the end of the bash bug the activity backlog contained eighty-three problems, fifty-five improvements and eight suggestions for new features.

With these artifacts, it was up to the product manager to analyze all the problems, improvements and suggestions mentioned, negotiating their prioritization with the development team. The team would spend another three months working exclusively on these items, thus delivering a more stable product.

Product management, communication interface and delivery time renegotiation

Initially, the period for the first delivery of the product was three months. After some uncertainty of requirements caused by the lack of understanding of how the customer actually used the module and the impact of not implementing some features that would not be delivered in the first phase of the product, the team needed to renegotiate the term, extending it to six months. During this time, the team delayed the completion of several stories due to technical difficulties with graphic interface creation libraries, in addition to uncertainties regarding integration with the Classic platform, and finally, the first delivery of the module was only implemented one year and three months after the project started.

Meanwhile, the active participation of the product manager was completely necessary, as it was this professional who performed the communication interface between the development team and the organization's board, understanding the technical difficulties, their impacts, and taking them in a filtered way and in a language with a relatively high abstraction level for those in charge. This communication needs to be effective, since it is from there the board chooses to authorize the extension of the term or to cancel the product.

What we delivered: a monthly delivery presentation meeting

At the end of each month, a meeting called "what we delivered" presented to the entire company what all the squads had delivered. Moderated by the product manager of each team, it was a great opportunity to contextualize people from different areas about what the teams were developing, in addition to being another moment to collect feedbacks and ask questions about the features presented.

Feedback from users and customers

With the completion of the functionalities delivered in the first phase of the module, before putting the product into production, some validation techniques should be used with customer representatives. For this validation, the usability testing technique is used again, this time no longer with the objective of validating a hypothesis (the prototype), but rather the product already implemented. The objective this time was to analyze the level of user satisfaction with the implemented solution, collecting feedbacks that would generate adjustments and improvements later implemented by the development team in a short period.

For these tests, the analysis environment was the company itself, within a special laboratory that the organization set up, simulating all the customers' space, from the physical space to resources such as computers and internet. The laboratory advantage was the possibility to analyze several customers on the same day, having a larger sample of customers in one week of tests application. During the process, the UX team was able to take notes and prepare a final report.

In general, customers pointed out some adjustments, some suggestions for improvements and rarely missed the way that some functionality was performed on the old platform. Most were satisfied with what was being presented, eager to use the module, and grateful to analyze the application before it was put into production.

Conclusion

Unlike models of traditional software development, where the phases of a project are sequential and always very well conducted in a previously planned period, with the adoption of product management and endorsed by the use of agile methodologies, the project made its processes less plastered, having its flexible dynamics. According to Lourenço (2011), traditional methodologies make the stages of the development process more sequential and linear, requiring a perfect understanding of the problem in the initial phase and the development of a vast amount of requirements and restrictions documentation; and only from there, allowing the development team to be involved in implementing the solution. That is, each role within the project has its responsibility within each stage of the project, providing artifacts for each other, but hardly building them together. Usually, this type of methodology only allows the solution to be validated after all its completion, that is, only after a long period of work can the team obtain feedbacks about what has been developed.

Product management reinforces the values and premises of the Agile Manifesto, where collaboration with customers is more valuable than extensive documentation and contracts, which makes changes in requirements somewhat considerable, as knowledge of features increases as software is developed. Thus, the ceremonies conducted by product management collaborate so that people from different areas act daily in conjunction with the development team, according to Torres (2015).

Kostoff and Schaller (2001) bring us the statement that the roadmap is an intelligent graphic representation that presents us paths and roads in a given region, that is, different paths a development team can follow to reach a certain place: product view. The roadmap was used throughout the project and is still used as a guiding tool, providing essential information and ensuring that planned objectives are being delivered. Using the authors' analogy, it is possible to affirm that, with effective roadmap management, the product manager was able to constantly revisit where the development team was heading, and, whenever distancing from the objective, redirect the team to the initially planned path.

The strategy of using story mapping to describe the features helped the team to stay focused on building small things, which integrated the big module in an integrated way, because it is easy to get lost in a global view of the module (i.e. big picture). Following this idea, Patton (2014) states that breaking stories into small parts (i.e. tasks) ensures that more people deliver small results that together add value to the developed business, in addition to guaranteeing the employees' feeling of participation. Also according to the author, this mapping keeps the team focused on users and their experiences, and the result is a better work dynamic and, consequently, a better product.

