



A Product Owner's Navigation in Power Imbalance Between Business and IT: An Experience Report

Lotte Mygind¹ , Jens Bæk Jørgensen¹ , and Lutz Prechelt² 

¹ Mjølner Informatics A/S, Aarhus, Denmark

{lmy, jbj}@mjolner.dk

² Freie Universität, Berlin, Germany

prechelt@inf.fu-berlin.de

Abstract. [Context and motivation] We consider a company where software development was previously a minor activity and today is a major activity with high priority and attention. Software is now developed according to Scrum, and the company can be seen as being in an *agile transition*. [Question/problem] What are the relevant specifics of this organization and which product owner behaviors appear to be valuable or problematic, respectively? [Principal ideas/results] A fear of disruption put the development teams under pressures that led to low efficiency in an interesting way. The introduction of an *IT product owner* to assist a *business product owner* reduced this effect, but a problematic power imbalance still remains. [Contribution] Not only do agile technical teams need feedback from their product owner, the product owner also needs meaningful, effective feedback from the teams. Our experience report shows how this can be improved by the introduction of an IT product owner when the organization otherwise has insufficient focus on certain important dynamics of software engineering.

Keywords: Requirements Engineering · Product Ownership · Project Organization · Cooperation between Business and IT · Software Quality

1 Introduction

We consider a company where one of the authors of this paper has been a consultant as product owner. The product owner, an integral part of Scrum [1], is a key role in modern agile software projects in the industry, and the role is crucial for the quality of the developed solutions. A product owner should have knowledge of requirements engineering and should facilitate the interplay between business stakeholders and software development teams. The product owner must contribute significantly to the elicitation, specification, validation and prioritization of requirements in a given organizational context.

In [2], Lauesen's taxonomy for functional requirements includes goal-level, domain-level, product-level, and design-level. In most software development projects, functional requirements on all four levels are relevant and exist with different degrees of explicit

or implicit representation. A product owner must be able to work at all four levels and must facilitate bridging between the levels with assistance from experts, such as an IT architect or a user experience expert.

Product ownership in software projects is described in numerous papers, for example, [3–6]. These papers are case studies by researchers, while this paper is an experience report based on a practitioner’s participation in a project for two years.

This paper introduces the company, project and product owner role in Sects. 2, 3 and 4. Section 5 presents a quantitative analysis of the backlog. Section 6 describes key observations about the product owner’s work and the lesson learned: there is a power imbalance between business and IT which is too big. Related work is discussed in Sect. 7, which is followed by the conclusions in Sect. 8.

2 The Company

The company¹ is a traditional company in the fossil fuel industry. The main products are threatened by disruption in the form of electrification. In response to this, the company has introduced a new business area with a number of electrical products, which are highly interactive and digital. Looking back 5–10 years, the company had fewer than 20 employees working with IT, and today this has grown to around 150 people.

The company has a business division and an IT department, including a large IT development department headed by an IT development manager. A user experience department is part of the business division, see Fig. 1.

The product owner role is shared between a business product owner and an IT product owner, employed in the new business area and IT respectively. The IT product owner is one of the authors of this paper.

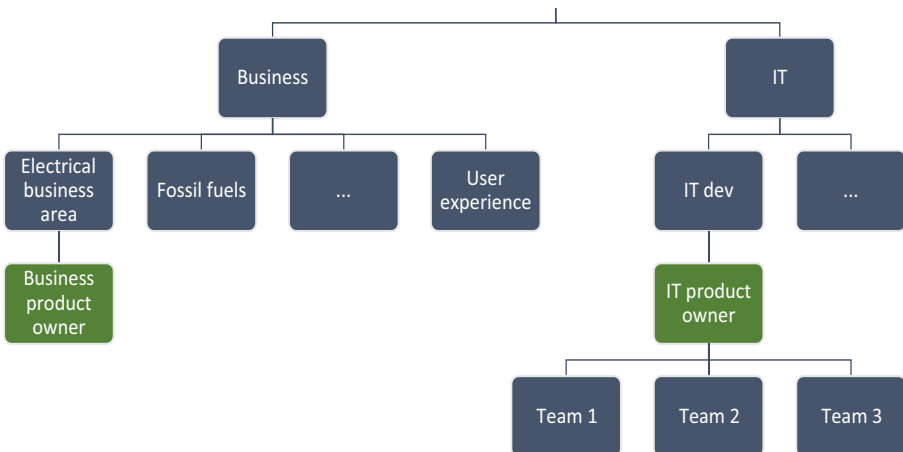


Fig. 1. Company organization.

