Kantola, Konsta; Vanhanen, Jari; Tolvanen, Jussi

Mind the product owner: An action research project into agile release planning

Published in:
Information and Software Technology

DOI:
10.1016/j.infsof.2022.106900

Published: 01/07/2022

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.
Mind the product owner: An action research project into agile release planning
Konsta Kantola a,b,*, Jari Vanhanen a, Jussi Tolvanen b
a Aalto University, P.O. Box 11000, AALTO, FI-00076, Finland
b Accountor Finago Oy, Keilaniementie 1, Espoo, 02150, Finland

A R T I C L E   I N F O
Keywords:
Canonical action research
Agile release planning
Product owner

A B S T R A C T
Context: This paper studies agile release planning in a software development organization with 13 development teams. It is important for software development organizations to be able to plan work in an efficient way that supports development work.
Objective: The research aims to understand issues within agile release planning in the studied organization, and to make some improvement to the agile release planning practices there.
Method: The study followed canonical action research methodology completing one cycle of diagnosis, action planning, intervention, evaluation, and learning. Qualitative methods were used during these phases to identify preliminary issues, to support the choice of action, and the evaluation of those actions.
Results: The research identified issues of strain on the role of Product Owners. Sources of strain in the organization include changing priorities, the effort required to build up domain competence for new projects, and external pressure to push out new features. Additionally, there was difficulty for people participating in agile release planning to suggest improvements to the used practices due in part to the complexity and scale of planning practices in a multi-team development organization. The actions taken as part of the research provided ways for Product Owners to share knowledge between themselves, to better affect the working practices in the organization, and promoted a sense of team spirit between the Product Owners.
Conclusion: Organizations should be mindful of their Product Owners when looking at their release planning practices. Problems for Product Owners are problems in planning for the whole organization. Having an active, collective, and structured channel for continuous improvement for Product Owners can help drive improvements to agile release planning.

1. Introduction
It is important for software development organizations to be able to plan work in a way that is efficient and supports development work. In agile organizations these activities can be called agile release planning. Research into release planning in agile development organizations is limited and often lacks guidance on how to address identified issues.

From an interest in improving planning processes at Accountor Finago, and a desire to try out our hand at research came this study. We decided on an action research project with the goal of improving agile release planning practices in the organization and to understand issues in agile release planning in the organization.

The research project went through one cycle of diagnosis, action planning, intervention, evaluation, and learning in the structure of canonical action research. During those phases we used qualitative methods of semi-structured interviews and participant observation.

The actions taken as a part of this cycle focused heavily on Product Owners as they have a central role in planning development work. The actions introduced three new practices for the Product Owners, two of which were continued at the end of the research project. These were two meetings called the Scrum of POs and the Retro of POs. They in part helped addressed identified issues of strain on the role of POs and a lack of self-efficacy for POs concerning release planning practices.

With this study we extend the body of knowledge on issues in agile release planning in real-world contexts; a topic which is relevant to practice and not yet researched enough. We also add new perspective to the topic from our choice of methodology, since to our knowledge this is the first study of issues in agile release planning by way of action research.

The rest of this paper has six sections. Section 2 presents some of the relevant literature. Section 3 introduces the organization where the
study took place. Section 4 explains the research framework used and the methods used for each distinct phase of that framework. Section 5 has the results of each of those phases. In Section 6 we discuss the answers to the research questions and reflect on the limitations of the research and suitability of the methods used. Finally in Section 7 we conclude with the contributions of this paper for both research and practice.

2. Background and related work

In this section we will first give some background into what agile release planning is. Then we will go over related literature on issues in agile release planning. Finally, we will cover research on the role of POs, as it is relevant to discuss the reactions to the actions taken as part of this study.

2.1. Agile release planning

We take the stance that agile release planning is all the activities that go into answering the question of what work an agile software development organization intends to do in the near future. The resulting answer is the release plan. The release plan has no inherent value, only instrumental value in helping the organization reach its goals. Hence, it is also important to consider how the release plan is used. This paper is mainly focused on the use of the release plan to support development work.

Ameller et al. [1] present a comprehensive survey of existing mathematical models for software release planning. Although empirical studies have found such models lacking in practice [2], they give a good overview of what different activities to consider. The inputs for the models include a list of candidate requirements, effort estimates for the requirements, dependencies between requirements, time constraints like deadlines for particular requirements, and resource constraints like a requirement needing some specific skills to implement [1]. The models also need to consider the different business value of requirements and the interests of different stakeholders [1]. The outputs for the release planning models include a prioritized list of requirements, a schedule for their completion and assignments of tasks to developers [1].

Requirements are often expressed with user stories in agile requirements engineering [3]. But more than written documentation for requirements, agile release planning relies on face-to-face communication, which has reduced many communication problems that would lead to misunderstood requirements [3]. Thanks in part to the limited documentation and the emphasis on direct communication, agile release planning is seen as a comparatively efficient process [2,4].

The Scrum Guide instructs keeping requirements in prioritized backlogs [5]. This is one of the more commonly reported traits of agile release planning in practice [3]. However, a simple backlog of user stories is not enough for many development organizations, instead larger development organizations may opt for a hierarchical requirements model [4]. It is especially important in large projects to have clear priorities [6].

Release plans should not be set in stone once made. The agile manifesto values responding to change over following a plan [7]. Regular feedback on completed requirements allows for adjusting plans as necessary [5]. Planning and adjusting continuously is common in agile organizations [3,4]. The increased flexibility of agile release planning methods is perceived as a benefit [4].

In summary, agile release planning is a complex process consisting of many different inter-dependent activities. There is a wide range of interactions to consider and a broad human element to understand in order to study the agile release planning of any organization. Finally, it seems noteworthy to mention that even though we use the term release planning for this process, the process is not strictly tied to the actual release of the software.

2.2. Issues in agile release planning

Related literature into issues in agile release planning is not too common. Inayat et al. [3] presents a review of papers on agile requirements engineering published between 2002 and June 2013, most of which were case studies. After that we found a dissertation on agile release planning by Heikkilä [2] containing three case studies [8–10], one of which was later expanded on by Heikkilä et al. [4]. All of them answer some version of the question what issues are there in agile release planning.

Much of the above research comes from a context of organizations that have recently undergone an agile transformation. Many of the challenges identified are specific to that context and are not relevant to an organization that has been using agile methods for a longer time. We will briefly list the issues that do seem relevant to our context.

Inayat et al. [3] find as practical challenges of agile requirements minimal documentation, budget and schedule estimation, inappropriate architecture, neglecting non-functional requirements, and requirements change and change evaluation. Many of the issue descriptions given do not lend themselves to being solved. In fact, trying to examine them through a lens of action just raises more questions.

For instance, minimal documentation is a challenge because communication lapses can cause problems to crop up [3]. But why would documentation somehow be less fallible? What and how much documentation would the affected organizations need? Have they tried making a little more documentation? Valuing working software over comprehensive documentation [7] does not mean all documentation is forbidden.

Heikkilä [2] describes as challenges associated with the release planning methods the effort spent on release planning, balancing between specialized and generalist teams, and planning the product architecture.

Heikkilä et al. [4] find perceived problems related to the requirements management processes with balancing planning effort, over-commitment caused by the product management, defining the role of Product Owners, and growing technical debt. Interestingly, the descriptions of these problems are vastly improved by also reporting how the subjects in the case organization are trying to address the problems. This shows that the subjects find the problems significant enough to try to solve, and really illustrates the problem descriptions.

However, this poses new questions. The study [4] had a two-year gap between the first and last rounds of data collection. During this time, we would expect to see significant improvements, but at least such improvements of the perceived problems are not reported. And if no improvement in the two years was made, it could suggest that the perception of the problems is incorrect.