Stickdorn and Schneider (2014) emphasize that the possibility of using elements of psychology aligned with technology allows, in a motivational way, to support design thinking in this process of better understanding the user and his problem. The correct use of design tools, methods and processes brought a better view of the scenario to be later addressed by the development team, in addition to being a great way to insert the user into the product construction process from the beginning. .

One of the great advantages observed was the prior validation of solutions until then hypothetical before taking them to the development team. Rather than presenting written solutions, which often give scope for dubious and vague understanding, the adoption of prototypes made it possible to present the user with several solutions to his problem, where it was possible to analyze the difficulties and advantages presented in each proposed solution. Pressman (2006) reinforces that prototyping is an efficient mechanism for identifying software requirements due to their usability, since it is not just a written requirement where the user can have several projections about what will be implemented later.

Fernandez et al. (2011) argues that usability testing is one of the methods most adopted by current companies, where it is possible to analyze user data while they use the application. Through these tests it was possible to obtain data that helped in the decision of which approach would be implemented by the development team.

With the adoption of product management, communication was the most cited and reported advantage by the team. Having a collaborator responsible for making the interface between deliveries, needs and technical difficulties of software engineering – and the functional needs and expectations of the business area – made the relationship between the team and the company's board easier, with advantages for both sides: the team felt prestigious and the directors more confident in the work of the squad. Bahia (2013) reinforces the importance of investment in communication, showing that companies in Brazil and around the world have been hiring professionals with creativity, who are able to issue information in a clear and efficient way.

Product management proved to be the most efficient way to understand a customer's need, in addition to being the best way to strategically target products in order to serve and retain customers. According to CBS News (2012), product management is listed as the fourth most important function in companies, behind executive director, general manager and senior executive.

During the 1980s and 1990s, the development model of software was based on traditional methodologies, with a plastered and repetitive model. For two decades this model has sustained large organizations in a scenario of low competition and with a market that is difficult to penetrate. Currently, consolidated companies have several small competitors (startups) which, thanks to the democratization of information and the advancement of market development, ended up taking small portions of the market that previously belonged to large monopolies. With the arrival of this competition, the institution was forced to change its software development process, adopting disruptive processes that could respond quickly, creatively and innovatively to a still subtle competition. Product management is the ally of this change, since its adoption brought the team a product vision, disseminating the responsibility for what was developed to all members. Product management is responsible for all aspects of the product, ensuring the manager is concerned from user experience to engineering, including operational, sales and legal issues. This responsibility did not entrust the manager to implement all these specialties, function of the related areas and their professionals, aligned with the development team and the product's operational strategies. The development team being led by product management had its work dynamics, objectives and deliveries aligned with the organization's strategic objectives – deliveries represented not just code or another web system, but the organizational vision. The management of the backlog allowed requests for changes identified during development to be easily discussed and assumed by the team. The dynamics, ceremonies and meetings brought the client closer to the team, making small deliveries more accessible, also allowing route deviations – errors, misinterpreted requirements and improvements – to be discovered and solutions implemented in a timely manner. Product management has shown itself to be in a privileged position within the organization, with greater visibility about internal and external factors, equipped with information provided by several areas that are impacted by the product under development, carrying managers who are up to date with the market and, therefore, updated on the threats that competitors may present. And in light of that, demonstrating that product management is the most effective way to generate value to customers and accomplish the organization strategic goals.

Author's contribution: The author participated effectively in the conceptualization, elaboration, acquisition and analysis of data, methodology design and writing of the manuscript Product management and the construction of the required software: the case of the Commercial Agreements module.

How to cite: Azevêdo, L.S.L. 2021. Product management and the construction of the required software: the case of the Commercial Agreements module. Quaestum, 2: e26750561.

References

Anderson, D.J. 2010. Kanban - Mudança evolucionária de sucesso para seu negócio de tecnologia. Blue Hole Press, São Paulo, SP, Brasil.

Bahia, J. 2013. Introdução à comunicação empresarial. Mauad Editora Ltda, Rio de Janeiro, RJ, Brasil.

Bezerra, I. 2017. Guia rápido da história do computador. 1ed. On Line Editora, Barueri, SP, Brasil.

Daher, E. 2020. Okr. Clube de Autores, Joinville, SC, Brazil.