¹ The company is kept anonymous.

3 The Project

3.1 History

After the new business area was introduced in late 2019, a software development project to support it was initiated in early 2020. An IT architect, several software developers and the business product owner joined the project within a few months.

The staffing has grown significantly since then. As of now, there are three Scrum teams, each with a Scrum master. In addition, there is an IT architect, a technical supporter, two testers, and the IT and business product owners², and these roles support all teams. The total number of team members in the IT department is 24 plus eight at a subcontractor. Moreover, the company plans to hire more developers for an additional team in the near future. Lastly, several user experience specialists work with the product. Figure 2 illustrates the timeline.

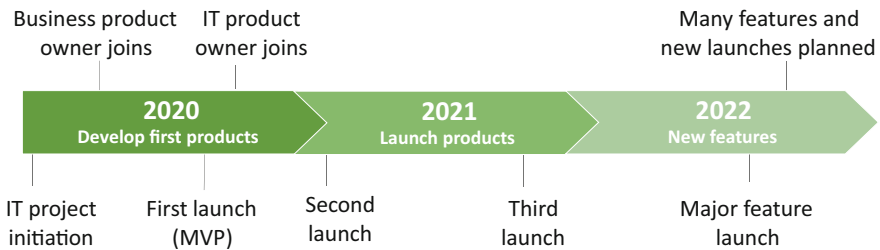


Fig. 2. Timeline of the project.

The company expects significant IT development for a decade ahead across several product suites in the new business area. For this reason, the activities can be seen as numerous, concurrent projects, or a *program* in the sense of project management literature. We use the term *project*, because this is how the company describes it.

3.2 Project Organization

The IT product owner works with all three Scrum teams. Furthermore, there is an IT project manager whose main responsibility is to create realistic plans for major releases. He is also a member of a steering group along with the IT development manager.

The chart in Fig. 3 illustrates the main flow of development tasks. The business product owner describes new domain-level requirements and communicates these to the IT product owner for analysis and feedback. When needed, the IT product owner consults the architect for an early, deeper analysis of the most complex requirements. For most requirements, the architect will not see the user stories until sprint start, and this works well most of the time. If the IT product owner has overlooked any non-trivial technical issues, the user story is taken out of the sprint to be analyzed further.

² Two additional IT product owners were hired shortly before the author left the project. This article describes the time with only one IT product owner.

When applicable, the user experience team cooperate with the IT product owner and describe a detailed solution for the user interface, i.e., design-level requirements. There are numerous alternate flows and backward loops, in addition to the main flow depicted in Fig. 3.

The IT product owner writes user stories (in a broad interpretation of this term) for specification to the development teams and defines the verification criteria. When the developers have implemented a user story, the testers verify it according to the verification criteria. Finally, the functionality is deployed to the production environment.

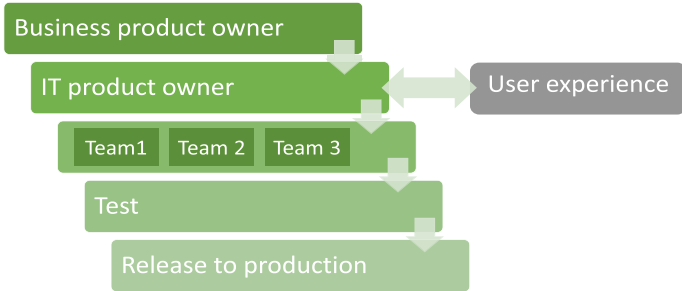


Fig. 3. Flow of requirements-related and development tasks in the project organization.

4 The Product Owner Role

Before this project, the company used product owners from the business division, with the mindset that the business would order IT deliveries from the IT department.