Midgley [11] argues that observation is a specific form of intervention. All of the research on issues in agile release planning covered here is based on observation by the definition given by Midgley [11]. Midgley [11] advocates for methodological pluralism, where more explicit forms of intervention were embraced. Action research could add to such pluralism and bring explicit intervention to the topic of agile release planning.

2.3. The role of product owners

The Scrum Guide [5] says of the Product Owner (PO): The Product Owner is responsible for maximizing the value of the product resulting from work of the Development Team. Judy and Krumins-Beens [12] describe that to be successful in this role POs should share ownership of the vision, priorities, and execution with their team while remaining accountable to the outcome. The PO also needs to have unrestricted authority within their organization to make the necessary decisions for their team [13].

Sverrisdottir et al. [13] find that POs have a make-or-break role in the success of development teams, and that the role needs very skilled
people. The PO is central to providing their team visibility into the future in projects and they need to manage the expectations of and communicate with all relevant stakeholders [13].

In a large project with several teams a single PO cannot reasonably handle all the teams. In a multiple-case study Pasiviivaara et al. [6] identify patterns for scaling the role of the PO. Regardless of the scaling pattern the POs should form a PO team, each PO should communicate frequently with their Scrum team, and the PO team should have both technical and business domain experience [6].

In a qualitative interview study of 45 practitioners in different organizations Bass [14] defines nine functions that POs fill in scaled agile organizations with distributed development: groom, prioritizer, release master, technical architect, governor, communicator, traveler, intermediary, and risk assessor. The breadth of functions of the PO extend beyond the scope and skill of any single individual [14]. Bass [14] suggests forming an explicit team of POs to collaborate over the defined functions.

In a case study in an organization with 13 development teams Bernsten et al. [15] describe ways POs can coordinate with each other and their teams. The Product Owner team should have frequent meetings and workshops to share knowledge and discuss goals [15]. A clear and predefined agenda keeps such meetings as efficient use of time and resources [15]. Bernsten et al. [15] also recommend having regular retrospectives with the Product Owner team, which can help improve coordination, strengthen knowledge sharing, and build mutual respect and trust between the POs.

3. Organization background

Accountor Finago Oy is a Finnish provider of financial management software. Finago’s main product line is the Procountor software suite. The line has a dedicated development organization called Procountor development.

Procountor development consists of 13 development teams. There are about 90 people working directly in the development teams. Each team has a dedicated PO, who usually serves primarily in that role. Additionally, there are dedicated teams for operations and design, as well as various personnel in supporting or management roles.

The teams are divided into five areas titled Invoicing and accounting, Salaries and Tax Interfaces, Integrations & Business Intelligence, Mobile & Core User Functionalities, and Architecture, CI, Infra and Tools & Regression and Performance Test Automation. Each area has two to three teams, which work on things within that area.

The individual teams follow Scrum in their work. However, they have broad freedoms to adjust their own processes as needed. The teams use two-week sprints, and each has their own backlog.

The development organization is split between offices in Espoo, Finland and a near-shoring partner in Krakow, Poland. However, during the time of conducting the research most employees were working remotely.

The development organization does not prescribe to any common framework above the level of Scrum, like SAFe or LeSS. Instead, the used practices have grown organically over time.

Much of the higher-level decision making that constitutes release planning takes place at committee, at least according to the documentation. Different groups are in charge of organization wide prioritization and roadmapping, major technical decisions, and teams and resourcing.

Fig. 1 shows how requirements make their way to development. Requirements are first placed into area backlogs from where they are divided into team specific backlogs. Those are then prioritized, and the resulting work is pulled into sprints for implementation.

4. Research methods

This study used the framework of canonical action research as presented by Davison et al. [16,17]. Additionally, the work of Staron [18] was followed for ideas on the specific application of action research within the field of software engineering. Staron [18] takes a positivist stance on research. Davison et al. [16] takes a broader view, although they note that: it is impractical to develop principles and criteria that will guide all epistemological positions and indeed all forms of AR. This paper is grounded thoroughly in a constructivist view, hence the latter source was primarily used.

In this section we will first present the framework of canonical action research. Then we will cover the methods applied within each distinct phase of the research cycle. An overview of the methods for the phases is available in the results summary 5.7.

The research questions for this paper along with their more detailed explanations can be found in the results for the researcher–client agreement and action planning. The researcher–client agreement sets the goal for this entire research project which naturally translates into a research question. Since the research in action research should be inherently tied to the action that is taken, and since together with the client we wanted to go into this project with an open mind on possible actions, it seems useful that the research questions should reflect that. Hence additional questions were added during action planning.

4.1. Canonical action research

Action research is a research methodology centered around taking action within the context of the research [18]. The research actively drives change in a real-world context through action, but there can be a rather broad definition for what action means [16]. This change should be to the benefit of the organization it is taken in [16], and it should lead to the generation of knowledge or theory [18].

There are many different forms of action research of which what is called canonical action research (CAR) is followed here. Davison et al. [16] describes CAR as iterative, rigorous, and collaborative. So, the research follows a cyclical model, each iteration should be carefully planned, and the research should be conducted together with the client.

Fig. 2 shows the research cycle for CAR. Each cycle consists of the distinct phases which take place within an explicit researcher–client agreement that sets the goal, focus and scope of the research [16]. The cycles are iterated until the goals of the research are met or some other exit condition is filled [16]. The separate method sections for each phase of the cycle will explain the goals and purpose of those phases.

Staron [18] notes that iterations in action research typically take three to six months with the first iterations usually at the longer end. The schedule for the research project was six months, therefore only one iteration could reasonably be completed. The durations for the different phases for that iteration can be seen in Fig. 2.

There are inherent challenges to balancing the interests of practice and academia. For action research to work it has to be genuinely valuable for the client, and it has to result in valid research. Davison et al. [16] present five principles for CAR: the Researcher–Client Agreement, the Cyclical Process Model, Theory, Change through Action, and Learning through Reflection. These principles aim to support research that is both rigorous and relevant to practice.

Davison et al. [17] introduces the concepts of instrumental theory and focal theory to action research. Instrumental theory refers to the theoretical framework by which observations are made, data is collected, and analyzed [17]. Focal theory should guide what action is planned and explain how that action solves issues discovered during diagnosis [17]. These two concepts are distinct from one another, but clearly the chosen theories need to support each other.
4.2. Diagnosis

The goal of the diagnosis is to identify the problem that the action will address [18]. It also serves to understand the current context within which the action will be taken [18]. Thirdly, it can provide a baseline against which later findings in the research cycle can be reflected [18].

The instrumental theory during diagnosis resembled lightweight grounded theory, insofar that no existing theoretical framework was applied. Instead, the findings were grounded in the data. This reflects the desire with the client to be open about the choice of action.

Data for the diagnosis was collected using semi-structured interviews. They are a suitable method for studying perceptions on complex issues [19]. This fits the needs of understanding the release planning practices well. The interviews roughly followed an interview guide that helped ensure essential questions were covered and provided rigor over the different interviews [19]. Follow up questions were asked where interesting or new information came up.

After identifying the suitability of semi-structured interviews, forming an interview guide should consist of retrieving existing knowledge on the subject, creating an initial interview guide, pilot testing that guide, and finally using it for the interview [19]. Initial questions were drafted based on a review of literature on agile release planning. Minor adjustments were made to the guide after the first interview, which served as the piloting. The full interview guide can be seen in appendix A.

Interviewees were selected using purposeful sampling to ensure different views in the data and to maximize the richness of that data [20]. Seven people were interviewed: four developers, one architect, one PO and one Scrum master. One of the developers was from the organization’s Polish office, and the rest of the interviewees were from the Finnish office. All the interviewees were from different teams, and they covered all the different product areas of the organization.

