CEUR-WS.org/Vol-3506/paper08.pdf

Roadmapper: a Tool for Supporting Communication in Software Product Roadmapping

Henri Bomström[1], Markus Kelanti[1], Elina Annanperä[1], Kari Liukkunen[1], Outi Sievi-Korte[2], Kari Systä[2] and Veli-Pekka Eloranta[3]

[1]*University of Oulu, Oulu, Finland (e-mail: firstname.lastname@oulu.fi)*
[2]*Tampere University, Tampere, Finland (e-mail: firstname.lastname@tuni.fi)*
[3]*Vincit Oyj, Visiokatu 1, Tampere, Finland (e-mail: firstname.lastname@vincit.fi)*

Abstract

Software product roadmaps are practical tools that provide direction for product development. Software product roadmapping combines the reasoning *why* something is done with *what* should be done, often in the form of items to be delivered when constructing a software product. Successful roadmapping activities require collaboration from multiple stakeholder groups, such as business, development and management. However, aligning company goals, business strategy and development efforts is far from trivial. To this end, we conducted an action research study investigating how information exchange should be supported in software product roadmapping. As our results, we contribute the open-sourced *Roadmapper* tool and provide insights on how information exchange should be supported in software product roadmapping. Roadmapper supports information exchange in software product roadmapping by allowing different parties to clarify their views and making them understandable to other stakeholders, facilitating the discussion when they meet. Thus, Roadmapper visualises a common situational picture of software product development and acts as a group memory – helping to remember what the other stakeholders think about the matter.

Keywords

Software roadmapping, Software engineering, Information needs, Visualisation, Action research

1. Introduction

Roadmaps are practical tools for strategic planning as they facilitate communication between stakeholder groups, bringing together viewpoints from different parts of an organisation [1]. A roadmap refers to a plan that presents decision alternatives over time [2], to which the term roadmapping refers to the roadmap creation process [3]. Software product roadmapping combines the reasoning behind product development decisions, i.e. why something is done and what should be done, usually in the form of items to be delivered [4]. Clear strategy and goals, derived from company vision, play a central role in aligning software product roadmapping – having all involved stakeholders committed around the same plan [5]. However, practical business strategy and product development alignment are far from trivial [6, 7].

This paper presents the open-sourced software product roadmapping tool Roadmapper[1], developed to help product owners (PO) visualise the business impact of features – easing the planning of future software revisions[2] while aligning perspectives from both business and development. To do so, we conducted an action research (AR) [8, 9] study in collaboration with Vincit Oyj[3], studying and evaluating the tool's suitability for roadmapping activities from different stakeholder perspectives during the tool's development. The data for this study were collected in three iterative AR cycles, all of which included a simulated software product roadmapping session using the Roadmapper tool. Three expert reviewers from the hosting company represented the key stakeholders – one product owner, two account directors (ADs), and a developer – involved in the software roadmapping activities. We investigated how information exchange should be supported in software product roadmapping. Besides the tool itself, we contribute insights on how tool-assisted software product roadmapping may support practitioners from multiple viewpoints and how information should be represented.

The rest of the paper is organised as follows. In Section 2, we provide the theoretical background of our paper and explore related scientific research. Next, we describe our research process in Section 3. The findings of this paper are presented in Section 4 and discussed in Section

TKTP 2023: Annual Symposium for Computer Science 2023, June 13–14, 2023, Oulu, Finland
henri.bomstrom@oulu.fi (H. Bomström); markus.kelanti@oulu.fi (M. Kelanti); elina.annanpera@oulu.fi (E. Annanperä); kari.liukkunen@oulu.fi (K. Liukkunen); outi.sievi-korte@tuni.fi (O. Sievi-Korte); kari.systa@tuni.fi (K. Systä); veli-pekka.eloranta@vincit.fi (V. Eloranta)
0000-0002-7028-9044 (H. Bomström); 0000-0003-1886-8521 (M. Kelanti); 0000-0002-2615-9576 (E. Annanperä); 0000-0002-0719-4712 (K. Liukkunen); 0000-0002-4956-8989 (O. Sievi-Korte); 0000-0001-7371-0773 (K. Systä)
© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

[1]https://github.com/Vincit/VISDOM-Roadmapper
[2]https://iteavisdom.org/news/142
[3]https://www.vincit.com/

5. Section 5 also presents avenues for further research. Finally, the conclusions are given in Section 6.

2. Background and related work

2.1. Background

Roadmaps present decision alternatives over time [2], and roadmapping as an activity refers to the roadmap development process [3]. Software features are *"prominent or distinctive user-visible aspect, quality, or characteristic of a software system or systems"* [10]. However, various definitions for features are available, and a universally agreed-upon definition has yet to emerge [11].

Several grey literature reviews – based on the analysis of articles, blogs, white papers, and books – have been conducted around software product roadmapping in recent years (see, e.g. [4, 12, 13, 5]). Software product roadmapping has been found to combine the reasoning behind product development decisions, i.e. why something is done and what should be done, usually in the form of items to be delivered [4]. Clear strategy and goals, derived from company vision, play a central role in aligning software product roadmapping – having all involved stakeholders committed around the same plan [5]. Traditional roadmapping approaches face several issues: the lack of shared and well-communicated product vision, not reviewing and updating roadmaps, failure to integrate client feedback channels into roadmapping, using the wrong tools, unrealistic expectations, lack of criteria for roadmap item prioritisation, and creating a single roadmap for all stakeholders [12]. Overly detailed, feature-driven roadmaps can be prone to failure due to the inability to react to changes in uncertain business conditions [12].

Software product roadmaps can be divided into four categories: feature-driven, goal-oriented, outcome-driven, and theme-based roadmaps [13]. Software product roadmapping as a process can be divided into five phases: capturing, analysing, and prioritising features, roadmap validation and agreement, and change management – of which prioritisation and capturing features are the most critical ones [14]. Previous research has shown that at least three perspectives should be present in feature prioritisation activities: development, finances, and clients [15, 16]. Other studies continue on similar lines, promoting customer and partner representatives, marketing, product management and development as key stakeholders in software product roadmapping [14].