When the project started in early 2020, the business area was new, and a product owner was hired as the second employee in the business area. He has a background in business administration with limited experience of IT development projects and no domain knowledge. As the project grew, it became clear that he needed assistance in requirements engineering, including elicitation, specification, validation and prioritization of requirements.

The company hired one of the authors as a consultant who could teach the user story technique to the business product owner. She is an experienced product owner and has extensive domain knowledge in the relevant industry. She joined the project in October 2020. During the following months it became clear that the business product owner only had the time for and the interest in goal- and domain-level requirements, so the author was given a long-term contract, and people gradually started to describe her as the “the IT product owner”, a new role in the organization (Fig. 4).

In June 2021 the management of the IT development asked the IT product owner and the IT project manager to describe the IT product owner role and a list of qualifications. After this, IT management decided that all IT projects should have such an IT product owner, so most projects in the organization now have an IT product owner.

The Scrum Guide [1] does not state whether the product owner is a part of the business organization or a part of the IT organization. The role of the product owner

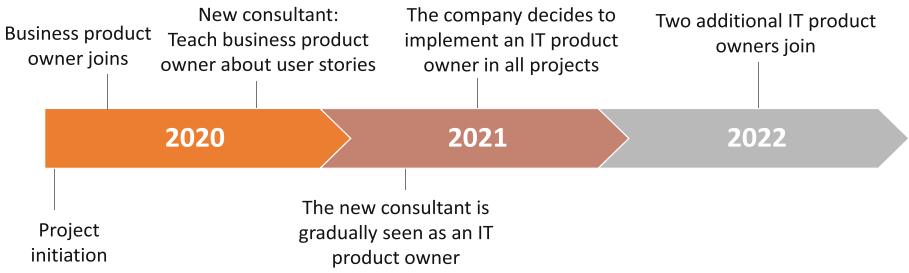


Fig. 4. The product owner role has evolved during the lifetime of the project.

is to “maximize the value of the product resulting from the work of the Scrum Team.” The product owner must “define and communicate product goals, maintain the product backlog and ensure the backlog is transparent, visible and understood.” In the project considered here, the product owner role is implemented by the joint work of the business product owner and the IT product owner.

4.1 Division of Work with Requirements

In most cases, the requirements elicitation, specification/description and prioritization are shared between the steering group, product owners and user experience as shown in Fig. 5.

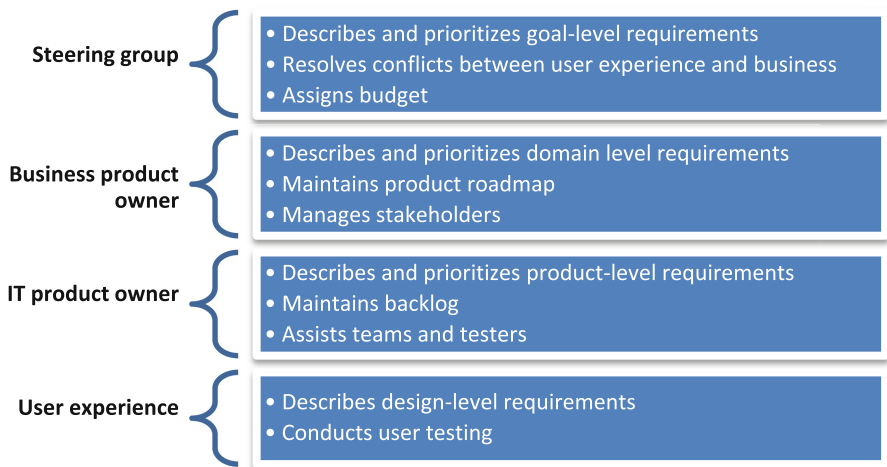


Fig. 5. The work with requirements is shared between various roles.

The goal-level requirements of the steering group are based on input from the product roadmap. The steering group study the product roadmap in order to find the most important areas and define goal-level requirements, such as “At the end of 2023 the

number of customers for product X should be more than Y” or “Feature A must be ready for all private customers in July 2023”.

The business product owner gathers input for the product roadmap based on feedback from sales and end users (primarily via customer support). The roadmap is normally presented to the steering group without any estimates of the complexity of the features.

