Long-term Planning of Development Efforts by Roadmapping – a Model and Experiences from Small Software Companies

Jarno Vähäniitty, Casper Lassenius, Kristian Rautiainen and Pasi Pekkanen
Software Business and Engineering Institute
Helsinki University of Technology
Espoo, Finland
firstname.lastname@tkk.fi

***Abstract*—*Success in the software product business requires timely release of new products and upgrades with proper quality and the right features. For this, a systematic approach for managing the contents, timing and roles of future product releases as well as the product architecture is needed. In practice, such an approach is often missing, especially in small companies, due to inexperience, unclear priorities, time-to-market pressures, or the lack of suitable process infra-structure. In this paper, we present a model to visualize product roadmaps developed together with three small software companies and experiences from its use. The model depicts release and development schedules, the composition of individual releases, services that require attention from the developers, changes to the underlying technology and the planned resource usage. We also present lessons learned from the case companies and outline directions for future research.*

Keywords-roadmapping; software development; small company

I. INTRODUCTION

In addition to the capability to invent new solutions and realize them as software, success in the software product business requires delivering the right kind of products to the market at the right time. Software product development often involves evolving both the individual products and the technologies they are based on at the same time [5], and planning the product architecture together with future releases is crucial for success. However, there is still little guidance available for organizing the interface between software development process and business processes. For example, connecting feature and release cycle planning to business planning is an area that remains relatively unaddressed [8, 10].

In small software companies (those with less than 50 employees), a systematic approach for long term product and business planning is often missing because of inexperience, time-to-market pressures [1], or the lack of process infrastructure such as requirements management [8]. Small companies also face the challenge of coherently expressing and communicating their long-term plans to various stakeholders such as venture capitalists and potential customers [13]. To address these challenges, we present in this paper a model for visualising long-term software development plans developed together with three small software companies. First, we discuss roadmapping in the context of software product development. Second, we present the developed model. Third, we present our experiences from applying and discussing the approach in practice. Finally, we round up with discussion and directions for further work.

II. ROADMAPPING AND SOFTWARE PRODUCT DEVELOPMENT

Roadmapping is a popular metaphor for planning and portraying the use of scientific and technological resources, elements and their structural relationships over a period of time. The process of roadmapping identifies, evaluates and selects strategic alternatives that can be used to achieve desired objectives [9], and the resulting roadmaps summarise and communicate the results of key business decisions [6]. Product roadmapping is a ”disciplined, focused, multiyear approach to product planning”, with the roadmap’s implementability viewed as important as its strategic value [9].

Software products typically evolve in releases, with each release including new and improved functionality intended for keeping the vendor ahead of the competitors. Wiegers uses the notion of business requirements to represent the needs customers have for the product [19]. Bosch defines software requirements as consisting of functional requirements and quality attributes, with the term *feature* referring to a group of related software requirements [1]. Combining from these, roadmapping consists of defining user needs, prioritising them and then responding with features. As a release gets closer, its content is elaborated from the level of business requirements and features to functional requirements and quality attributes. When product roadmapping is used in time-to-market -driven development, moving features and their parts between releases should be based on the relative importance of the business requirements in question.

In the software product business, the software itself is not the only component – it is often combined with services. The whole product concept [11] implies that the delivery of the core benefit the customer is buying can be enhanced by modifying the way it is packaged or complementing it with services. For the product vendor, however, incorporating and managing services can be challenging [12]. Restricting product and release planning to product features only limits the view on what has to be achieved in order to put together a compelling offer [18]. For small companies, failure to recognise the resource implications of providing the needed servicing is likely to lead to a crisis [7, 17]. Thus, product roadmapping should cover the whole product, not only the software component of it.

III. A MODEL FOR PRODUCT ROADMAPPING

Our roadmap visualization - shown in Fig. 1 below - aims to define and concretise the company's plans for technology and product development and servicing to a degree of accuracy through the use of semi-formal notation. It has been adapted and further developed from the models described by Wells et al. [18]. It expresses the release and development schedules for the product(s), the composition of individual releases, changes to the underlying technology, services requiring attention from product development and planned resource usage. The roadmap consists of five layers, with the four topmost depicting the development of various parts of the whole product as *activities*, and the bottom layer showing the estimate of human resources required at a given moment.

Activities and their planned schedules and effort estimates are presented as horizontal bars in the product roadmap. Possible activity types are performing services, preparing releases and developing product components and platforms. A *product platform* is a core software asset on top of which the product is built and expanded on, and may be generic enough to be used in other products as well [4].