The interviews were conducted over video calls, due to ongoing remote work guidelines during the duration of the study. Microsoft Teams was used, since it was already in use in the organization. After
going over the purpose of the interview and affirming consent for recording, the video call was recorded.

Analysis of the data was done directly based on the recordings. The analysis followed a pattern adapted from suggestions by Halcomb and Davidson [21]:

1. Recording the interview and taking field notes
2. Reflecting on the interview in the notes immediately after
3. Watching the recording and updating the field notes
4. Applying codes and updating codebook
5. Applying codes again
6. Reviewing and relistening to sections relevant for each finding

These steps were taken for each interview. Although more passes of coding were used for initial codebook development and testing. This process resulted in roughly five hours of analysis work for each hour of interview data.

The codebook consisted of a collection of codes, which each had a name/label, a description, and an example where necessary [22]. Initial codes came from activities identified in the literature review of agile release planning, defined concepts in Scrum, as well as terms and practices known in the organization. Additional codes were added during the first round of coding, where new concepts arose. Existing codes were also adjusted for clearer definitions at the same time. The first round of coding served to test and develop the codebook [23].

Table 1 shows some examples of codes from the codebook.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate</td>
<td>Evaluation of how much effort some work will take</td>
<td>Release planning literature</td>
</tr>
<tr>
<td>Confluence</td>
<td>The system used internally for various documentation</td>
<td>Organization concepts</td>
</tr>
<tr>
<td>Inter-team</td>
<td>Between teams, any reference to another development team</td>
<td>During coding</td>
</tr>
<tr>
<td>Blocker</td>
<td>Something that prevents a work item being worked on</td>
<td>During coding</td>
</tr>
<tr>
<td>PO</td>
<td>The Product Owner</td>
<td>Scrum</td>
</tr>
<tr>
<td>Refining</td>
<td>Improving backlog items to be ready to be pulled into a sprint</td>
<td>Scrum</td>
</tr>
</tbody>
</table>

4.3. Action planning

The goal of action planning is to plan the action to take during the research cycle. Additionally, the researcher and client should set the goals for that action, decide how those goals should be assessed, and prepare to collect the data to make that assessment [18]. The principle of change through action states that the planned actions should address the hypothesized causes of problems identified in the diagnosis [16]. How the action addresses the problems should be guided by the focal theory [17].

The first step for action planning was to form the action team. A call for volunteer participants was made when the results of the diagnosis were presented at a weekly organization wide meeting. This same call was also made within internal communication channels. Both the researcher and the client should be motivated to improve the situation for good action research [16]. We had two different options for possible targets for action from the diagnosis results (Section 5.2), which helped with this, as the option with more interested participants could be picked.

Next, the research proceeded with meetings with the action team. These meetings functioned as focus group interviews [26], where the practitioners in the team could express their views on the subject at hand. In total the action team had two of these meetings. Both were held over Microsoft Teams and lasted an hour.

The first meeting served as a kick-off for the action team. The agenda consisted of discussing the results of the diagnosis, any ideas raised from those results, and the possible actions that could be planned. This meeting was recorded for reporting purposes.

The second meeting had a more specific agenda to form a consensus on the followed points:

- Findings: What could be de-prioritized?
- Ideas: What could any achieved freed time be used for?
- Methods: What Action to take to create meaningful change?
- Goals: What is the goal of the Action for the organization?
- Evaluation Criteria: How do we judge whether or not the Action achieved its goals?
- Focal Theory
- Research Questions

Most of these items for the agenda come directly from the goals for action planning [18], with the first two points continuing on discussions from the first meeting.

The collected data for this phase did not undergo any formal data-analysis since results are based on consensus building with the action team.

4.4. Intervention

During the intervention the planned action is taken in the client organization [18]. It is important to report the details of how the action is applied, so that readers can judge the impact of those details on the final evaluation of the actions. Relevant data about the intervention should also be collected to support the evaluation [18].

The actions come from the results for action planning (Section 5.3), where a more thorough description can be found. The planned actions for the intervention consisted of new meetings, Scrum of POs and Retro of POs, and a rough guideline to follow for individual POs. The new meetings required participation from all of the POs in the organization, whereas it was agreed that the guideline would be tried by the three POs in the action team.

The practical organization of the meetings was planned together with one of the internal agile coaches within the client organization, who also served as facilitator for the meetings. A few minutes were reserved at the end of the meetings for feedback discussion. This feedback was focused on the practical organization of the meetings—whether or not changes are needed to the structure or goals of the
meetings and if the facilitation went smoothly from the perspective of meeting participants.

Data collection of the intervention consisted of participant observation of the meetings. To ensure necessary observations were recorded a prepared data form was used [27]. The form is available in appendix B. The focus of the data form follows from what data was decided to be needed for evaluation during action planning. Analysis of the observation data is conducted as part of the evaluation.

4.5. Evaluation

The purpose of evaluation is to understand the impact of the action on the organization [18]. The goals for what to understand were set as part of the action planning.

Data was collected with semi-structured interviews. They are a good tool to gage the experiences related to the action from participants [20]. Additionally, the participant observation data collected during the intervention was used. The interviews followed an interview guide [19]. Additional follow up questions were asked where more depth was needed or new points of interest emerged [20]. The interview guide was formed based on ideas what information is needed for evaluation discussed during action planning. The full interview guide can be seen in appendix C.

Interviewees were selected for different perspectives on the effects of the action. Two POs from the action team, and one developer from each of their Scrum teams, as well as two POs who did not participate in the action planning were interviewed. The interviews lasted about fifteen minutes each and were recorded after asking for consent for later use in data analysis.

Affinity diagramming was used for the evaluation data analysis. Lucero [28] describes affinity diagrams as a good way to make sense of unstructured or seemingly dissimilar qualitative data and suggests a process for forming the diagram consisting of four stages: creating notes, clustering notes, walking the wall, and documentation.

Creating notes means lifting data from observations and interviews onto sticky notes [28]. The observations from the intervention phase were fit for making notes as is. For the interviews notes were made listening to the recording and taking a note on anything noteworthy. To help with that assessment the observation guide in appendix B was repurposed. Each created post-it note had an identifier for its data source to ease returning to the source from the wall.

Clustering the notes means placing notes on the wall one at a time with similar notes close together [28]. Walking the wall consists of discussing and pruning the forming categories on the wall and continuing to add notes to the categories [28]. With a single researcher, these two steps were somewhat fused as one.

Once the categories are finalized, documentation consists of taking pictures of the wall, writing the descriptions of the categories into an appropriate form, and presenting the results to relevant stakeholders [28].

Following the principle of theory, the evaluation of the intervention should be informed by theory [16]. Since grounded theory was used for the instrumental theory during the diagnosis, it cannot readily give an objective evaluation on the success of the intervention. The evaluation should still be checked against the focal theory [17].

4.6. Learning

The learning phase should be used to reflect on the previous cycle of research. Following the cyclical process model an explicit decision should be made on if additional cycles are needed, and if not the justification for the conclusion of the project should be clearly reported [16]. According to the principle of learning both the researcher and the client should reflect upon the outcomes of the project [16]. Before moving to a possible next cycle reporting and documentation for the current cycle should be completed both targeted at academia and for practitioners [18].

5. Results

The results section is broken down following the structure of CAR. First, the researcher–client agreement establishes goal and focus for the research project. Then there are sections for the results of each of the individual phases of the cycle. And finally, there is a summary of the results.