The two product owners meet at least twice a week to coordinate work. The business product owner typically brings a list of domain-level requirements, while the IT product owner brings a list of questions, feedback and analysis results, which often include estimates and risk assessments.

When the IT product owner believes there is a good breakdown of a requirement with sufficient implementation details, the architect makes a rough estimate. Whenever the estimate is higher than expected by the business or high compared to the relative value of the feature, the IT product owner discusses the situation with the business product owner. This often leads to re-prioritization.

4.2 User Stories, Verification and Definition of Done

When the IT product owner joined the project, she intended that all work items should be user stories describing the direct value for a user role. We explain below that the IT product owner applies a modified approach.

The teams use a continuous integration strategy, where automated tests are implemented by the developers as part of the coding, and these tests are supplemented with work item verification by a tester. If a work item has been active for a long time before verification, it is difficult to release it without also releasing other (unverified) changes.

As the project grew, the verification process often blocked the release pipeline, so the teams now prefer small, independent work items that can be released independently. Therefore, the IT product owner creates small, independent work items that add either value or functionality, and these are not necessarily traditional *user stories* that add direct value for users. Implementation of a new internal endpoint is an example of a work item that does not add value for users, but it can be tested and released independently. If a work item is not testable, the IT product owner makes sure that it is tested at a later point.

There are two checkpoints that each work item must pass before release. The first is a checklist for the software developers before the work item is assigned to a tester. This includes checks of code quality, peer review and unit tests. The second checkpoint is verification according to the criteria described in the work item. Thus, the definition of done in the sense of Scrum [1] combines the developer checklist and the verification.

5 Quantitative Analysis of the Product Backlog

The status and history of the project’s work items are analyzed in this section in order to see if the data may serve to illustrate the state of the project and the IT product owner’s work. We discuss the data in further details in Sect. 6.

The project has a history of a growing backlog³, see Fig. 6. The number of bugs is low due to an explicit prioritization: At each sprint planning, resolution of bugs is prioritized highest.

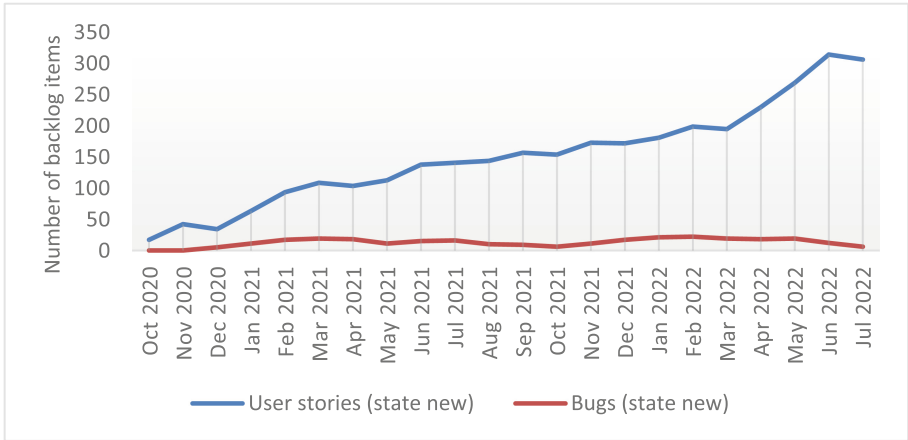


Fig. 6. The size of the backlog (new work items) is growing, but the number of bugs is stable.

The work items are grouped in features, and there are 93 features in state new or active⁴ (65% see Fig. 7). In Sect. 6.1, we explain how a high business pressure may simultaneously result in many features in state new. Furthermore, Sect. 6.2 shows how the pressure at the same time may create a situation with many active, incomplete features.

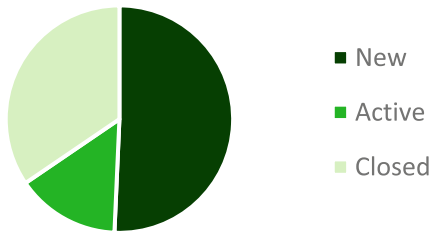


Fig. 7. 35% of features are closed. The majority of features (51%) are in state *new*.