Roadmaps can be effective tools for strategic planning by facilitating communication between stakeholders and bringing together viewpoints from different parts of organisations [1]. Increasing the visibility of software planning activities within companies is essential in linking business and requirements engineering [17]. Understanding value creation from a customer perspective [17, 7] is vital for roadmapping activities, as fragmented customer knowledge is one of the main issues with software product roadmapping [7].

2.2. Related work

To remain competitive in turbulent marketplaces, contemporary software product companies must be able to deliver *value* to both themselves and their customers. Value-based software engineering [18, 19] (VBSE) is an approach that promotes thinking regarding value when making decisions in the context of engineering software. Feature value estimation is a multi-dimensional problem [20, 19] consisting of several dimensions: customer value, market competitiveness, economic value/profitability, cost efficiency, technology & architecture and company strategy [19]. However, best practices for value estimation are still missing.

One of the more interesting related research works is tied to the VALUE framework [21] and the VALUE tool [22, 23] – developed to assist companies in adopting value-based decision-making. The authors have observed improved stakeholder decision-making with the tool-assisted representation of stakeholder value propositions and facilitated meetings. Another example is the ReleasePlanner tool [24] which formally evaluates features by distilling their value from multiple criteria to produce optimal release plans. Other tools, such the Software Product Management Workbench [25], focus more on the product management aspects of roadmapping.

A study [26] was performed among more than 100 (Belgian) product builders, recognising six techniques to estimate and prioritise values of features:

- **Clustering in importance groups:** approaches that divide the requirements into a small number of different importance groups
- **Consensus-based approaches:** approaches specifically geared towards reaching consensus among a group of stakeholders.
- **Multi-criteria ranking:** approaches that automatically rank the requirements, based on the value of multiple relevant criteria and a specific formula that combines these values into a single value.
- **Pairwise comparison:** approaches that rely on mutually comparing all requirements, and identifying for each comparison the most valuable requirement.
- **Voting systems:** approaches that involve different stakeholders and ask each one of them to express their preference in some way or another.

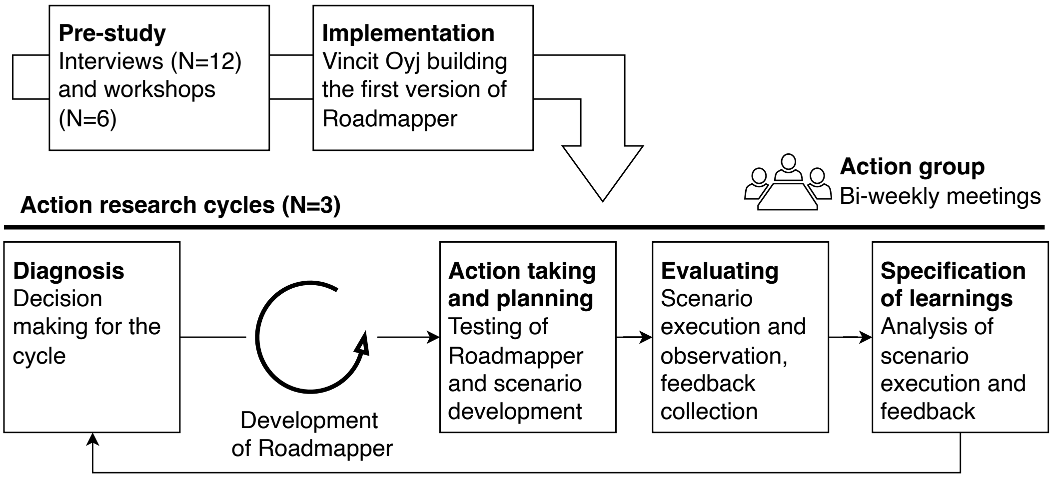

Figure 1: The research process consisted of a pre-study [30], implementation of Roadmapper's initial version, and three iterative action research cycles. Action research is an iterative research method with five phases: diagnosis, action planning, action taking, evaluating, and specification of learnings [28, 29].

- **Financial approaches:** approaches based on financial measures.

Furthermore, the authors [26] analysed these techniques in two dimensions: the number of requirements and uncertainty of value estimation – concluding that none of the techniques works well in cases where there are many requirements and the uncertainty of value estimation is high. Development effort and costs are also one of the more traditional ways of estimating the monetary impact of development and are notoriously challenging to predict. Prior research has found that effort estimations based on the subjective assessment of experts are the most prevalent effort assessment techniques in agile software development [27].

3. Methodology

This study aimed to support the software roadmapping process from the viewpoint of different stakeholder groups. This goal was refined into the following research question: *How should information exchange be supported in software product roadmapping?* We conducted an action research (AR) study [8, 9, 28, 29] in collaboration with Vincit Oyj[4], or Vincit for short from here on, to answer our research question. Three iterative AR cycles were conducted during this study (Figure 1).

[4]https://www.vincit.com/

3.1. Study context

This study was conducted as a part of the ITEA3 VISDOM[5] research project between September 2021 and May 2022. The research project's high-level goal was to develop new visualisations that combine data from multiple sources in modern DevOps development [31] and create visualisations related to the health of software processes. This study is based on the results from an earlier study [30] from that project, conducted between 2019 and 2020, to better understand stakeholders' information and visualisation needs in agile software development. The previous study included interviews and workshops to identify the information needs of software practitioners. One of the main findings of the previous study was that unsystematic management of client feedback and feature requests resulted in several information needs from multiple stakeholder perspectives. Thus, this study investigated further how input from multiple stakeholder perspectives can be combined collaboratively to improve decision-making related to software product roadmapping.