5.1. Researcher–client agreement

Davison et al. [16] says that it is important to document and report how research with clients is agreed and what that agreement is. We had wanted to try research at Accountor Finago Oy prior to the beginning of the study. The topic came from some general dissatisfaction with how work is planned in the development organization.

We chose action research, because we hoped it would provide tangible improvements for the organization. The other option considered was a case study. However, we worried the results of a case study would not lead to improvements even for ourselves, let alone for anyone else.

We agreed on a research project with the admittedly vague goal of making some improvement to our release planning practices in Procountor development. This goal translates into an initial research question:

RQ1: What issues can exist in a software development organization’s agile release planning practices?

The general gist of the project was presented to relevant managers, and it received a green light to go ahead, with a set schedule of six months for the entire project.

5.2. Diagnosis

The diagnosis affirms the original suspicion that there is something not quite right in the organization’s release planning processes. Based on the interviews we ended up with the following diagnostic findings:

1. Heterogeneity between teams
2. Mostly independent teams
3. Difficulties lie in scaling over time
4. MVP process does not serve its intended purpose
5. POs are a funnel
6. Day-to-day workflow is detached from release planning

Notably these are not an exhaustive description of the release planning practices in the organization or the issues therein. They are focused more on things that directly affect the choice of action.

Here we will elaborate on these findings in more detail. Then we will present some intermediate conclusions based on those findings, and two initial suggestions for possible actions to pursue.

Heterogeneity between teams

Specific practices for planning work vary a lot across teams. For instance, how developers participate in drafting backlog items was different between different teams. Some developers described receiving ready tickets, that could be pulled into a sprint as is. Other developers said their team gets assigned larger epics, which they then break down into tickets that fit into sprints together with the PO. Interviewees also reported different orders of what are called sprint planning part one and part two in Scrum.

The differences in work practices are not surprising. The teams work on really different things. Practices suitable for API development will be different from those for a mobile application. Which in turn will both be different from batch jobs and integrations with banks. The teams have had the flexibility to adjust their process to their needs, which they have used.

The takeaway is that there is variance, which is assumed here for a valid reason. Any planned intervention will have to accommodate for that flexibility. Any rigid and forcing model would likely be met with opposition.
Mostly independent teams

Assignment of work to teams and teams to projects follows pre-existing set areas of responsibilities. Teams also have ownership of existing features if something in those features need to be fixed or future changes need to be made.

The division into areas works well enough. Or at least interviewees reported being mostly satisfied with it. The general view was that each area requires specific expertise, so that jumping frequently between areas would not be practical.

The area division also keeps teams from stepping on each other’s toes. Teams do not often need to directly coordinate sprint work with other teams and when they do it is usually with only one or two other teams at once that are working on the same project. So, mostly the Scrum teams are able to work independently.

For the level of coordination that does take place several different practices came up. Existing practices include cross team code reviews, attending other team’s Scrum events, as well as shared communication channels and regular meetings.

When problems with coordinating work between teams came up in interviews, they were always followed by a description of how those problems were identified and what changes were subsequently made. For instance, one developer had had their own work blocked by something they believed another team was working on, when in fact the other team did not even have a ticket for it. This conflict had reportedly been identified, and relevant changes made to shared practices.

The downside of these fixed areas is that resources are a bit set in stone. The loss in flexibility means the organization cannot dogpile resources on specific projects.

Ultimately rigid areas are not necessarily good or bad. However, we suspect that any changes to that structure would not be met favorably. This needs to be considered in the planned action. It also means that an action can be applied to individual teams without significantly impacting the work of other teams.

Difficulties lie in scaling over time

If we consider the two directions projects can scale, over time or over resources, scaling over time seems to cause more trouble for the client. It is not really one clear thing, more so a bunch of small sources of friction.

Developers felt they occasionally did not have enough visibility into upcoming work to successfully design solutions to problems worked on earlier. Instead, when new requirements come up, they require unnecessary rework of the earlier solution.

These little things build up on long projects, which leads to delays in schedule. Project scopes and schedules are not flexible enough or do not reflect this part of how projects go. Instead, the projects end up with tails and with developers feeling frustrated.

Projects have schedules and deadlines that come from somewhere higher up. Often these do not reflect the actual effort needed for the project. Developers often do not know where and how these schedules are set. They do not directly participate in setting those estimates. This lack of visibility to decision making according to one interviewee leads to speculation and the speculation is never flattering.

In short, release planning does not support those plans being followed through. A lot of it is just a vague sense of unease with regards to release planning across the development organization. And surely some of it is unavoidable, release planning is a complex undertaking which is never going to be perfect.

MVP process does not serve its intended purpose

The process for making minimum viable product (MVP) versions of projects was the most common answer when interviewees were asked to identify problems with existing release planning processes. In practice it often does not seem to serve its intended purpose and leads to its own problems when used.

Several participants expressed dissatisfaction with how projects would have milestones for an MVP, but in reality, it functions as an excuse to cut corners. Then once the MVP is released further work on the project is not done, since the schedule and the budget for the project are finished. So, the end result is being left with less maintainable code.

From the view of release planning this seems like a symptom of the underlying issues. It will not be explored further within this paper.

POs are a funnel

POs are the central medium through which most information on the content and the context of backlog items is passed to developers. Knowledge of what should be done, background information on where these items come from, and why they are being pursued, are all funneled through the PO.

This is consistent with Scrum, and not necessarily a bad thing. However, there is a certain risk with placing all your eggs in one basket. And as the organization scales funnels risk becoming bottlenecks.

This also leaves POs with a lot on their plate. They have to explain and justify a really broad and complex view of things. They become the arbiters of truth, who not only have to know and remember all the details but are expected to judge what is relevant for whom.

One developer noted that if the PO gets overwhelmed the development process breaks down quite quickly. Which just emphasizes the key role POs have in how we do Scrum.

POs need to have had time to think about answers to questions. One interviewee felt that when the PO answers questions about backlog items, the answers are much more useful, if the PO has already considered that viewpoint somehow prior. This suggests that overburdening POs may lead to unexpected symptoms within release planning even before there are any obvious problems.

Development workloop is detached from release planning

If we consider the regular activity that makes up most of the development work: pull backlog items into sprint, break down into tasks, complete tasks until the item is done, that entire cycle is somewhat distant from the organization’s release planning practices. This workloop is detached from release planning. It seems entirely possible to work in development without ever having to directly engage with the big picture. Put simply backlog items and how they are communicated do not feature long-term structure.

This is not a property unique to our organization. Common definitions of user stories say that they should be independent [29]. Similarly, the client has guidelines for JIRA tickets that reflect this. Again, this is not necessarily a problem, but is worth considering.

From the developer perspective decisions for any long-term view happen up there somewhere. Visibility into where, why, and how the decisions are made is quite limited.

Diagnosis conclusions

In summary, it seems like a good idea to plan and try out the action for this research for some limited part of the organization. Since the teams are quite independent in their work this can be easily done, and focusing on a few teams at once will allow tailoring the solution to fit the needs of those teams if need be.

The principle of change through action states that the problem and its hypothesized causes should be specified as a result of the diagnosis [16]. Two hypotheses for possible contributors to the problem emerged that formed initial suggestions for a focal theory:

1. In a situation where the role of POs has been strained, freeing up time for POs will make the agile release planning process work better for everyone.
2. Building in a long-term view into the content of how backlog items are documented and communicated will increase the presence of a long-term view in the associated implementation work.
Both of these theories came with a paired pitch for an action to address them and to test them out. The first would be to go through POs’ role, tasks, and schedules and to ensure that enough time is available. The second would be to update our conventions for backlog items and take those conventions into use.