To understand the features that have not yet been fully implemented (closed), we have investigated the lifecycle of the individual backlog items, see Fig. 8. In total 22% of the backlog items created during this time period are still in state new. This will be discussed further in Sect. 6.

³ All data in this section is for work items created before August 1st, 2022. The data has been extracted from the historical user stories in Azure Devops.

⁴ The state of the work items on August 1st, 2022.

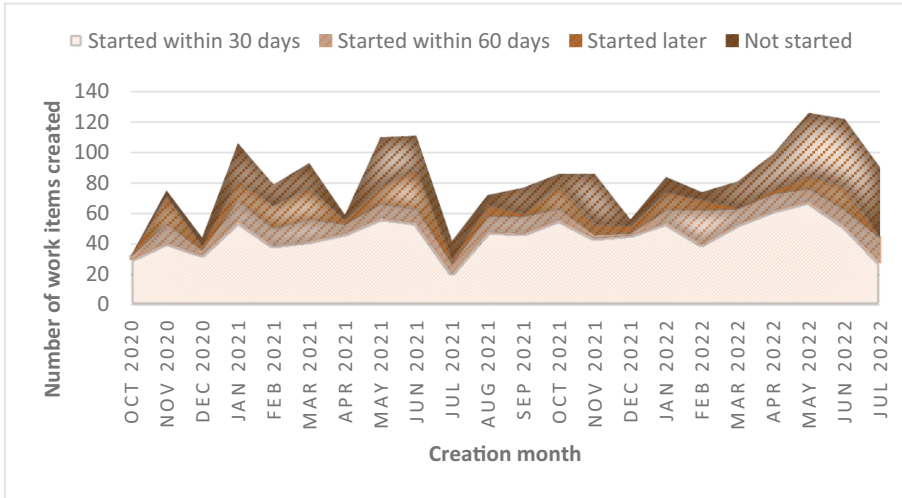


Fig. 8. Number of work items per creation month. The darkest color shows user stories that have been left in state *new*⁵.

6 IT Product Owner Observations and Lesson Learned

Based both on the description of the product owner role and the quantitative analysis of the preceding sections, we will now describe two observations and a main lesson learned, which have their roots in the power imbalance between business and IT. This imbalance occurs in many companies, since their main focus is to manufacture and sell their products. Software development is just one activity among many others, and sometimes not a high-attention activity seen from the perspective of a company’s business side and senior management. The imbalance is predictable, and it is our responsibility as software professionals to handle it properly to ensure good working conditions for software teams for the benefit of the company as a whole.

6.1 Mismatch Between Organizational Pressure and Product Owner Capacity

In early 2021, the business pressure increased. In fact, the business product owner extended the IT roadmap of the project to include 81 domain-level requirements.

From mid-2021 it became difficult for the IT product owner to find time for all the work. The number of customers grew rapidly, and since the IT product owner was the point of contact to the business, she constantly had to delegate production errors and support issues to team members. At the same time, the company gradually added more developers, so the task of supporting the work of all developers and testers grew. In the autumn of 2021, the project went from two to three Scrum teams.

⁵ The state of the work items was extracted from Azure Devops on September 30th, 2022. Only work items created before August 1st, 2022 have been included.

The early estimates made by the IT product owner have only had limited impact on the priorities of the 81 domain-level requirements. There was, for example, one domain-level requirement that the business expected to be completed in a few sprints, whereas the IT product owner estimated it to be a complex subproject with a time horizon of many months. The business product owner did not accept this as a fact, so the IT product owner spent several months breaking the domain-level requirement into work items for more detailed estimation. In the end, the project was postponed because of the high estimates, and this was escalated to the steering group. The requirement was, however, immediately replaced by another subproject that had not been estimated at all. In this way, nobody evaluated whether the business value would match the effort for the new requirements, and implementation started before the full analysis had been done.

In this way, postponed subprojects are part of the reason behind the many features in state new on Fig. 7. Furthermore, the corresponding user stories are represented as a spike of unimplemented work items in Fig. 8. At the end of 2021 we estimate that the product owner spent 75% of her time on day-to-day support of the development teams. At this point, there were still approximately 80 domain-level requirements. Some of these were themselves full-sized projects. If we estimate an average of two workdays of product owner work for each domain-level requirement, this adds up to eight months of full-time work, which can be completed over 32 months (at 25% capacity for this work). This amount of time is far beyond reasonable and is an impediment for agile development.