The design work of Roadmapper was initially based on the findings from the previous study, where a document was composed to outline the goal of Roadmapper and its initial features. The aim was to create a tool that visualises how software development efforts related to a higher-level plan – a goal, product vision or story, or other suitable aims that drive development in a specific direction. Collecting, estimating, and ordering development items was expected to increase visibility on the

[5]https://itea3.org/project/visdom.html

underlying business value expectations and workload estimations while deepening the understanding of current feature evaluation criteria. Tool support was presumed to provide visibility and traceability to software development decisions, benefiting stakeholder groups such as management, sales and development as roadmaps can be a base for communicating development plans and facilitating a shared understanding of development goals and priorities.

Vincit agreed to build the necessary software to begin studies with stakeholder information needs in software product roadmapping. Vincit is a multi-national small and medium-sized enterprise (SME) established in Finland that specialises in software development, service design, consulting and software products. At the time of the study, Vincit had around 470 employees globally.

The action research team consisted of four researchers and one assistant from two universities, and of Vincit's employees – one manager responsible for the company's research activities related to the tool, one user interface (UI) designer, a senior developer, and three junior developers that alternated participating in the action team meetings. Vincit's participants were selected by the manager responsible for their research activities based on the available resources and the team's skills, interests and availability. The action team held bi-weekly remote meetings to synchronise progress, which researchers recorded into a research diary shared between the action team in Google Drive[6]. The research diary included topics of discussion, possible research avenues, decisions made and recordings of dates and participants. The action team aimed to develop a tool to assist software product roadmapping by providing information from multiple stakeholder perspectives.

3.2. Summary of action research cycles

The Roadmapper tool was developed during three iterative AR cycles by Vincit while observed and evaluated by the research partners. Each cycle followed similar steps: 1) issues were diagnosed to be solved during each development cycle, 2) a simulated roadmapping session was planned and conducted with three expert reviewers, 3) researchers analysed the results to evaluate whether the diagnosed issues were solved, and 4) researchers specified the lessons learned, which were validated by the expert reviewers.

Several documents were composed in collaboration with company representatives before any sessions were conducted: facilitator guidelines, scenario descriptions for each session, and invitation letters for the participants. The facilitator guideline included instructions on conducting the sessions from beginning to end, reminding facilitators of what should be done. The action team discussed each scenario's general aim, after which Vincit produced a scenario that would represent as close as possible to a typical roadmapping session for the company – correspondence to real-world usage determined by Vincit, within reasonable limitations. The resulting scenario was then commented on by the researchers and revised as needed.

Three expert reviewers from Vincit represented the key stakeholders described in the scenarios: a PO, two ADs, and one developer. The number of expert participants was determined by the minimum number of roles needed to carry out the scenarios. Each of these roles had specific tasks: PO for leadership and product expertise, ADs to evaluate the business value of features and represent their client's needs, and a developer with technical knowledge of the product. For practical reasons, the selected developers were the ones that had developed the tool and could reliably answer technical questions the expert reviewers might have. The reasoning for these specific roles is further explained in Section 4.1. Vincit selected the expert reviewers based on their experience and availability for the sessions.

The scenario descriptions were sent to the three expert participants as attachments to the session invitations. The participants were also given customised descriptions of specific needs for their dedicated clients in the simulation and a more concrete mission to achieve during the session, such as *"In the upcoming roadmap planning meeting, your job is to make sure that your customers get the features they need on the roadmap."* The features to be discussed in the expert evaluation sessions were the Roadmapper tool's backlog tickets, imported by the software developers from Trello[7]. An example task description can be seen as *"Auto-accept user invitations on register"*.

Roadmapper was developed to include the necessary views and features to support software product roadmapping tasks before the evaluation sessions. The tool was further developed during the AR cycles based on the feedback from previous cycles. For each evaluation session, the Roadmapper was pre-filled with imported Trello tickets from the development backlog, called tasks within the Roadmapper, related to the scenario description to simulate ongoing development work in a product. Each task was pre-evaluated for the sessions based on their value for specific clients and overall implementation complexity. The participants were each given tasks to be added based on the scenario description. Then, ADs were responsible for evaluating task value for their clients, and developers were responsible for evaluating task implementation complexity. When these initial steps were complete, the participants were free to organise the soft-

[6]https://www.google.com/drive/

[7]https://trello.com/

ware product roadmap using the tool as they pleased to complete their mission in the scenario.

The expert evaluation sessions were conducted in a hybrid setting. The study participants, company observers, and researchers from Tampere University were located in a meeting room within the company premises. Researchers from the University of Oulu participated remotely due to the travelling restrictions related to the COVID-19 pandemic at the time. The sessions were recorded via Microsoft Teams[8] with informed consent, and privacy procedures were explained to the participants. The facilitator described the session for the participants, led the session's activities, asked clarifying questions, noted the time used, and kept the session moving if the participants needed help with what to do. Other participants observed the sessions and took notes. The PO was asked to share their screen in each session and lead discussions with ADs and software developers to create a balanced software roadmap.

After each session, one researcher watched the related recording and created an annotated file by marking the time when something interesting happened or was discussed and wrote a description of what was discussed without any further analysis. An example of an annotation can be seen in *"57:22 - Participant C noted that development always featured surprises and that the total amount of time would not be completely known. They noted that a value indicating development speed (in the same unit that work estimates use) would be useful."* After completing the annotated file, the notes were grouped by their overarching theme, such as the numerical scale of feature evaluation. Then, the notes under each theme were re-read and condensed into a description that conveyed the essence of each theme.

The themes, their meaning and proposed actions to address deficiencies were reported each iteration in an analysis document shared within the action team. In addition to research-related topics, any encountered bugs in the tool were reported in the analysis document. The results were then further condensed as Powerpoint[9] slides for presentation to the expert participants for validation. Based on the results, the researchers made recommendations for actions for the next cycle, which were then discussed with the company participants when planning the next cycle. The action team then selected the issues to be addressed in the next cycle.