Product components are business requirements translated as software, meaning relatively independent (groups of) features. The related business requirements, as well as more detailed information on the functionality should be kept in the requirements management system. Documentation, whether internal or intended for the end-user, is depicted as product components when necessary. This is a logical choice, since associated documentation is usually defined 'software' as well [14].

Preparing a *product release* consists of integrating the related product components and platform(s), doing system testing and error correction, as well as performing other release-related activities. In the notation there are three kinds of possible releases: major releases, minor releases and patches. The first diamond in a new release denotes a major release, and subsequent diamonds and circles mark minor and patch releases, respectively. Only releases on the release layer are visible to customers.

The *services* a company offers are classified and dealt with based on the kind of attention they need from product development. The classification consists of *product accessories*, *customer-specific development services* and *other services*. Product accessories are a one-time effort requiring product development resources to fulfil a need common to many customers. Typically, they are initially developed for a specific customer, but are to be included as part of the standard offering. Product accessories are expressed in the product roadmap as regular product components integrated into a future release of the product. Customer-specific development services require resources from product development, but their outcome is limited to the customer receiving the service. They are depicted on the service layer. Other services refer to services that at the moment do not appear to require attention from product development. We currently think that it is not necessary to include these kinds of services into the product roadmap.

The thicker an activity is in the visualisation, the more resources are allocated for the period indicated. In our case companies, different people or teams worked on the platform and product components. To help in balancing resources, textual notation is used inside the activities to denote the allocated *resource type(s)*. Resource information is also summed to the bottom layer of the visualisation.

Arrows going from one activity to another denote composition and timing for integrating the activities' outputs and may, depending on the context, imply reuse. Thus, the product roadmap visualisation contains information of the product architecture over time, specifically, the relationships between product releases, components and platforms, and how and when they are composed of each other. As no exhaustive rules exist for discriminating between 'plain' technologies, product platforms and product components for roadmapping purposes, a reasonable conceptual structure must be resolved case-by-case.

The example roadmap of Figure 1 shows the plans from ToolCo (see section 4) regarding a toolkit for rapid creation of web-based user interfaces. The only service currently identified to require product development attention is the one-day basic training per license sold, provided starting from the major release 5.0 in 8/2009. Preparing materials for the training is depicted as a product component. The other product components are add-in modules for various terminal devices, and end-user documentation to be shipped with the product starting from a minor release in 8/2008. On the platform layer, the roadmap shows two generations of the toolkit 'engine', with the second generation to be used starting from 5.0.