On Fig. 8 the darkest color illustrates requirements that are dropped. This reflects the high number of ideas on the roadmap that the IT product owner has spent time on, but the team has not. Thus, there is a mismatch between the business' requests for analysis and the capacities of both the IT product owner and the development teams.

At the same time Fig. 8 shows that many work items are implemented after more than 60 days, and for these, the IT product owner must often update the descriptions before the backlog item is handed over to the development teams. After two months the software has often had changes to relevant data structures or interfaces invalidating the old description. Thus, the IT product owner must find time to analyze and describe more items than the teams implement, and the items that do get implemented must often go through an extra revision, because implementation is delayed.

The data illustrates a high-pressure environment where the IT product owner must analyze a high number of requirements. The effect was that she could not always provide qualified input to the prioritizations of the business product owner and steering group in time.

There are two obvious solutions. The first is to add more resources to the IT product owner function, which the company has done in mid-2022. The new resources are strong profiles who will be able to work as two cooperating IT product owners. It is unclear whether the two new IT product owners have enough capacity to make sure that steering group prioritizations are consistently based on estimates.

The second solution is to add a mechanism for prioritization and long-term planning of the IT product owner's work. This could, for example, be in the form of a board where the work of the IT product owner is tracked (for example, in status new, active and done), and this board could be used for communication with the business product

owner in order to prioritize the work and limit the number of parallel tracks. This has not yet been implemented. Such planning of the product owner's work is difficult to accept for the business, because the high pressure creates an environment where the business always wants more analysis in the hope that the product owner can find an easy solution for the next problem.

6.2 Insufficient Trade-Offs Between Business Demands and Software Quality

In this project, we have often seen that when the most central functionality supporting a domain-level requirement has been implemented, the business wants to prioritize something else. Under the pressure, the business wants new features quickly, and as a result, the product has a lot of barely-viable solutions, which are reflected as active features in Fig. 7.

While the number of customers has grown significantly, the number of people working in administrative functions to perform workarounds for missing functionality has also grown significantly. At the same time the performance of the IT development teams is negatively affected by an increasing number of database updates and support for the administrative workers.

In Scrum, one of the product owner's most important tasks is to make prioritization continuously, sprint after sprint. As we see here, domain-level requirements with high priority might get lower priority at some point, and many features are never completed. In the opinion of the IT product owner, architect and, in fact, all development team members, the business product owner's prioritization is too volatile, resulting in incomplete features that create extra work for the development teams in the long run.

While a lot of bad decisions have been avoided because of good communication between the business and IT product owners, it has been difficult to find the right balance; this has sometimes been subject to intense debates between the business product owner and the IT product owner. When everything is broken into small, independent work items for prioritization (see Sect. 4.2), the business product owner can pinpoint elements that seem minor from a functional point of view.

In the aftermath of all these minimal implementations, the IT product owner has learned that due to the high business pressure, it is difficult for the business product owner to prioritize features that are not directly linked to the product roadmap. After a lot of discussions between the two product owners, the business product owner has extended the roadmap with some features that ensure continued scalability, operability and performance – more generally, software quality. In this way, the IT product owner has pushed technical requirements to be recognized and prioritized.

6.3 Lesson Learned: Power Imbalance Between Business and IT is Too Big and Must Be Addressed

The observations above illustrate the power imbalance between the business and IT. The company is in a quick transition from a traditional, industrial company to an IT service company, but the company has not yet reached maturity as to how to prioritize larger business roadmaps.

The company has taken one important step in the transition, namely hiring IT product owners to work with all development teams, but the IT product owners have not been given a proper mandate. In the current set-up, the IT product owners can argue their cases and deliver estimates for early prioritization, but they are often overruled.

The power imbalance between business and IT must be addressed to ensure continued and efficient progress in the company's agile transition. Looking back, the IT product owner could have attempted to do more to communicate this observation to upper management at an earlier point in time in an attempt to contribute to a resolution. This is difficult, however, as the root causes lie in the company's organization and its inherent culture, so it is likely that the necessary changes will not happen in a short time frame.