4. Results

We report our results and lessons learned from the viewpoint of how and why the Roadmapper tool was constructed as it was based on the three AR cycles. Our results cover three main areas of interest: specifying the necessary information content and representation for supporting information exchange, how value and complexity estimation was perceived by the participants, and answering the question of *what to do next based on this information* via collaborative roadmap planning.

4.1. Information contents of Roadmapper

Roadmapper was designed to facilitate information exchange in software product roadmapping with three main types of stakeholders in mind: *product owners* that provide the direction and vision for software product development, *account directors* that know their clients and their needs within the product's context, and *developers* who provide technical expertise on implementing the requested changes to said product. Roadmapper was intended to aid in discussion sessions led by the PO, committing the relevant stakeholders to the same action plan concerning the product. Thus, the essential point is that one should not think of our results only from a technical roadmapping perspective but as an attempt to support information exchange in general – where supporting the needs manifests as a roadmapping tool.

Roadmapper is divided into five main views. The first default view is the dashboard that visualises the current roadmap's value output of the current plan versus a numerically optimal plan. The dashboard also includes all tasks that still need to be rated. The second main view is the tasks view, providing an overview and details of all the tasks within the project. The third view, clients, represents a list of clients interested in the product being developed. The weights of each client can be altered to affect the optimal roadmap. The fourth view, the team view, provides a way to manage the persons included within the selected project, selecting their roles and granting them access to specific clients. The final view, the plan tab, is the most important as it features the functionality needed to create milestones and visualise their current composition (Figure 2).

Conceptually, Roadmapper deals with development items in a ticket format imported from external project management tools such as Trello, Jira[10] and Gitlab[11]. The tickets within those systems, with abstraction levels corresponding to each company's practices, are called tasks within the Roadmapper. Each task includes ratings for *value* and *complexity*. Account directors are responsible for rating task value for each of their clients: how valuable that item would be from the client's perspective. On the other hand, developers evaluated complexity, estimating the total challenge and effort needed to implement said task to the product.

[8]https://www.microsoft.com/en-us/microsoft-teams/log-in
[9]https://www.microsoft.com/en-us/microsoft-365/powerpoint
[10]https://www.atlassian.com/software/jira
[11]https://about.gitlab.com/

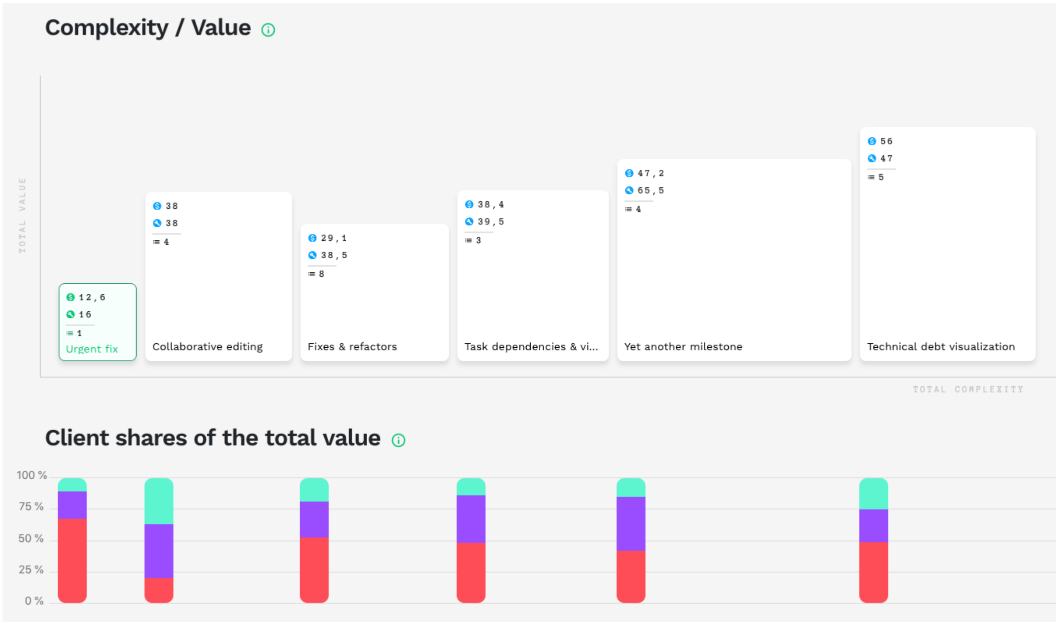

Figure 2: An example visualisation of value output of the current development plan. The X-axis represents task complexity, and the Y-axis represents task value. Each rectangle represents a milestone, with its name, total complexity and value, and the number of tasks. Below, the share of the value created for each customer per milestone is presented with coloured bars.

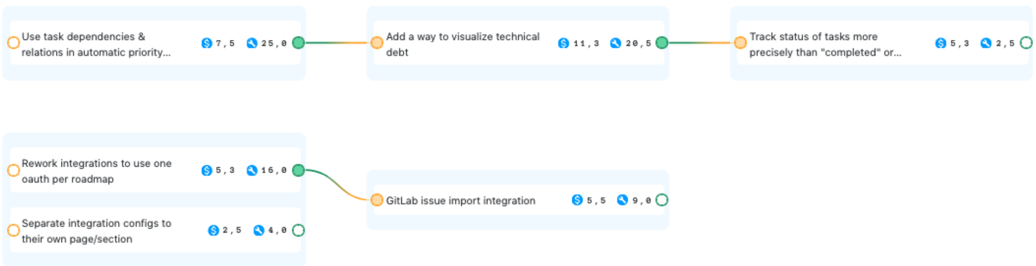

Figure 3: An example of a task map within Roadmapper. Tasks within blue containers represent synergistic tasks, which should be constructed from a technical viewpoint. The connecting arrows represent dependencies between tasks, communicating which tasks depend on other tasks before they can be completed.