5.3. Action planning

The action team was formed with three POs and one designer. This team formed around the first suggested theory from the diagnosis results, which piqued more interest within the organization. The results of the meetings with the action team are in line with the theory built in the diagnosis and the findings of that phase. Some updates were made, and the revised focal theory was:

As the Organization and the scale of the service have grown over time the role of POs has become strained, which in turn contributes to friction and inefficiency within how our release planning practices work. So, we should be able to achieve more by freeing up time and cognitive load for POs and by pushing POs to do less.

This focal theory is the central guideline for both the research and the action here. The clarification of also needing to consider cognitive load is to rule out just doing the same things faster or more efficient as a solution. There is a limit to how far that line of thinking can go and stretching toward that limit risks burnout and other harm. In general, we would consider that unethical, hence the clarification. The inclusion of pushing less comes from findings that some of the strain on the role of POs can come from external pressures.

In the meetings with the action team, we identified various issues related to this focal theory. To address these issues, we decided on a three-part action plan:

1. Time-boxing guideline for own planning work
2. Scrum of Product Owners
3. Retro of Product Owners

where the first item would be tried out by the POs on the action team. The two other parts would require participation from all POs.

The time-boxing guideline for own planning work means setting aside and reserving in their own calendars time for specific tasks. For all sprint planning and backlog refining sessions the POs should reserve at least the length of the session worth of time for preparing for those sessions. That preparation can be used for going over materials, reading, writing, really anything that seems useful.

The Scrum of Product Owners will be a meeting similar to the Scrum daily, except held weekly and for POs from all teams. The goal is to contribute to regular knowledge sharing on who does what and who knows what, and to identify and address impediments in a timely fashion.

One of the sources of strain discussed in the action team meetings was having access to information. Individual POs do not have a full picture of all parts of the product and the work being done on it. Getting relevant information when it is needed can be a problem. Passive channels for getting help from peers exist but are not adequately in use. For instance, often asking questions in Slack channels does not lead to answers, so it does not feel worth the effort to even bother. Whereas asking for help directly is sometimes limited by not knowing who might be able to help. The Scrum of Product Owners can hopefully help with this.

The Retro of Product Owners similarly pulls influences from the Scrum retrospective. The participants will be all the POs and the meeting will recur monthly. The goal is to create and implement action points to improve working processes and conditions among our POs. The action points should be formed by identifying impediments and supporting factors for fulfilling the PO role during the Retro.

The need for the Retro is apparent in both the focal theory as well as the discussions with the action team. The focal theory we are using suggests that there is a need to have a way to push back against sources of strain. And if that theory is true, as the organization continues to grow, POs will need active channels to adjust how that growth affects them.

Many of the sources of strain discussed during action team meetings were separate practical things in our processes. But they were not such that the action team could address them directly, instead rough edges should be smoothed out when they come up. However, there has not been an active channel for continuous improvements to processes for POs collectively. Hence the Retro.

The main goal of the organization for these actions is to find out if there is actual merit to this focal theory, which could suggest the need for further structural changes in the future. Each of the actions should also reach their own goals.

To assess whether this cycle of action reached its goals we will need to be able to answer the following questions:

1. What changes did the practicing POs notice?
2. What changes did their team notice?
3. What action points did the Retro lead to?
4. What impediments were identified in the Scrum?

Answering these questions will be part of the evaluation phase and details on how these questions are approached can be found in the methods for evaluation.

Additional research questions for this paper were also formed at this stage. A two-part question was added on top of the first research question presented in the researcher–client agreement. The two parts are as follows:

RQ2: (a) What are the perceived effects of introducing practices to counteract strain for POs?
(b) Is there an improvement in agile release planning?

The first part asks what the effects of the planned actions are. All noted effects should be considered as to try and avoid confirmation bias towards the overall goal of this research project. The second part is there to acknowledge the fact that the introduced practices come in the context of trying to improve agile release planning in the organization. That will inevitably affect the data collection and analysis needed to answer either part of the question.

5.4. Intervention

The first step in the intervention was to present the results of the action planning to the organization. All POs in the organization were sent calendar invitations to the Scrum of Product Owners and the Retro of Product Owners. Accompanying the invitations were pages in the organization’s internal wiki that explained the purpose and structure of the meetings. These pages can be seen in appendices D and E.

Two instances of the Scrum of Product Owners and one instance of the Retro of Product Owners were held as part of the intervention. Initial feedback for both the meetings given at the end of the meetings was quite positive. The participants felt that it was easy enough to take part, and that the structure served the goals of the meetings. This is important since it means that those goals can be reasonably evaluated.

The time-boxing guideline was tried by the POs in the action team. They were asked to set aside dedicated time before each Sprint planning and Backlog refining for preparation. This preparation time could be used for going over materials, reading, writing, structuring thoughts, or anything else the POs deemed relevant. The important guideline was to reserve as much time for preparation as the length of the corresponding meeting. At this stage there is nothing to report on the reaction to the time-boxing guideline yet. That information will come as a part of the evaluation.
5.5. Evaluation

The analysis of data collected during the intervention and from interviews for evaluation resulted in the following categories:

1. Business as usual
2. Domain competence
3. Team resources
4. Fragmentation and ownership
5. Role definition and self-efficacy
6. Pressure, instability, and stability
7. Time management
8. Team spirit
9. Action Feedback

The affinity diagram with the categories had a total of 127 notes. We will explain each of the categories first and then summarize the conclusions of the evaluation next.

Category: Business as usual

This category contains notes on seemingly normal orders of affairs. POs have been prioritizing backlogs, communicating with stakeholders, following up on feedback on new features, and so on. Developers are proceeding with sprint work like ironing out bugs for an upcoming release and implementation in general.

Various little hitches and minor impediments exist. These include slight miscommunications in requirements and synchronizing work with external stakeholders. These were also things that participants felt could be addressed, for instance in development team retrospectives.

All of which seems like fairly normal application of Scrum. It is also in line with results from the diagnosis, and fits with the generated focal theory well.

There has not been significant direct change attributable to the actions taken as a part of this cycle of research. Which is not to say that change has not happened since the diagnosis. On the contrary, participants reported that their teams have made changes into how they plan their own work recently. Also, the development teams are working in different phases of projects, or on all together different projects than they were several months earlier.

Category: Domain competence

This category contains notes related to the building, sharing, and maintaining of knowledge that POs need for their role.

The most common type of impediment observed during the Scrum of Product Owners meetings was problems related to access to knowledge. So, any PO might lack information on some specific subject matter or existing functionalities in the service, or they might not know where that information is available. The other POs were usually readily able to offer help with these, so it seems the Scrum of POs may be able to ease impediments on access to knowledge.

When moving to different areas of the service or starting new projects, there seems to be a significant time expenditure for picking up the needed knowledge for the PO to successfully perform in their role. Even more so for newly starting POs in the organization.

Category: Team resources

An observed topic during the retrospective and Scrums was issues with team resources. If a developer is away for some reason or leaves the company, teams need to react to it. However, this does not really inform the focal theory or evaluation one way or the other, so there is no need to discuss it further.

Category: Fragmentation and ownership

Many parts of the organization’s planning processes and different responsibilities were split between multiple people or committees. This fragmentation made it more difficult for POs to reason about those processes. It was felt that often times it was not clear who had ownership or accountability of those processes and responsibilities.

Taken together the fragmentation and lack of clear ownership make it more difficult to make improvements to the organization’s processes. Some POs noted that they had previously individually brought up issues with relevant managers, that were now being targeted by action points from the retrospective. However, the previous attempts at bringing up issues had not led to anything.

The POs felt that the collective nature of the retrospective could help highlight priority improvement needs, and that the structured commitment to those improvements would be more likely to lead to change. Handling this challenge of fragmentation and ownership seems like something that would get more difficult as the amount of people in the organization grows.