One approach for the communication could be to show some of the data in Sect. 5 to higher-level management. The figures illustrate the ever-changing directions of the project with many features that are not *closed* (Fig. 7) and many work items where implementation starts late or never starts (Fig. 8). Furthermore, the IT product owner could have kept a record of goal-level requirements where the teams were told to start implementation immediately without further analysis (see an example in Sect. 6.1) and situations where the recommendations of the IT product owner were overruled by the business (see examples in Sect. 6.2).

7 Related Work

The Scrum guide [1] insists that “The Product Owner is one person, not a committee.” In the project considered in this paper, it was vital that the product owner became a committee consisting of business product owner and IT product owner. The reason was threefold: scale - the business product owner being too busy to fully do all the things the teams needed; maturity - the business product owner not understanding the implications of his actions well enough; and power - there was no one who could successfully oppose the business product owner when needed.

Our experience, that the product owner gets too busy when the organization around her grows, is of course a common problem, and there are several approaches to scale agile development, including the scaled agile framework (SAFe) and Scrum@scale. For a recent overview of approaches, see Edison et al.'s survey paper [7].

Scalability is obviously an important issue, but there are also other issues which are relevant to discuss. We use Bass and Haxby's framework of roles in product owner teams as a theoretical lens for obtaining a useful alternative view of our case [8]. They consider three dimensions they call Scale, Distance, and Governance. Distance effects are absent in our case; scale effects have already been discussed; regarding governance, Bass and Haxby suggest three roles for solving Governance issues: Technical Architect, Governor (who watches over adherence to corporate technical policies), and Risk Assessor. The latter was required for solving the problem in our case: Evaluate and highlight risks involved in a particular course of action, so that problematic routes can be avoided. Bass and Haxby also recommend making product owner teams explicit, which happened naturally in our case.

For understanding the maturity dimension, the mapping study by Unger-Windeler et al. [3] is helpful. It maps out topics and findings from 30 studies of product owners in

industrial settings. Like [8], they find that product owners for larger-scale settings will have to be teams. In terms of roles, they list no fewer than 14 distinct roles for product owners that have been discussed in multiple (as opposed to only one) publications. Two of them help understanding what went wrong in our organization initially, before the IT product owner was introduced: Prioritizing the backlog [4, 9, 10] and managing expectations [11, 12]. Both are rather basic aspects of product owner work. Yet our business product owner was unable to manage expectations. The reason was that he did not know what to expect because he was relatively inexperienced with large-scale software development. For the same reason, his prioritization used inappropriate criteria, and resulted in the negative side effects we have described.

As for the power imbalance problem, Scrum has two roles responsible for solving that: The product owner, by prioritizing work appropriately, and the Scrum Master, by fighting against distractions (such as the support work arising from the incomplete features). Boehm and Turner called these tasks the “protector” role already in 2005 [13]. Yet in our case, coach roles had barely developed in the organization and the product owner was the very source of the problem. Therefore, it required the introduction of the IT product owner to get to effective software development.

8 Conclusions

In this paper, we have described the work of the IT product owner and her navigation in an organization where there is an imbalance between business and IT. We have supplemented this with a quantitative analysis to illustrate and discuss the state of the project and the product owner’s work. While the analysis has the obvious weakness that it may show the product owner’s work methods more clearly than the general state of the project, it would be interesting to compare such data for a larger number of projects.

In general, a product owner builds a bridge between business and IT, but this requires an organization that is in better balance than the one described in this paper. As IT professionals, we should work on strengthening our communication with stakeholders with no or low interest in software development in the organizations we work in. An example of this is described in [14].

We have been partially successful in doing that: Upper management has recognized the benefits of strong product ownership, and for this reason, as mentioned in Sect. 4, it has been decided to apply a set-up with a business product owner and an IT product owner to all other IT projects in the organization.