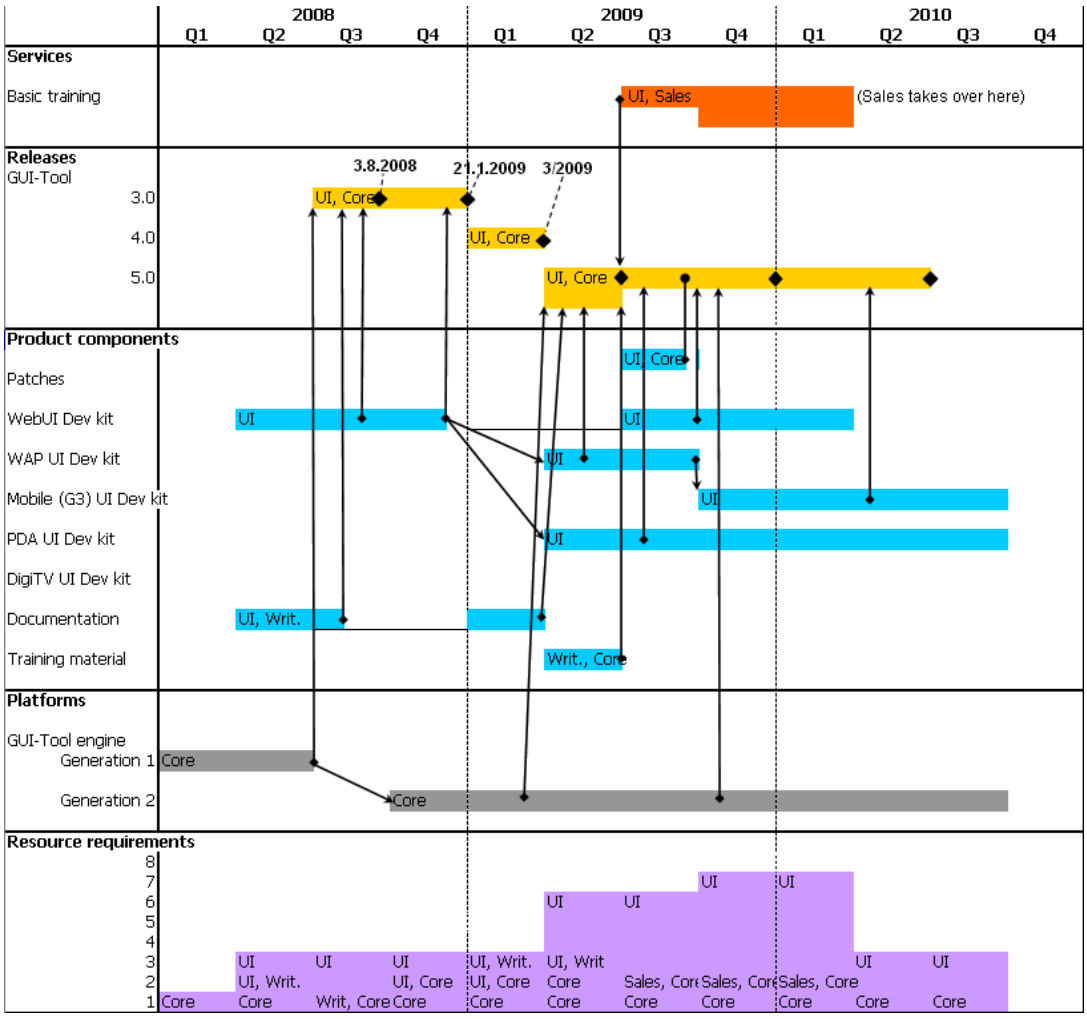

Figure 1. The model for visualising product roadmaps

Together with one of the case companies (see section IV) we summarised a four-step process for creating and updating product roadmaps. The process can also be thought as a checklist for what practitioners should take into consideration when conducting long-term release planning. The steps in the process should be performed periodically to adjust the roadmap to new information and changing market situations, and smaller updates should be done to ensure the roadmaps always hold current information. Tabrizi and Walleigh present an example in [15] in which senior management of a technology-intensive company updates the company's product roadmaps bimonthly, and redraws them completely every six months. Our process is summarised in Table 1 and explained below.

The first step is to define (or revise) and analyse the strategic mission and vision of the company. All companies, no matter how small, should have an idea of their purpose and desired future clear enough to be written down before they plan their operations in more detail. Often some kind of product vision exists even if the company's mission and vision are not explicitly defined. Mission and vision should act as the guideline for shaping the product vision and choosing between strategic alternatives.

The second step is to identify major trends in the general environment. This encompasses looking at potential customers, competitors, the industry and developments in relevant technology. Many well-known models and techniques, such as Porter's five forces, strategic group analysis and competitor profiling can be used to steer the management's attention. This should result in an understanding of the desired focus and position for the company and its products as well as guide in technology selection.

The third step is to revise the product vision(s) based on the analysis conducted, and distil these as product roadmaps taking internal factors of the company such as human and financial resources, competencies and infrastructure into account. Construction of the product roadmaps should start from defining the major and minor release cycles and continue with defining the business requirements and expectations for the upcoming releases.

By including business requirements and their objectives explicitly into the requirements repository and keeping track of their history, the rationale behind roadmap evolution becomes visible.

TABLE I. STEPS FOR CREATING AND UPDATING PRODUCT ROADMAPS

Step	Objective
Define strategic mission and vision. Outline product vision.copy	Clarify and communicate what business the company is in
Scan the environment	Choose position and focus, assess the realism of the product vision and examine what technologies should be used
Revise and distil the product vision as product roadmaps.	Establish release cycle, objectives for releases and allocate resources. Record decision rationale with business requirements
Estimate product life cycle and evaluate the mix of development efforts planned	Check sanity. Assess whether the planned development is parallel to the product vision

The final step is to state expectations regarding the life cycles and financial implications of product releases, components and platforms, and consider the mix of planned development activities from the business objectives' perspective [12]. This acts as a financial sanity check and evaluates whether the planned development is parallel to product and company vision.