Category: Role definition and self-efficacy

Within the scope of the fragmentation of responsibilities in the organization the role of the PO had also become somewhat unclear on the edges. For instance, the POs had used to participate in roadmapping, but no longer did. Not participating in such long-term planning was viewed to make their work harder.

Not being able to participate in setting goals and not contributing to decisions on how those goals are reached was considered demotivating. The POs enjoyed being able to suggest improvements for their own working practices and felt that the action points formed during the retrospective were relevant for their work.

Category: Pressure, instability, and stability

This category contains the bulk of external strain noted in the role of the PO. The external strain consists of pressure to push out new features and instability in priorities and overall goals.

POs felt that the pressure to push out new features was detrimental to their work. New projects starting on top of old projects means both difficulties in wrapping up tails and insufficient preparation. Overall, it was felt that this contributes to lower maintenance code.

POs reported that instability of priorities and goals caused a loss of motivation for teams and inefficient work. Sudden changes to priorities undermine trust in how those priorities are set. Inversely, stability is enjoyed for providing sufficient time to concentrate on the things at hand. Unpredictable goals do not provide time for POs to prepare for projects and pick up necessary domain competence. This in turn can lead to development work feeling scrambled.

This is not to say that pressure for new features or changing goals are inherently bad. Some pressure to get things done is surely a part of a competitive market and adapting to change rather than sticking to plans is a core tenet of agile methods. So, the point is not that these should not be done. The question is instead how to apply these constructively.

Category: Time management

This category contains notes on how POs managed their own time. As a topic it brushes up quite closely to the time-boxing action, which ultimately was not fit for purpose. Going over these notes will hopefully illuminate why that is.

There are a lot of different interests vying for POs’ time. There also is not a clear way on how different tasks for POs should be prioritized. Some POs also worried that juggling too many things would inevitably lead to some of those things accidentally being dropped. Looking at the focal theory this seems like a source of strain.

When a PO’s calendar gets filled too much, the first things to be dropped or moved aside were often their own tasks. Trying to reserve time for these tasks, however, fails to address the underlying issue
of there simply being too much on the table at once. On the other hand, when there is enough time, explicit reservations are not strictly necessary. At that point it is more a matter of personal preference. Although having some way of structuring and prioritizing their own work, was considered a good idea by the POs.

Secondly, the amount of time needed for different tasks by POs varies wildly. Interviewees noted that for instance a new epic could easily take half or even a full workday to prepare for a Sprint planning or backlog refinement. On the other hand, during the later parts of projects once the subject was already familiar, they could get by with very little separate preparation. So having a fixed rule of thumb does not really accommodate for that.

**Category: Team spirit**

This category contains notes on a sense of and the effects of team spirit between the POs.

The new meetings worked because of an air of mutual trust between the participants. Interviewees felt that discussions during the meetings were constructive and that they could take part in the discussions openly. The POs were also quick to offer help when impediments came up during the Scrums.

The new meetings also contributed to an increased sense of team spirit. The participants expressed that they enjoyed having regular shared possibilities to discuss with other POs. They also noted that they had not previously really had that possibility. Earlier they have had mainly a meeting called the monthly, which however serves more as a reporting meeting, and which does not apparently have time for meaningful discussions between POs.

**Category: Action feedback**

The category of action feedback consists of statements that form direct feedback on the actions taken as part of this cycle. POs enjoyed the clear goals and structure for both the added meetings and felt the amount of time reserved for the meetings was fitting. Feedback for the retrospective was overwhelmingly positive, whereas for the Scrum of Product Owners it was just mostly positive. The time-boxing guideline was not seen as fit for its purpose, the reason for which is explained under the category of time management.

**Evaluation conclusions**

In summary, the Scrum of Product Owners and the Retro of Product Owners are worth keeping. The time-boxing guideline was not fit for purpose and as such does not merit taking into broader use.

The Scrum meetings uncovered impediments, which also prompted reactions to help address those impediments from other POs. Interestingly the POs mentioned the knowledge sharing aspect of the Scrums more as a benefit, they felt it useful to regularly hear what their peers are working on.

The retrospective led to action points that the POs felt were meaningful and relevant for their work. Moreover, the meeting responded to a genuine need to provide a way for driving continuous improvement, that is active, collective, and structured.

The main goal for all these actions was to find out whether there is merit in the focal theory developed earlier in this study. The answer would seem to be yes. The various sources of strain will need further work to be addressed.

The results of the evaluation do not directly contradict the focal theory developed earlier. However, they do reinform and provide insight to many parts of it. Especially the idea of what strain on the PO role is and what effects the strain has seem much clearer now.

### Table 2

<table>
<thead>
<tr>
<th>Phase</th>
<th>Key methods</th>
<th>Key results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnosis</td>
<td>Seven semi-structured interviews, grounded theory</td>
<td>Identified relevant findings in agile release planning including POs being central to effective following of release plan and that challenges are in planning work over time. Formed initial focal theory of PO role strain.</td>
</tr>
<tr>
<td>Action planning</td>
<td>Two action team meetings</td>
<td>Decided on actions of time-boxing, Scrum of POs, and Retro of POs. Refined focal theory.</td>
</tr>
<tr>
<td>Intervention</td>
<td>Participant observation</td>
<td>Actions were successfully taken so they can be meaningfully evaluated.</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Six semi-structured interviews, affinity diagramming</td>
<td>Time-boxing was considered not fit for purpose. Scrum of POs and Retro of POs got good feedback and are continued. Identified sources of PO role strain, challenges with changing priorities, and difficulty of continuous improvement of agile release planning at scale.</td>
</tr>
<tr>
<td>Learning</td>
<td>Reflection</td>
<td>Considered topics for further cycles. Concluded research project for scheduling reasons.</td>
</tr>
</tbody>
</table>

### 5.6. Learning

Learning began with presenting the evaluation results to the client organization. During the evaluation the taken action was assessed against goals set as part of action planning, which were mainly reached. There is still the broader question on how the overall goal for this research project has been achieved set as part of the researcher–client agreement. That goal was to make some improvement in the organization’s agile release planning practices.

As noted in the evaluation results there has not been significant direct change attributable to the actions taken as part of this cycle of research. So, if the standard for what counts as an improvement in planning were getting more or more valuable work done, then no the goal has not yet been reached. Still, we think the Retro of Product Owners and the Scrum of Product Owners are steps in the right direction on that front and serve a legitimate need in the organization.

There are still many things left to improve. For instance, the evaluation results contain many open questions that would be useful for practitioners and interesting for research to look further into. Similarly, there was a second option of an action to pursue at the end of the diagnosis, which could still be explored. Due to scheduling reasons further cycles were not performed.

The research project received positive feedback from research participants and managers. Surely this is partly due to the collaborative nature of canonical action research. There was also an open and accepting atmosphere towards change, which should hopefully be true for any agile software organization.

### 5.7. Summary

To give a more concise overview of the results of this research cycle, it is good to go over the outcomes of the different phases again in brief. Table 2 shows a summary of the key results for the different phases. Each phase added to the understanding of the issues in agile release planning being studied in their own way.

### 6. Discussion

#### 6.1. Answers to research questions

**RQ1: What issues can exist in a software development organization’s agile release planning practices?**

Defining the role of the PO can be an issue, and as the organization and service scale that role can become strained. Sources of strain
include balancing different interests, prioritizing their own tasks and time usage, and responding to pressure to push out new features. Heikkilä et al. [4] find that defining the responsibilities of POs is a challenge in organizations that have recently adopted agile. We add that it is still a challenge in a more established agile organization.