A reason for this is that the product ownership in the considered project has been strong in many respects, and it has been visible for upper management that this has been the case. There is a well-structured, well-described backlog, strong communication between business and IT and good access to support for developers and testers. The organization saves development time, because it is ensured that focus is on features that bring the right amount of value compared to the effort. The developers are, in general, very satisfied with this approach. They get well-defined user stories in sprints, so they can focus on their favorite activity, namely coding. They have immediate access to clarification from the IT product owner when needed and they are always able to discuss alternative implementation ideas with the architect or IT product owner. Furthermore, they experience only few interruptions from analysis of future features.

On the other hand, the company's agile transition is certainly not yet completed. It is a problem that many business stakeholders in the company still see IT as a supporting function where they "order" IT systems. The terminology in the business is: "We tell you what we want and when, and you tell us the cost".

We believe that the company could get more value by involving IT more and earlier in identifying where to get the best value compared to the effort. The company has to some degree done this by introducing the IT product owner role, but the IT product owner has only had the power of argumentation - and has often been overruled.

References

1. Schwaber, K., Sutherland, J.: The Scrum Guide – The Definitive Guide to Scrum: The Rules of the Game. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
2. Lauesen, S.: Software Requirements – Styles and Techniques. Addison Wesley (2004)
3. Unger-Windeler, C., Klünder, J., Schneider, K.: A mapping study on product owners in industry: identifying future research directions. In: 2019 IEEE/ACM International Conference on Software and System Processes (ICSSP), Montreal, Quebec, Canada. <https://doi.org/10.1109/icssp.2019.00026>
4. Bass, J.M., Beecham, S., Razzak, M.A., Canna, C.M., Noll, J.: An empirical study of the product owner role in scrum. In: ICSE 2018: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings, Gothenburg, Sweden. IEEE (2018). <https://doi.org/10.1145/3183440.3195066>
5. Maturro, G., Cordoves, F., Solari, M.: The role of product owner from the practitioner's perspective. An exploratory study. In: 16th International Conference on Software Engineering Research and Practice (SERP 2018), Las Vegas, Nevada, USA, pp. 113–118. CSREA Press (2018)
6. Kristinsdottir, S., Larusdottir, M., Cajander, Å.: Responsibilities and challenges of product owners at spotify - an exploratory case study. In: Bogdan, C., et al. (eds.) HCSE/HESDD -2016. LNCS, vol. 9856, pp. 3–16. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44902-9_1
7. Edison, H., Wang, X., Conboy, K.: Comparing methods for large-scale agile software development: a systematic literature review. *IEEE Trans. Softw. Eng.* **48**(8), 2709–31 (2022). IEEE. <https://doi.org/10.1109/tse.2021.3069039>
8. Bass, J.M., Haxby, A.: Tailoring product ownership in large-scale agile projects: managing scale, distance, and governance. *IEEE Softw.* **36**(2), 58–63 (2019). <https://doi.org/10.1109/MS.2018.2885524>
9. Bass, J.M.: How product owner teams scale agile methods to large distributed enterprises. *Empir. Softw. Eng.* **20**(6), 1525–1557 (2014). <https://doi.org/10.1007/s10664-014-9322-z>
10. Bass, J. M., Beecham, S., Razzak, M. A., Canna, C. N., Noll, J.: An empirical study of the product owner role in scrum. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings (2018). <https://doi.org/10.1145/3183440.3195066>
11. Finsterwalder, M.: Does XP need a professional Customer? In: Proceedings of the XP2001 Workshop on Customer Involvement (2001)
12. Sverrisdottir, H.S., Ingason, H.T., Jonasson, H.I.: The role of the product owner in scrum: comparison between theory and practices. *Procedia Soc. Behav. Sci.* **119**, 257–267 (2014). <https://doi.org/10.1016/j.sbspro.2014.03.030>

13. Boehm, B., Turner, R.: Management challenges to implementing agile processes in traditional industrial organizations. In: *IEEE Softw.* **22**(5), 30–39 (2005). IEEE. <https://doi.org/10.1109/ms.2005.129>
14. Jørgensen, J.B., Christensen, H.L., Hansen, S.T., Nyeng, B.B.: Effective communication about software in a traditional industrial company. In: *2022 IEEE 44th International Conference on Software Engineering (ICSE), 5th International Workshop on Software-Intensive Business*, Pittsburg, Pennsylvania, USA. IEEE (2022). <https://doi.org/10.1145/3524614.3528625>