IV. EXPERIENCES

We have developed and applied our model in co-operation with three small software companies, which we call ToolCo, TeamCo and MobAppsCo. ToolCo specialises in the development of applications and software development tools for Internet-, intranet- and extranet environments. TeamCo offers mobile operators, service providers and enterprises solutions that facilitate group interaction. MobAppsCo provides mobile business solutions and professional services for mobile operators and enterprises. ToolCo had 14 employees, and both TeamCo and MobAppsCo were standing at roughly 40. Some common denominators for the three companies are the sizes of their product development organisations, product-orientation in their current (or desired) business models, and inexperience in planning new product development. Below we summarise the experiences and lessons learned from the cases.

A. Conducting Roadmapping at ToolCo

When we started working with ToolCo, the company had envisioned a product concept based on its software toolkit for rapid creation of browser-enabled user interfaces and managing the presentation of information to users. Putting together a toolkit to help in performing the project work, the main source of revenue at the time, had been a conscious effort for the past three years, but its commercialisation was a more recent idea and little over one year old.

Roadmapping for the product was conducted over a period of four calendar months. The work was mainly carried out by the CEO, and required about one man-month of effort. The most important results of creating the initial roadmap were a clearer understanding of what had to be achieved in order to launch the product, and realising the schedule and timing implications of sales, marketing and other aspects not directly related to development. This involved planning the release cycle, the schedules for the major releases and their contents, and considering what whole-product issues needed to be taken into account along the way. ToolCo got a more complete view on what they had at the moment, what was missing, and what would be a realistic schedule for launching and subsequently improving on the product. Especially schedules were revised during the process. The CEO was positive he would use a similar approach in the future for product and release planning.

Concerning the roadmapping process, estimating the life cycles and financial implications of products, components and platforms was seen both important and challenging. Also, identifying and analysing the competition was found difficult. However, the moral of the exercise is in forcing the management to think ahead and coercing them to state their current expectations, rather than in obtaining accurate forecasts of future cash flow or competitors' strengths, weaknesses and plans.

The visualisation was found helpful because it showed the development of the product, its parts and the resource allocation over time in a single picture. These issues had previously been found difficult to express and communicate. The feedback on the visualisation resulted in several changes, with the most important ones being the introduction of the service layer, including explicit resource types, and simplifying the notation for minor releases and release composition.

B. Experiences from TeamCo and MobAppsCo

At TeamCo, we used the notation to discuss the company's plans for its products. At the start of the study, TeamCo had just released a major version of their product, TeamMaster, and the exact schedule, content or role of the next release had not yet been planned. Based on the discussions with TeamCo's managers, we prepared an example roadmap to demonstrate the use of the model using TeamCo's own plans. However, neither these, nor any other plans regarding future releases expressed at that point were subsequently followed. Instead, the key development resources were caught up servicing the current customers, for example installing the system, doing systems integration, customer-specific tailoring, consulting and training. Six months later, TeamCo filed for bankruptcy.

Besides the impact of servicing the current customer base, another lesson from the TeamCo case was the need for product conceptualisation before the roadmap visualisation could be used. This means finding a common language to refer to the components of the software and their relationship to the envisioned product. This had also taken place at ToolCo when the initial version of the visualisation was being developed. Both of these cases suggest that a common conceptual view of the product required for product and release planning may be lacking even when the organisation is small.

At MobAppsCo, our study was intentionally limited to discussing our model and the company's practices in the area of release and product planning. MobAppsCo usually launches product development projects based on the needs of some pilot customer, and the end result is integrated back to the product. The platform is altered according to the functional and non-functional requirements encountered in these projects. As long as the correct focus in selecting the projects can be maintained, this practice is a good example of utilising synergies between the product and services, in this case, to share risk. However, this is not possible when developing a completely new kind of solution because customer feedback is not available from the start.

In the past, MobAppsCo's management had conducted roadmapping by writing a document that described as closely as possible the platform, the set of applications and their features as a function of time. However, the approach felt too cumbersome, and the document was not kept up-to-date. Since then, the practice has been scaled down to having one or two bulleted pages with basically the same information but with less detail and a shorter time range.

MobAppsCo's management considered it feasible to apply the visualization for co-ordinating the more traditional R&D-type work with prioritising, selecting and planning customer-initiated development projects. This could help in communicating the schedule and resource implications better.

C. Lessons Learned

In our study, product roadmapping seemed to help in bridging the gap between management, marketing and product development. Our study also suggests that including servicing in the product roadmap is crucial. Somewhat surprisingly, we also learned and that a common conceptual view of the product may be lacking even when the development organisation is small. In our primary case company, using the roadmapping approach forced management to consider both product positioning and development aspects at the same time, which helped in making resource allocation trade-offs between product development and servicing.

