

Towards an Approach for Managing the Development Portfolio in Small Product-Oriented Software Companies

Jarno Vähäniitty and Kristian Rautiainen
Software Business and Engineering Institute
Helsinki University of Technology
P.O. Box 9210, FIN-02015 HUT, Finland
{jarno.vahaniitty | kristian.rautiainen}@soberit.hut.fi

Abstract

Managing product development activities as an explicit portfolio is crucial to the long-term success of product-oriented software companies. Portfolio management has been studied in the field of new product development for over two decades, but existing approaches transfer poorly to small software companies due to contextual differences. Based on new product development and software engineering literature and three company cases, this paper presents an approach for implementing portfolio management in small, product-oriented software companies, along with initial experiences. The approach integrates portfolio management basics such as strategic alignment, portfolio balancing and go/kill/hold decision-making with modern, time-paced software development processes for the small company context. Our findings suggest that using the proposed approach increases awareness of what projects and other development activities are underway, and how these are resourced. It also helps in making informed decisions and trade-offs when necessary.

1. Introduction

Success in the software product business demands more than just succeeding in individual development projects. Keeping an eye on the big picture and building products at the right time with the right features are at least as important [31]. An effective and continuous process for defining, evaluating and prioritising the set of current and planned product development activities has been identified as crucial to product-oriented software companies' long-term success [13,14,26]. This is roughly the equivalent of *portfolio management of new product development projects* (or portfolio management for short) in the field of managing new product development (NPD). In NPD literature, portfolio management is defined as the decision process for actively revising the set of new products and product development projects. It

encompasses product strategy formulation and enactment, making go/kill/hold and resourcing decisions for individual development projects on an ongoing basis, and conducting periodic reviews of all projects as a whole. [9]

The majority of software companies are small (under 50 people, according to the European Union standard) [16]. These companies play a crucial role in the industry because of their innovativeness, popularisation of new technologies, job creation, promotion of growth and keeping established firms on their toes [35]. Up to one half of small companies founded in any one year are not in business five years later, with inadequate management considered as one of the top reasons for this [2,40]. Indeed, there seems to be a need for supporting deliberate and systematic product development decision-making, such as portfolio management, in small product-oriented software companies [2,12,31].

Software engineering (SE) research has traditionally been primarily technical [18]. It also mostly adheres to the viewpoint of individual development projects as far as management is concerned, thus neglecting the portfolio perspective. The SE community has during the recent years taken two directions to make up for this. The first is research on product lines and families (for example, [3,7,34]). This research addresses software architecture rather than products-to-be-sold, and views portfolio management in a more narrow sense, for example, as a tool that can be helpful when defining which features and characteristics should be covered by a product line infrastructure [7]. However, what product line literature does not cover is how portfolio management can be used for dynamic and business-driven resource allocation or prioritisation of development efforts. The second, and from the perspective of this paper more fruitful direction has been to adopt models and frameworks from literature on NPD, where similar issues have been studied, although in the context of large companies, for more than two decades (for example, [8,20,32,33]). Despite the potential for cross-domain knowledge sharing [27], adopting the lessons learned from NPD to the SE context is still at the beginning. Further, the direct applicability of portfolio

management, as described by the existing literature on NPD, is subject to question in the context of small companies [1]. These limitations are further discussed in Section 3.3.

This paper draws on new product development and software engineering literature as well as experiences from industry to help small, product-oriented software companies conduct more systematic product development decision-making. Section 2 presents the specific research questions and the methods used in seeking answers to them. Section 3 presents the state of practice of portfolio management in small software product companies and the limitations of existing work in addressing the problems and challenges found. Section 4 presents the developed approach, and Section 5 presents results from using it in three small software companies. Section 6 concludes the paper with discussion and directions for further research.

2. Research questions and methodology

This paper aims at answering the following questions:

- 1) What is the state of practice of portfolio management in small, product-oriented software companies, and what are the related problems and challenges as perceived by the practitioners?
- 2) To what extent is state of the art literature helpful in solving the problems and challenges found, and what are the main gaps between existing literature and practitioners' needs?
- 3) How can these gaps be overcome in the context of small, product-oriented software companies?

Answering these questions involves a literature review and investigating portfolio management within its real-world context. For the latter, we conducted multiple qualitative case studies [37] and observed real-time portfolio management decision-making. The case companies, here referred to as *Achilles*, *Hector* and *Odysseus*, are all small, with 12, 14 and 7 developers, respectively. The selection of the three case companies was opportunistic and based on accessibility, i.e., the companies' participation in our research project.