A lack of clear priorities and fluctuating goals can cause problems. Changing priorities can erode confidence in how those priorities are set leading to lower motivation and commitment to those goals. Quickly changing goals also do not provide POs enough time to pick up necessary domain competencies to perform their role. Paasivaara et al. [6] has noted that clear priorities are needed to successfully scale the PO role.

The inherent complexity of agile release planning at scale is in itself a problem. The fragmentation of different roles and responsibilities, and the lack of clear ownership of different parts of the process together make it more difficult for people participating in the release planning and users of the subsequent release plan to suggest improvements to the process.

This lack of self-efficacy is the result of this study most particular to action research. One of the actions taken was found to be good, because it improved participants chances to affect their working context. This implies that prior there was a lack of such possibilities. Taking the action provided a unique opportunity to study the underlying issue.

The strongest result of this study are the ideas regarding strain on the role of POs. It was looked at one way or another during each phase of the research cycle. And for every phase the view of that strain gained a little more clarity.

**RQ2: What are the perceived effects of introducing practices to counteract strain for Product Owners? Is there an improvement in agile release planning?**

The introduced new practices take time and effort. The practices that the organization decided to keep are two meetings called the Scrum of Product Owners and the Retro of Product Owners that take 30 min a week and 60 min a month, respectively.

The Scrum of Product Owners had the stated goal of sharing knowledge between POs and addressing impediments in a timely fashion, both of which it managed to do. The participants felt that hearing what other POs were working on was interesting and useful for themselves. Berntzen et al. [15] describe a weekly coordination meeting between POs in an organization that also had 13 POs, which had very similar goals to the Scrum of POs. However, this coordination meeting did not have a defined agenda [15], whereas the Scrum of POs had a more set structure.

The Retro of Product Owners was aimed at generating action points that lead to meaningful change in the ways of working within the organization. The meeting led to action points that participants felt were meaningful and relevant for their own work. Whether or not those action points result in change remains to be seen.

POs enjoyed having meetings focused on discussions between each other. The participants felt that it was useful to hear what other POs were doing, and that having shared activities added to a sense of the POs being a team. Berntzen et al. [15] find similar benefits of regular meetings between POs in their studied organization. Both Paasivaara et al. [6] and Bass [14] emphasize the importance of the POs forming a team.

The introduced practices were added as part of a project trying to improve agile release planning within the organization. As to the question of whether there was an improvement the short answer is no. If we take the stance that the metric for better release planning is getting more work done or more valuable work done, then there has not been improvement by that metric yet. The introduced practices are instead more tools to support slow and gradual change, the effects of which would require further research.

If we consider a lack of practices for improvement of a process to be an issue of the process itself, then there has been an improvement in agile release planning. POs were confident that the Retro could lead to real change and enjoyed having a channel to bring up things that were issues for them.

There is also the likely but indirect effect of the Retro of Product Owners. It seems quite clear that POs have a central role in how the release plan is communicated to developers and followed. Hence many things that are issues for POs are issues in agile release planning.

### 6.2. Limitations

The results of this study are quite context dependent. Few claims can be made on the generalizability of the findings. Similar ideas may be true for other organizations around the same size that employ a similar way of scaling Scrum.

The traditional way of reporting on threats to validity is considered unfitting for constructivist research [30]. Dittrich et al. [31] propose some criteria for evaluating the quality of qualitative research. Creswell and Creswell [32] proposes some strategies for improving the validity of constructivist research. We will discuss the ones that require separate reporting here.

To start with the good news, both Creswell and Creswell [32] and Dittrich et al. [31] suggest the use of triangulation for better qualitative research. The use of canonical action research in this study had a strong element of triangulation built in. The different phases look at the same topic with variation in methods, and the actions themselves serve as a probe of the issues being studied.

Possible biases should be clarified openly [32]. Staron [18] notes that there is a risk of bias particular to action research from the engagement of the researcher and participants to the actions. Put simply, an action researcher would like for the actions taken to be successful, and we do not believe the effects of that can be fully nullified.

Additionally, there is the more general risk of researcher bias having only one researcher perform data collection and analysis. The methods and results have been described in verbose detail so that the reader can evaluate the effect of this.

One way researcher bias was counteracted was with member checking, which is recommended as a practice by Creswell and Creswell [32]. With the emphasis on collaboration this is built into canonical action research. The results of each phase were presented to the organization at the end of the phase. Members of the organization had a clear stake in ensuring the truthfulness of the results, since incorrect results could have led to unfavorable actions for them. This effect can be seen, for instance, in the changes to the focal theory between diagnosis and action planning results.

Perhaps the biggest limitation of this research is one of scope. There are many perspectives that are not present in the research, even though they would surely be relevant for understanding the agile release planning practices of the organization. The research explicitly does not answer the question of what agile release planning in this organization is. Someone might think it odd to identify issues in a system, without fully understanding what that system is, but that is what has been done.

The diagnosis also did not proceed to any level of saturation, and there was only a single cycle of action. That means that the issues reported in the answer to the first research question are by no means a complete list of all the issues within the agile release planning practices of the organization. They are just some issues that have so far been identified and understood.

Finally, the results for the individual phases should not be taken in isolation. The amount of data collected and analyzed in any single phase was not sufficient to build such confidence in the results, nor was it intended to. Instead, the meaningful results arise from ideas built over the entire research cycle. That is why results from the diagnosis that were not relevant for the future choice of action are largely absent from the discussion.
6.3. Suitability of applied methods

The principle of learning through reflection instructs that results should be considered in terms of the general applicability of canonical action research [16]. The use of instrumental and focal theories, as well as their interaction with each other, should also be reflected on [17]. Dittrich et al. [31] also suggests that qualitative research in software engineering should reflect on the applicability and the usefulness of the chosen research design.

In general, the use of canonical action research worked well for us in this project. The 5 principles and the 31 accompanying criteria introduced by Davison et al. [16] structured and guided the research project well. The framework helped make balancing scholarly and practical interests easy enough.

Before the beginning of the research projects the option of a case study was also considered. With the benefit of hindsight, the choice of action research was the better choice for the client organization. After all, the research resulted in relevant change in the organization.

The results of this study would not have been achieved by a different methodology as is. Much of those results is built on the immediate reaction to the actions taken and would not have been visible by way of case study for instance. Similarly, having to justify and evaluate the actions taken challenged the ideas of issues found in this study. The distinct use of focal and instrumental theories played a strong role in guiding the academic output of this paper.

The downside of the choice of action research can be seen in the narrowing of the research scope after the diagnosis. Instead of moving to other phases, we could have used the following eleven weeks for further rounds of data collection and analysis. In that case we would have likely had a broader understanding of issues in agile release planning within the organization. This seems to be a tradeoff between action research and case studies.

The focal theory should be selected informed by the instrumental theory [17]. Since grounded theory was used for instrumental theory during the diagnosis, and the developed theory there was used as the focal theory, it is quite straightforward to see how one led to the other. It is nevertheless interesting to consider how the details of how the diagnosis was conducted influenced the focal theory and the remainder of this research project.

The diagnosis was built on individual in-depth interviews with a focus on different activities that fall under the umbrella of agile release planning. The resulting findings and the action taken based on the results of the diagnosis was centered around individual agency. The point here is, that had we together with the client wanted to change, for instance, the structure of decision making that goes into the organization's agile release planning, then the data collection and analysis should have been built around understanding what that structure is.

Davison et al. [17] views that the focal theory should not be tweaked, adjusted or changed based on the outcome of the intervention. We disagree. The immediate reaction to the action provides a unique probe into the underlying issues that are being addressed. It would seem foolish not to take the insight into account while theory building. Of course, care must be taken in analysis that any updates to the focal theory do not contradict the data collected before the action. But this can be achieved by re-examining the relevant parts of that earlier data, similar to how updates to generated theory would be formed after additional cycles of data collection and analysis typically in grounded theory.