Besides numeric metrics, each task was assigned dependencies and synergies concerning other tasks (Figure 3). Dependencies dictate whether a task must be completed before other tasks from a technical point of view or the other way around. The synergy between tasks represents an alternative, optional way of evaluating task value – developing synergistic tasks simultaneously within a specific milestone would increase the roadmap's overall value creation concerning the effort spent (complexity within Roadmapper).

Projects represent the most high-level concept within Roadmapper, each project including one roadmap. Individual roadmaps consist of milestones (Figure 4) – which are collections of tasks. The milestones are read from left to right, the leftmost representing the next development cycle. Each milestone provides average and total scores for values along a client share score, denoting how well the proposed milestones follow the current client weights set by the PO. Each client score can be set to denote their importance to the company using Roadmapper, assuming that a software product is being developed to be used by multiple client organisations. Milestone planning in Roadmapper provides discussion support for the collaborative identification of valuable tasks, ordering them to deliver value.

The expert participants all agreed that the tool, to be

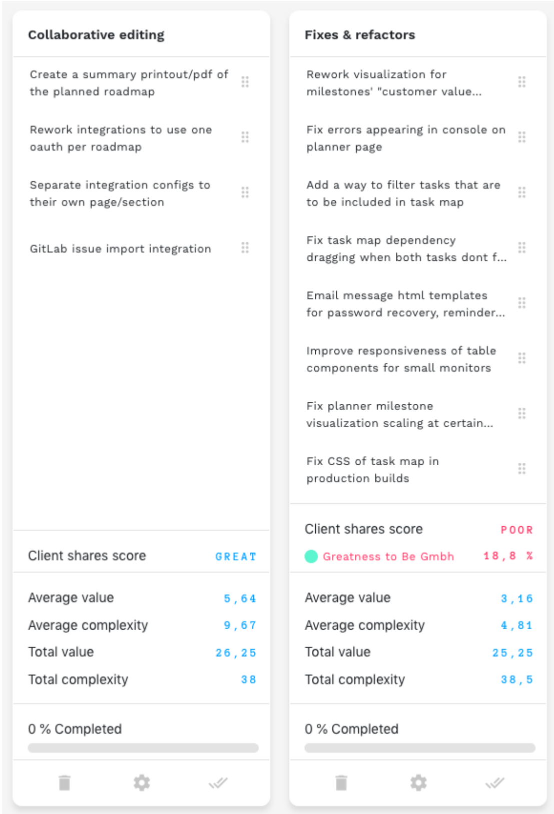

Figure 4: A typical example of milestones within a roadmap initially created by researchers and modified by study participants. The roadmap consisted of several milestones, depicted as columns in the figure, with several metrics associated with each milestone.

helpful, should focus on simplicity to allow for flexibility over rigid practices. This guiding principle was followed during the tool's construction and provided critical reasoning on why it ended up as it did. Concrete examples of this principle can be found in how value and complexity are estimated and how the tool always reflects the current situation and is updated as needed. The core idea of the tool was not to represent everything perfectly but to present the necessary information in an understandable format to support collaboration and information exchange.

4.2. Value and complexity estimation

Numerical evaluation of tasks is an old issue in software engineering – and an essential aspect of the tool since it relies on providing numerical task values. The three AR cycles showed that two main design aspects received the most feedback from the participants: the numerical evaluation scale of value and complexity, and interpreting criticality of tasks – evaluating tasks between the opposite ends of the spectrum in importance: irrelevant or business critical for specific clients.

We found that a five-step scale proved the most effective in rating tasks when accompanied by textual descriptions of the values to guide the scale's usage between stakeholders. The following scale was used in the third iteration: 1 for Not relevant, 4 for A bit important, 9 for Somewhat important, 16 for Important, and 25 for Business critical. Three evaluation scales were tested in total – a linear scale from one to ten, a linear scale from one to five, and the third non-linear five-step scale from one-to-25 – and ended up with the last scale as it best corresponded to participant needs and produced noticeable value differences between tasks.

The business criticality of tasks was often referred to as the most important aspect from a customer viewpoint. Whether or not delivering the task would have ramifications with doing business with a specific client. Several discussions were had about evaluating business criticality separately or indicating it with flags, but the consensus remained to keep the tool simple to allow for flexibility. The caveat of not including separate indicators for business criticality is that one task may be critical for one client and irrelevant for others, implying that the criticality of tasks may only sometimes be related to the overall business value of tasks – another example of why supporting discussions in product roadmapping are essential for companies. Another issue uncovered was rating tasks when ADs were unsure of the task's value – situations such as when a task might have value to the company, but further discussions were required to determine the actual value. Resolving this issue was left as future work: determining whether the task is good enough to decide on and how to indicate it. Lastly, free-form comments within tasks were considered helpful in conveying information such as layouts for planned features and implementation deadlines – dates that determined when the feature had to be complete and delivered to receive any value, such as before an event.

For future work, a *technical debt* score was requested to dictate how the roadmap should be balanced towards the company's own needs, which could help communicate long-term commitments and maintainability towards clients. Developers would evaluate technical debt from a technical point of view and display it as a core metric along with task value and complexity.

4.3. Supporting communication in milestone planning

Milestone planning lies in the heart of Roadmapper, where the participants answered the question of *what* *to do next based on this information* via collaborative roadmap planning. This view combines the knowledge of each perspective into action, where the inputs of everyone involved are resolved to a common cause. Following the principle of simplicity and flexibility, we found that each task must contain the following information for quick milestone planning: how complex each task was to complete, how much value the task would create when completed, and which tasks were essential to clients – confirming the initial presumptions on the tool's design.