Eloi, D.S. De A. 2016. Um estudo comparativo de metodologias ágeis no desenvolvimento de aplicativos móveis. Undergraduate Thesis of Computer Science Course. Informatics Center, Universidade Federal de Pernambuco, Recife, PE, Brazil.

Fernandez, A.; Insfran, E.; Abrahão, S. 2011. Usability evaluation methods fot the web: A systematic mapping study. Information and Software Techcology, 53(8): 789-817.

Fridschtein, R. 2005. O Segredo do Sucesso na Internet. 1ed. Amazon.com Services LLC, São Paulo, SP, Brasil. Available in: <https://www.amazon.ca/sucesso-Internet-vis%C3%A3o-Mercado-Portuguese-ebook/dp/B07PRMQKS6>. Accessed in: jan. 10, 2020.

Iarandine, D. 2019. Agile Ceremonies: The details you were missing. Independently published, Lane Cove, NSW, Australia.

Kostoff, R.N.; Schaller, R.R. 2001. Science and Technology, Roadmaps. IEEE Transactions on Engineering Management, 48: 132-143.

Layton, M.C.; Morrow, D. 2019. Scrum para leigos. Starlin Alta Books, Rio de Janeiro, RJ, Brasil.

Leopoldino, C.B.; Borenstein, D. 2011. Componentes de risco para a gestão de projetos de software. REAd. Rev. eletrôn. adm., 17(3). doi: https://doi.org/10.1590/S1413-23112011000300003.

Liedtka, J.; Ogilvie, T. 2019., A magia do design thinking: um kit de ferramentas para o crescimento rápido da sua empresa. Editora Alta Books, Rio de Janeiro, RJ, Brasil.

Lombardo, C.T.; McCarthy, B.; Ryan, E.; Connors, M. 2017. Product Roadmaps Relaunched: How to Set Direction while Embracing Uncertainty. O'Reilly Media, Sebastopol, CA, USA.

Machado, F.N.R. 2016. Análise e gestão de requisitos de software: onde nascem os sistemas. Érica, São Paulo, SP, Brasil.

Malhotra, V. 2020. Single reference guide for scrum certification. Amazon Servicos de Varejo do Brasil Ltda, Seattle, WA, USA.

Mello, F.S.H. de. 2019. OKRs, From Mission to Metrics: How Objectives and Key Results Can Help Your Company Achieve Great Things. Qulture.Rocks, São Paulo, SP, Brazil.

Norlin, E.; Winters, C. 2002. Usability testing for library web sites: a hands-on guide. American Library Association, Chicago, IL, USA.

Oliveira, J.P.M.; Santos, C.S. 1998. A Informática nos Anos 90: Alguns Avanços e Desafios. In: I Conferência Científica da UFRGS, 1997, Porto Alegre, RS, Brazil. p. 2.

Patton, J. 2014. User story mapping. O'Reilly Media, Sebastopol, CA, USA.

Pressman, R.S. 2006. Engenharia de Software. 6ed. McGraw-Hill, São Paulo, SP, Brazil.

Rose, D. 2017. Agile at work: planning with agile user stories. Smart eBook Shop, Redwood, CA, EUA. Available in: <https://books.google.com.br/books?id=0qV6DwAAQBAJ&lpg=PT14&dq=user%20story&hl=pt-BR&pg=PT14#v=onepage&q=user%20story&f=false>. Acessed in: .

Sganderla, M.A.; Lacerda, G.; Ribeiro, V.G.; Silveira, S.R. 2016. Aprimorando a Gerência e o Desenvolvimento de Software com Metodologias Ágeis. RCT - Revista de Ciência e Tecnologia, 2(2). Available in: <https://revista.ufrr.br/rct/article/view/2681/2000>. Acessed in: jan. 11, 2020.

Simon, H.A. 1996. The Sciences of the Artificial. 3ed. MIT Press, Cambridge, MA, USA.

Soares, M.S. 2004. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. NFOCOMP Journal of Computer Science, 3(2): 8-13. Available in: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68>. Acessed in: mar. 26, 2020.

Stickdorn, M; Schneider, J. 2014. Isto é design thinking de serviços. Bookman, Porto Alegre, RS, Brazil.

Torres, J. 2015. Gestão de produtos de software: como aumentar as chances de sucesso do seu software. 1ed. Casa do Código, São Paulo, SP, Brasil.

Vazquez, C.E.; Simões, G.S. 2016. Engenharia de requisitos: software orientado ao negócio. Brasport, Rio de Janeiro, RJ, Brasil.