The continued development of the focal theory throughout the research cycle maybe a special case of combining grounded theory with canonical action research. The specific combination merits further discussion and methodological development.

7. Conclusions

7.1. Implications to practice

This paper describes two practices in detail, the Scrum of Product Owners and the Retro of Product Owners. Organizations where the issues identified in this paper sound familiar may benefit from taking these practices into use. For a Product Owner team to be an actual team they should have shared practices and activities. Regular chances for POs to discuss with each other can help with this. For POs to be able to achieve continuous improvement in the ways that they work, they need an active, collective, and structured way to drive that improvement.

Organizations should also be mindful of their POs. As agile organizations grow the role of POs can become strained. Since POs have a central role in how agile release planning works, this strain can have a negative impact on the performance of the entire organization. Special care should be taken so that POs have working conditions in which they can work to the best of their abilities.

Furthermore, this paper demonstrates that canonical action research can be used to the benefit of software development organizations. Working in cycles and seeking to actively improve ways of working should be familiar concepts for any agile organization. Many of the methods used here could be used by any agile organization to help their own efforts in continuous improvement.

7.2. Implications to theory

This paper provides a rich and detailed description of the use of canonical action research in the context of software engineering. It adds to the body of knowledge on agile release planning and the role of POs in scaled agile organizations, building on top of works such as Berntzen et al. [15], Heikilä [2], Heikilä et al. [4], and Paasivaara et al. [6].

This is to our knowledge the first study to include issues in agile release planning by way of action research. The rigorous effort to mitigate or solve some of the found issues gives a new perspective and more depth to the understanding of those issues. The issues identified include strain on the role of POs, the effects of changing priorities, and the difficulty for different parties to suggest improvements to how agile release planning is done.

7.3. Future research

As is the case with any research conducted in a single setting, the ideas explored in this study should be looked into in other organizations as well. There are also plenty of natural continuations for the research.

Perhaps most immediate would be prolonged observation of the introduced new practices within the organization. Especially interesting will be to see the future development of the Retro of Product Owners. What changes will it lead to? What problems is it able to address? Answers to these questions would likely be useful additions to the broader literature.

Topics for further cycles of research were considered as part of the learning phase. One is how to apply flexible priorities constructively for the development organization. Another is how to make requirement artifacts like user stories better support a long-term view of development work.

CRediT authorship contribution statement

Konsta Kantola: Conceptualization, Methodology, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Supervision. Jari Vanhanen: Conceptualization, Resources, Supervision.
Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to https://doi.org/10.1016/j.infsof.2022.106900.

The research was conducted under employment of the studied organization Accountor Finago Oy.

Appendix A. Release planning interview guide

Intro

Explain the practicalities of the interview and why it is being done. Then ask for permission to record the interview and remember to start recording.

Background

• Who are you?
• What role do you work in?
• What team do you work in?
• How long have you been at the company?

Team level working practices

Goal is to find out what day to day/within sprint work is like

• Do you work following Scrum?
• How do you decide what work comes into the sprint?
• How are those work items communicated?
• How can you get more information if needed?

Participation and familiarity with existing release planning

Goal is to understand how the previous work relates to higher level goals

• Where do the work items come from?
• Do you affect work items before they are pulled into the sprint? What ways can you interact with them?
• How about before items are placed in the backlog?
• How is it decided which team gets what work items into their backlog?
• How is the effort for these work items estimated?
• How are schedules for work set?
• Does the work of other teams affect items you are working on? Or your work to others? How?
• How is work coordinated between teams in these situations? How do you interact?

Opinions

Goal is to understand how they feel about the current processes

• How well do you feel the existing release planning processes support your work?
• Prompt for opinions on the following topics
  – Deciding what should be done? Prioritizing?
  – The requirements artifacts ie JIRA tickets, user stories?
  – Deciding how work should be done? Larger architecture?
  – Deciding who should do what? Splitting work between teams?
  – Estimating and scheduling?
  – Communicating upcoming plans and work? Knowledge sharing?
• What would you like to see improved?

Appendix B. Observation data form

Observe and make notes of the following:

<table>
<thead>
<tr>
<th>Key</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td>Discussion Topic</td>
</tr>
<tr>
<td>I</td>
<td>Impediment</td>
</tr>
<tr>
<td>R</td>
<td>Reaction to impediment</td>
</tr>
<tr>
<td>AP</td>
<td>Action point</td>
</tr>
<tr>
<td>C</td>
<td>Change</td>
</tr>
</tbody>
</table>

Appendix C. Evaluation interview guide

Explain the context of the interview, ask for permission to record, start the recording

Questions for developers

• What have are your team currently working on? What were you working on during the previous sprint?
• How do you feel about your current work items? How were they communicated to you?
• How did your previous Sprint Planning go? And backlog refining?
• Have you noticed changes in how planned work is communicated to you recently? What?

Questions for Product Owners

• What was the Scrum of Product Owners like?
• Did it feel like a useful use of your time?
• What was the Retro of Product Owners like?
• Were the discussions and the resulting action points there relevant for you?
• Did it feel like a useful use of your time?

These questions only for Product Owners who practiced the time-boxing

• How much time did you use for the personal work time-boxes?
• How did it feel fitting that time in your calendar?
• What did you spend that time on?
• Did you notice an effect on meetings from the preparation time-box? What?
• Did you notice other changes in your own working practice related?
• Would you or would you not recommend this practice to your peers? Why?

Appendix D. Scrum of product owners

The Scrum of Product Owners is a weekly meeting for all product owners.

Goal

• Regular knowledge sharing between POs on who is doing what
• To identify and address impediments in a timely fashion
**Schedule and agenda**

Time-boxed to 30 min

- Intro (2 min): Go over the practicalities of the meeting and make sure everyone is online/present
- Round (2 min per person):
  - Each participant in turn answers these questions:
    - What are your goals in your role currently?
    - Is there something that impedes your work?
  - Immediate follow-up discussion with other participants within the given time-box
  - If there is more discussion than the time-box allows, it can be continued immediately after the initial round
  - Or if the discussion concerns most of the participants it can be continued immediately at the discretion of the facilitator

- Follow-up discussions
  - The facilitator and participants can suggest topics to go over in more detail after the round is over
  - But they can also be scheduled for later if there is a pressing reason
  - If the topics do not concern you, you are free to go

**Guidelines for the facilitator**

- Make sure everyone knows when their up during the round
- Keep things moving smoothly
- Take notes on possible discussion points to be continued after the initial round

**Appendix E. Retro of Product Owners**

The Retro of Product Owners is a monthly meeting between POs aimed at continuous improvement of our working practices.

**Goal**

- Identify activities/tools/processes/things that do not support work, block or prevent fulfilling the role of PO
- Identify and share practices that support or make easier fulfilling the role of PO
- Form Action Points that lead to meaningful change in our ways of working

**Agenda and schedule**

Time-boxed to 60 min

- Intro (2 min): Go over the practicalities of the meeting and make sure everyone is online/present
- Topic formation and selection (5 min): Select themes for things to look into during this retro
- For each topic (15–30 min):
  - Discuss and answer the following questions:
    - What is the issue? What does it mean?
    - What is the impact of the issue on your own and your team’s work?
    - What could be underlying causes that contribute to the issue?
    - Do we already have some practices in place that can help with the issue?
    - What changes can we reasonably suggest to try and mitigate the issue?
    - Do we need more information to make those changes?
    - What?
      - Form Action Points based on the discussion:
        - Should have an idea on who, what, when and how

**Topic formation**

Ideas and suggestions for topics can be added in advance here.

**References**