The milestones in Roadmapper were purposefully left more abstract than they could have been, once again echoing the principle of simplicity and flexibility. Even though the participants intuitively thought of milestones as scrum sprints, the abstraction level and purpose of creating milestones varied. The participants argued that some tasks, such as hot-fixes, were out of scope for this kind of tool and placed these kinds of tasks directly to strategic development – meaning in practice that they were added to the current or next development sprint as priority tasks and purposefully not included in the roadmap. An essential mental distinction is that the roadmap was not considered a concrete sprint plan either, supporting the findings of the conducted pre-study. Task complexity is abstracted away from actual working hours, even though the tool provides a sanity-check mechanism for comparing the complexity of milestones to estimations of real working time. Nevertheless, this feature is intended to judge the appropriate sizes of milestones against each other.

The participants created software roadmaps with what was titled "the greedy algorithm", first doing tasks that provided the most value with the least complexity. The PO determined the milestones within a roadmap using the following process. First, team velocity was needed to determine the correct size of a milestone. Then, a balanced composition of tasks was collected within said milestone based on client weights or other explicitly stated needs – repeated until as many milestones as needed were created. The "greedy algorithm" did, however, face issues that were expected by the action team, further verifying the initial presumptions: the dependencies of tasks must be known before they can be meaningfully arranged.

> *"As PO, I think that if I have an understanding from the technical experts that there are dependencies here or that these should be done at the same time. If I have an understanding from the account directors that how much this particular ticket serves this customer, with what kind of weightings and what is the order of priority of the tickets from that account's point of view, then I would believe that as a PO I would be able to make a pretty far-reaching ready-made guess that could be reviewed with the accounts."* (PO)

For future reference, the participants argued that initial software roadmaps could also be created automatically if task dependencies, the value and complexity of tasks, client weights and team capacity were known. Furthermore, task synergy estimation was considered helpful and highlighted the collaborative nature of planning when deciding which items should be completed in which development cycle.

The openness of information was considered essential by the participants. Specific views, such as the milestone planning view, were initially restricted to being accessible only to the PO. However, all the participants agreed that open information sharing within the tool was preferred over restricting access to information. The participants highlighted the tool's conversational nature, where some roadmapping insights only surface in joint sessions when looking at the same screen – showing the screen to other roles leads to making specific observations earlier. In practice, the participants agreed that it would benefit ADs and developers to access all parts of the tool, with read-only access at the very least.

The role of ADs was to support the PO while constructing roadmaps and provide client-related knowledge for the PO by rating task values from the viewpoint of their clients. A concrete visualisation was suggested as a list of tasks sorted by highest value and lowest effort concerning the specific client to help communicate their most pressing needs to the PO. Furthermore, understanding the status of each customer using the product and providing communication assistance towards the clients was considered a priority for further development action by the ADs. The ADs were interested in knowing how each client was doing regarding overall value output and whether their needs were being served.

> *"What's coming [features to be delivered] and a rough estimate of when it's coming. After all, it could show the tasks that generate a lot of value for the customer in question on a timeline and what comes from there at any time."* (AD1)

Finally, visually and conceptually explicating both technical and value-related information was considered to be the most valuable asset of Roadmapper, to which the PO and ADs provided the following shared testimony in agreement:

> *"[the Roadmapper tool] would help. I have done roadmapping with different tools. When [using other tools, you're] carrying similar types of information with other*

ways, or by asking, or in conversations, but here we get technical expertise information in a formal way, it is shown and it can be utilised when planning. And in the same way numerical information can be obtained from accounts [account directors], of course it helps. Information on the basis of which one does one's own work seems pretty good based on this." (PO)

5. Discussion

This study aimed to facilitate information exchange between stakeholder groups in software product roadmapping. To this end, we contribute an open-sourced software product roadmapping tool called Roadmapper and specify lessons learned during its construction with Vincit.

5.1. RQ: How should information exchange be supported in software product roadmapping?

Software product roadmapping allows stakeholders to commit to a shared product development plan [5] by combining perspectives from different parts of an organisation [1]. However, aligning business strategy and product development remains challenging in practice [6, 7]. To this end, we created a solution primarily focused on supporting the discussion between stakeholder groups to align their perspectives for a shared product development plan. Besides Roadmapper itself, the main results of this study are the lessons learned on how tool support can help foster collaboration in software product roadmapping. We highlighted three main areas of interest in our study: the information content and representation for supporting information exchange, how value and complexity estimation was perceived and done by the participants, and how milestone planning tied these aspects together into actions by answering the question of *what to do next based on this information* via collaborative roadmap planning.

We successfully supported information exchange in software product roadmapping with role-specific information representation. According to previous research, development, finances, and clients' perspectives should be present in feature prioritisation activities [15, 16]. Roadmapper follows a similar cast of stakeholders by including a technical perspective from development, a customer value perspective conveyed by ADs, and a product leadership perspective from a PO. Both developers and ADs were responsible for rating tasks according to their speciality: developers provided an abstraction of task complexity, implementation synergies and technical dependencies of tasks. ADs provided abstracted value estimations on how beneficial the features would be from their customers' perspectives. Roadmapper supports information exchange by providing the means to collect and present information the most relevant for each stakeholder group, allowing them to discuss it and form a Roadmap that provides a steady output of value while considering necessities such as technical dependencies.

Roadmapper helps address the most critical and problematic phases of product roadmapping: prioritising features [14]. Roadmapper does this by fostering communication between stakeholder groups and providing visual aids to obtain a joint agreement on the product roadmap. Furthermore, Roadmapper helps address several other issues reported in contemporary roadmapping, primarily:

- **The need for shared and well-communicated product vision.** [5, 12] Roadmapper-facilitated discussions are aimed at providing a joint agreement on the roadmap, where the PO is responsible for following and communicating the company vision.
- **The need for integrating client feedback channels into roadmapping.** [12] When ADs know their clients' needs, their primary responsibility in Roadmapper-facilitated discussions is to ensure their clients receive value and remain satisfied with the current product roadmap.
- **The need for criteria in roadmap item prioritisation.** [12] Roadmapper provides explicit means and metrics to prioritise features: customer value, task complexity, technical dependencies, and synergy between features. Each participant is able to see the criteria used for decision-making, increasing the transparency of product development decisions.