Question 1 was addressed through examining how the case companies' product development portfolios and long-term product development plans are managed, and what the problems and challenges involved are. From August 2003, Hector let us attend as observers in portfolio management meetings that ranged from 2 to 3 hours each. We took notes and after each meeting asked for necessary clarifications and discussed possible insights with the head of product development, who also attended the meetings. This helped us understand the nature of portfolio management in its real-world context and find the right focus as well as the right questions to ask at Achilles and Odysseus, where participation in portfolio-level decision-making meetings was not possible. In these

companies, we conducted structured monthly interviews of 1-2 hours in length. The interviews were tape recorded. The interviewed persons were responsible for managing product development. In each meeting we asked how the companies were doing portfolio management, the justification for possible changes to the process, as well as the current problems and challenges. This interview period started in November 2003 at Achilles, and in January 2004 at Odysseus.

In parallel to conducting the monthly interviews and observation, we addressed question 2 by conducting a review of literature. We discussed our findings with key personnel in the companies in order to understand the practical applicability of existing models and practices. This gave us additional insight into how the companies saw their portfolio management process and helped us identify some limitations of existing research.

We addressed question 3 through participative action research [30]. In parallel with answering questions 1 and 2, we outlined a preliminary approach for implementing portfolio management in small, product-oriented software companies to obtain a rough roadmap for improvement work in the case companies. Together with key personnel, we identified courses of action to take in each company to improve the portfolio management process. Each month we evaluated the progress of the improvement work, identified new challenges and courses of action to take, and refined our approach to reflect the insights and lessons learned.

In May 2004 we conducted in-depth interviews with the product development managers of each company to obtain a snapshot of the state of the portfolio management process, the current composition of the development portfolios, and the perceived benefits from the improvement work so far.

3. State of practice and related work

The state of practice in the case companies and the perceived problems and challenges at the beginning of the co-operation are presented in Section 3.1. After that, existing literature on portfolio management is reviewed and its limitations are outlined in Sections 3.2 and 3.3.

3.1. State of practice in the case companies

At the start of the co-operation, none of the case companies managed their development activities as an explicit portfolio, had long-term plans for their products, or explicit priorities for their development activities. Not surprisingly, all of the case companies had experiences of what can be termed as inadequate portfolio management, and agreed that their processes in the area needed improvement.

Important product development decisions were often made based on the opinions of key personnel without explicit discussion or justification. Sometimes decisions were not made deliberately, but rather inadvertently, or even through inaction. The key personnel were generally not aware of the gamut of important decisions they faced in managing their product development activities.

When product development has to be funded by cash flow, having sources of revenue besides licence sales may be necessary. In all of our case companies, the developers' attention was divided between developing new product releases, doing customer-specific development, customer deliveries, and other services such as consultancy. Our case companies also consciously developed new features in customer-specific projects in order to include the results as part of a later product release. Two of the companies had their developers doing consulting, and one even had a significant percentage of its entire development staff contracted to other companies. Because different activities were not necessarily explicitly managed as projects or even recognised as part of the product development portfolio, resource planning was seen as difficult, and it was often omitted entirely because of its perceived futility.

The topmost driver for improvement in all three case companies was the need to understand the big picture, that is, to get a common understanding of what projects and other development activities were underway, how these were resourced and what their relative priorities were. All of the case companies suffered from overbooking of resources, and with the big picture missing, the developers' efforts could not be systematically re-scoped or re-prioritised. Instead, the developers had to rely on personal judgement in deciding what tasks to do and what to drop. As a result, some important activities did not receive enough attention, which caused surprises later in the projects. Also, understanding what implications decisions in one project had on the other projects in the portfolio was identified as a challenge. Decisions were being made on the level of individual projects, with their effects rippling across the entire portfolio and causing a cascade of new decisions.

Another area of improvement was the dialogue between Business (in other words, top management and sales) and Development regarding the direction of product development. At Achilles, dialogue was scarce, and combined with a lack of long-term plans for the products, product development efforts had little explicit consideration for business case. Although this was a cause for concern for the interviewed managers, they did not perceive it very alarming since more than half of the revenue had always come from servicing. At Hector and Odysseus, dialogue between Business and Development was more active, but the downside were impulsiveness in setting up new projects and constantly shifting priorities.

This often led to not getting old assignments finished, as they were overridden by new ones on an ad hoc basis.