V. Conclusion and Future Work

In this paper we have presented a model for visualizing whole product roadmaps in the context of software development. We also presented experiences of applying and discussing our model at three small software companies. The product roadmap visualization expresses the release and development schedules for the product, composition of individual releases, changes to the underlying technology, services requiring attention from product development and planned resource usage, while project management tracks how successfully the roadmap is being acted on. By addressing these elements, product roadmapping seems to help in concretising and communicating the plans so that they can be acted on – or refuted – when necessary. Tracking service development and actual servicing jointly with product and release planning may help exploit potential synergies between the product and the services offered.

Currently, we are interested in more empirical experience from conducting long-term planning in small companies. However, because a sense of urgency is always present when dealing with small companies, long-term planning may not be the keyword of choice in discussions with potential case companies. In our experience small companies are usually on the lookout for free tools that could replace spreadsheets as the project management tool of choice. We believe providing roadmapping capabilities in such a tool may offer a suitable back door for continuing the research. Towards this end, we are implementing the roadmapping visualization in *Agilefant* (www.agilefant.org), an open source tool for managing software development efforts [16].

References

[1] Bosch, J. (2000). *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*. Addison-Wesley

[2] Weerd, I., Brinkkemper, S.., Nieuwenhuis, R., Versendaal, J. and Bijlsma, K. (2006) "Towards a Reference Framework for Software Product Management". *International Conference on Requirements Engineering*, 312-315

[3] Carlshamre, P., Regnell, B. (2000). "Requirements Lifecycle Management and Release Planning in Market-Driven Requirements Engineering Processes". *Proc. 11th Int'l. Conference on Database and Expert Systems Applications (DEXA 2000)*, 961–965

[4] Clements, P., Northrop, L. (2001). *Software Product Lines: Practices and Patterns*, Addison-Wesley

[5] Cusumano, M., Selby, R (1995). *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets and Manages People*. The Free Press

[6] DeGregorio, G. (2000). *Technology management via a Set of Dynamically Linked Roadmaps. Motorola, Inc*., Motorola Labs Software and System Engineering Research Laboratory

[7] Hoch, D., Roeding, C., Purkert, G., Lindner, S., Müller, R. (2000). *Secrets of Software Success: Management Insights from 100 Software Firms Around the World*. McKinsey & Co.

[8] Kamsties, E., Hörmann, K., Schilch, M. (1998). "Requirements Engineering in Small and Medium Enterprises", *Requirements engineering* Vol. 3, No. 2. 85–86.

[9] Kostoff, R. N., Schaller, R. R. (2001). "Science and Technology Roadmaps", *IEEE Transactions on Engineering Management*, Vol. 48 No. 2. IEEE (2001) 132–143

[10] Lehtola, L., Kauppinen, M. and Kujala, S. (2005). "Linking the business view to requirements engineering: long-term product planning by roadmapping", *Proceedings of the 13th IEEE International Conference on Requirements Engineering*, p. 439–443

[11] Moore, G. (1991). *Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers.* HarperCollins Publishers

[12] Nambisan, S. (2001). "Why Service Businesses are not Product Businesses", *MIT Sloan Management Review*, Summer 2001. Massachusetts Institute of Technology 72–80

[13] Rönkkö, M., Eloranta, E., Mustaniemi, H., Mutanen, O. and Kontio, J. (2007). *Finnish Software Product Business: Results of the National Software Industry Survey 2007*. Helsinki University of Technology

[14] Sommerville, I. (2001). *Software Engineering, 6th edition*. Addison-Wesley / Pearson

[15] Tabrizi, B., Walleigh, R. (1997): "Defining Next-Generation Products: An Inside Look", *Harvard Business Review*. Nov-Dec 1997. 116–124

[16] Vähäniitty, J. and Rautiainen, K. (2008). "Towards a Conceptual Framework and Tool Support for Linking Long-term Product and Business Planning with Agile Software Development". *International Workshop on Software Development Governance (SDG2008)*.

[17] Vähäniitty, J. (2006). "Do Small Software Companies Need Portfolio Management?" *Proceedings of the 13th International Product Development Management Conference*.

[18] Wells, R., Phaal, R., Farrukh, C. and Probert D. (2004). "Technology Roadmapping for a Service Organization". *Research Technology Management*. Arlington: Mar/Apr 2004. Vol. 47, Iss. 2; pg. 46-51

[19] Wiegers, K.: *Software Requirements*. Microsoft Press (1999)