Guided by the principle of simplicity, Roadmapper provides an open-ended and flexible way of evaluating feature value in a software product roadmapping context. In comparison to other tools, unnecessary complexity was found to be one of the VALUE tool's main weaknesses [22]. Tools such as ReleasePlanner [24] and Software Product Management Workbench [25] also rely on a more formal development planning. However, in this study the participants all preferred the free-form way of roadmapping that focused on ordering feature delivery based on value output and implementation complexity, while indicating implementation synergies and dependencies.

Based on our findings and the lessons learned while constructing Roadmapper, we conclude that abstractions of feature value for clients and implementation complexity, coupled with task dependencies and synergies, form a base on grouping and ordering tasks for a software product roadmap. Increasing the visibility of presumptions and expectations of value and complexity promotes decision-making transparency for all participants and aids in achieving a joint consensus between stakeholder groups when using the tool to support product development discussions. Finally, the following lines of research provide opportunities for further research.

The first avenue for future research is value estimation in software engineering. Estimating value for features and other development items, such as paying back technical debt, is often challenging. Feature synergy was one of the core ways of estimating the value of features in this study. Future research should investigate how the value of different development items can be estimated and communicated transparently.

Pre-emptive estimation of technical debt represents the second direction for future research. Implementing software features often incurs technical debt. Future research should investigate how decision alternatives in product roadmapping affect the upcoming technical debt of the product.

Lastly, supporting customer-directed communication offers a third avenue for future research. Roadmapping-related information is often easier to communicate to internal stakeholders than external stakeholders. Future research should investigate how customer-directed communication can be supported when making product development decisions.

5.2. Validity and limitations

There are several limitations associated with this study. The participants of this study and their number were decided by the hosting company, representing a selection of personnel suitable for expert reviews. However, the PO noted that they lacked experience in this specific kind of work where a product was sold to multiple clients, despite having strong experience otherwise. One potential flaw in the simulation is that the developers were part of the action team and those that built the tool in reality. The role of developers in these kinds of discussions should be studied further in future research.

The Roadmapper's backlog items were used instead of actual cases for confidentiality and practical reasons, which led to simulated scenarios that may have affected how the participants approached using the tool. The first iterations suffered from technical immaturity, potentially leading to different results given more development time each cycle. However, the discussions saturate towards the third session, giving the impression that the tool fulfilled its purpose with its current features.

We address validity, concerning the qualitative aspects of this study, through a detailed description of data [32] by providing supporting quotations for the presented findings and reasoning how and why the tool was developed as it was. The findings of each cycle were reported back to the participants for verification, and they were given a chance to comment on them – grounding them through expert review and validation. However, we consider it part of future work to see whether the conclusions presented in this paper can be applied to projects using different development methods from a general software engineering viewpoint.

Lastly, two researchers participated actively in the research cycles, and three other researchers participated in the validation of the results as outsider participants to reduce the risk of bias due to being an active part of the research. The company participants conducted the actual development.

6. Conclusions

Roadmaps are valuable tools that represent decision alternatives over time, combining the *why* and *what* should be done in software product development. However, aligning business strategy and product development in practice remains challenging. To address this challenge, we conducted an AR study on how information exchange should be supported in software product roadmapping. To this end, we contribute an open-sourced software product roadmapping tool titled Roadmapper. In addition, we specify lessons learned in supporting the exchange of information. We emphasise that the tool shows promise in supporting information exchange in software product roadmapping by explicating tacit knowledge, promoting the visibility of estimations, transparency of decision-making, and promoting the voice of each stakeholder group to be heard within those discussions – despite the small sample of participants and the simulated nature of the evaluation sessions.

The main takeaway from this study is that role-specific information representation was found to aid software product roadmapping by facilitating information exchange between stakeholder groups. Roadmapper supports information exchange by allowing different parties to clarify their views and making them understandable to other stakeholders, facilitating the discussion when they meet. Thus, Roadmapper visualises a common situational picture of software product development and acts as a group memory – helping to remember what the other stakeholders think about the matter. The findings from this study related to providing the necessary information contents for software product roadmapping, discussing value and complexity estimation of tasks, and outlining how communication in milestone planning can be supported. Abstractions of feature value and implementation complexity, coupled with task dependencies and synergies, were found to form a base for software product roadmaps.

Acknowledgments

The authors thank Vincit for implementing the Roadmapper tool, and all participants for the time and expertise they provided for the expert review sessions. The work was supported by the ITEA3 project VISDOM and the authors thank Business Finland for the funding.

References

[1] R. Phaal, C. J. Farrukh, D. R. Probert, Technology roadmapping—a planning framework for evolution and revolution, Technological forecasting and social change 71 (2004) 5–26.

[2] G. DeGregorio, Technology management via a set of dynamically linked roadmaps, in: Proceedings of the 2000 IEEE Engineering Management Society. EMS-2000 (Cat. No. 00CH37139), IEEE, 2000, pp. 184–190.

[3] R. N. Kostoff, R. R. Schaller, Science and technology roadmaps, IEEE Transactions on engineering management 48 (2001) 132–143.

[4] S. Trieflinger, J. Münch, J. Schneider, E. Bogazköy, P. Eißler, B. Roling, D. Lang, Product roadmapping processes for an uncertain market environment: A grey literature review, in: Lean and Agile Software Development, Springer International Publishing, Cham, 2021, pp. 111–129.

[5] S. Trieflinger, J. Münch, E. Bogazköy, P. Eißler, J. Schneider, B. Roling, Product roadmap alignment – achieving the vision together: A grey literature review, in: Agile Processes in Software Engineering and Extreme Programming – Workshops, Springer International Publishing, Cham, 2020, pp. 50–57.

[6] M. Komssi, M. Kauppinen, H. Töhönen, L. Lehtola, A. M. Davis, Integrating analysis of customers' processes into roadmapping: The value-creation perspective, in: 2011 IEEE 19th International Requirements Engineering Conference, 2011, pp. 57–66. doi:10.1109/RE.2011.6051662.

[7] M. Komssi, M. Kauppinen, H. Töhönen, L. Lehtola, A. M. Davis, Roadmapping problems in practice: value creation from the perspective of the customers, Requirements Engineering 20 (2015) 45 – 69. doi:10.1007/s00766-013-0186-3.

[8] R. N. Rapoport, Three dilemmas in action research: with special reference to the tavistock experience, Human relations 23 (1970) 499–513.

[9] G. I. Susman, R. D. Evered, An assessment of the scientific merits of action research, Administrative science quarterly (1978) 582–603.

[10] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, A. S. Peterson, Feature-oriented domain analysis (FODA) feasibility study, Technical Report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.

[11] S. Molenaar, E. Steenvoorden, N. van den Berg, F. Dalpiaz, S. Brinkkemper, Defining key concepts in information science research: The adoption of the definition of feature, in: R. Guizzardi, J. Ralyté, X. Franch (Eds.), Research Challenges in Information Science, Springer International Publishing, Cham, 2022, pp. 442–457.

[12] S. Trieflinger, J. Münch, D. Petrik, D. Lang, Why traditional product roadmaps fail in dynamic markets: Global insights, in: Product-Focused Software Process Improvement, Springer International Publishing, Cham, 2022, pp. 382–389.

[13] J. Münch, S. Trieflinger, E. Bogazköy, P. Eißler, B. Roling, J. Schneider, Product roadmap formats for an uncertain future: A grey literature review, in: 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020, pp. 284–291. doi:10.1109/SEAA51224.2020.00055.

[14] T. Suomalainen, O. Salo, P. Abrahamsson, J. Similä, Software product roadmapping in a volatile business environment, Journal of Systems and Software 84 (2011) 958–975.

[15] A. Davis, The art of requirements triage, Computer 36 (2003) 42–49. doi:10.1109/MC.2003.1185216.

[16] P. Berander, A. Andrews, Requirements prioritization, in: Engineering and managing software requirements, Springer, 2005, pp. 69–94.

[17] L. Lehtola, M. Kauppinen, J. Vähäniitty, M. Komssi, Linking business and requirements engineering: is solution planning a missing activity in software product companies?, Requirements engineering 14 (2009) 113–128.

[18] B. Boehm, Value-based software engineering, SIGSOFT Softw. Eng. Notes 28 (2003) 4. doi:10.1145/638750.638776.

[19] P. Rodríguez, C. Urquhart, E. Mendes, A theory of value for value-based feature selection in software engineering, IEEE Transactions on Software Engineering 48 (2022) 466–484. doi:10.1109/TSE.2020.2989666.

[20] P. Rodríguez, E. Mendes, B. Turhan, Key stakeholders' value propositions for feature selection in software-intensive products: An industrial case study, IEEE Transactions on Software Engineering 46 (2020) 1340–1363. doi:10.1109/TSE.2018.2878031.

[21] E. Mendes, P. Rodriguez, V. Freitas, S. Baker, M. A. Atoui, Towards improving decision making and estimating the value of decisions in value-based software engineering: the value framework, Software Quality Journal 26 (2018) 607–656.

[22] V. Freitas, E. Mendes, B. Turhan, Providing tool-

support for value-based decision-making: A usability assessment, in: 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2016, pp. 34–41. doi:10.1109/SEAA.2016.44.

[23] V. Freitas, M. Perkusich, E. Mendes, P. Rodríguez, M. Oivo, Value-based decision-making using a web-based tool: A multiple case study, in: 2017 24th Asia-Pacific Software Engineering Conference (APSEC), 2017, pp. 279–288. doi:10.1109/APSEC.2017.34.

[24] G. Ruhe, M. Saliu, The art and science of software release planning, IEEE Software 22 (2005) 47–53. doi:10.1109/MS.2005.164.

[25] I. V. De Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, L. Bijlsma, On the creation of a reference framework for software product management: Validation and tool support, in: 2006 International Workshop on Software Product Management (IWSPM'06 - RE'06 Workshop), 2006, pp. 3–12. doi:10.1109/IWSPM.2006.6.

[26] T. Tourwé, W. Codenie, N. Boucart, V. Blagojević, Demystifying release definition: From requirements prioritization to collaborative value quantification, in: Proceedings of the 15th International Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ '09, Springer-Verlag, Berlin, Heidelberg, 2009, p. 37–44. doi:10.1007/978-3-642-02050-6_4.

[27] M. Usman, E. Mendes, J. Börstler, Effort estimation in agile software development: A survey on the state of the practice, in: Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, EASE '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 1–10. doi:10.1145/2745802.2745813.

[28] J. McKernan, Curriculum action research: A handbook of methods and resources for the reflective practitioner, Psychology Press, 1996.

[29] R. L. Baskerville, A. T. Wood-Harper, A critical perspective on action research as a method for information systems research, Journal of information Technology 11 (1996) 235–246.

[30] H. Bomström, M. Kelanti, E. Annanperä, K. Liukkunen, T. Kilamo, O. Sievi-Korte, K. Systä, Information needs and presentation in agile software development, Information and Software Technology (2023). doi:10.1016/j.infsof.2023.107265.

[31] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, Devops, IEEE Software 33 (2016) 94–100. doi:10.1109/MS.2016.68.

[32] J. W. Creswell, D. L. Miller, Determining validity in qualitative inquiry, Theory Into Practice 39 (2000) 124–130. doi:10.1207/s15430421tip3903_2.