3.2. Portfolio management in the NPD literature

The concept of portfolio management is not new. It has appeared over the decades under various names, such as R&D project selection, prioritisation or resource allocation. The focus of research was long on developing quantitative techniques and methods for project evaluation, selection and prioritisation, but there is little evidence on the transfer of these into management practice even in large companies. The main reason for this may be that while portfolio management is essentially about resource allocation, the inherent complexity of the issues involved keeps it far from being a mechanical exercise. [9]

Successful portfolio management is about achieving a balance between the four potentially conflicting goals of 1) *maximising the financial value* of the portfolio, 2) *linking the portfolio to strategy*, 3) *balancing* it on relevant dimensions, and 4) ensuring that the *total number of ongoing activities is feasible*. Different portfolio management techniques, such as financial and economic models, scoring models and mapping approaches emphasise these goals differently. [11,24]

In practice, portfolio management is realised through integrating techniques for project evaluation, selection and prioritisation with a phased review process for ongoing development projects [11,24,36]. Phased review processes organise product development projects into a sequential set of phases having themes (beginning with idea generation and ending with product launch and maintenance), with a corresponding business and prioritisation decision point (i.e., the review) at the end of each phase. Development work is conducted during the phases, along with gathering the information needed to pass the next review. [9,10]

Literature describes two basic alternatives for implementing the portfolio management process in practice. The first one emphasises decision-making through in-depth reviews for each ongoing project and manages the portfolio in a bottom-up manner (e.g. *gates dominate* [11] and *model I funnel* [36]). The second is top-down, with decisions based on looking at the portfolio as a whole (e.g. *portfolio reviews dominate* [11] and *model II funnel* [36]). The first mentioned is better suited for larger firms in mature businesses with dedicated resources and fairly static portfolios, because the emphasis is more in making sound go/kill decisions for individual projects than re-prioritising the entire portfolio every few months. Likewise, the second is more appropriate for smaller companies in fast-paced and fluid markets because it allows for more dynamic resource

allocation through periodically reviewing the entire portfolio. [11,36]

3.3. Limitations in the existing literature

We have identified the main problems in applying existing literature as arising from 1) the differences between the large and small company contexts and 2) lack of examples on how to implement portfolio management with time-paced software development life-cycle models. These are further explained in sections 3.3.1 – 3.3.2 below.

Possibilities for misconception about the appropriateness of phased review processes for software development exist as well. First, phased review models are much reminiscent of the waterfall life cycle model [29] for software development, which is today deemed as inappropriate for many contexts [23]. Second, the phased review models are generally more descriptive of the life cycle of producing a completely new product than of making a new release of an existing one [20]. This seems especially true for the front-end phases of idea generation, preliminary investigation and business case building.

These misconceptions, perhaps stemming from the processes' original design for the context of manufactured products, have recently been addressed and corrected in SE research. First, phased review models are about *business decision-making*, which has traditionally received little attention in the context of software development processes [33]. Thus they do not prescribe the life cycle model used for software development itself, thus in principle allowing for iterative and incremental software development processes [20,33]. The second misconception can be accounted for by correctly emphasising different phases for different kinds of development work or releases [20].

3.3.1. Differences due to company size. In large companies, portfolio management means deciding on a multitude of development project opportunities concerning different product lines and their extensions, possibly across several business units [1]. In contrast, portfolio management in small companies emphasises the contents of upcoming releases for relatively few products [15] and the resourcing of different kinds of development-like or even non-development activities that require the developers' attention. Because, as explained in Section 3.1., there are many responsibilities per person in small organisations, resource planning and allocation are also more volatile. The issue of technical resources having multiple and potentially conflicting roles and responsibilities is also noted but not further discussed by Cooper, Edgett and Kleinschmidt [11]. We presume this is due to its lesser severity in the context of larger companies.

In small companies, resources are more limited in the absolute sense. This difference is illustrated by a quote from [10] describing the preferred course of action in a large company context when time-to-market is considered essential: *“Deadlines must be regarded as sacred ... Delays are dealt with via extra input of effort and resources, not [deadline] postponement.”* This strategy is less likely to work in small companies, or software development [4], for that matter.

In small companies senior management participates more in the hands-on work [7,21]. From the perspective of portfolio management, this suggests that added value from holding in-depth phase reviews [36] is lower because the managers tend to be more aware of what is going on at the floor level. On the other hand, understanding the big picture as well as explicit business decision points are still needed – otherwise managers may end up making portfolio level decisions inadvertently [31]. The role and emphasis of phase reviews should fit the size and the needs of the organisation, but little guidance exists on how to implement these in the small company context.

3.3.2. Differences due to pacing. In the software product business, there is typically major pressure on time-to-market, and release cycles have shortened from years to several months or even weeks [6]. Many modern approaches to software product development respond to these demands by being iterative, incremental and time-paced in nature [23]. *Time pacing* means that actions at different levels – ranging from entering new markets, creating new products, making new releases, and introducing new services to managing the development work itself – are done according to a pre-determined rhythm (fixed schedule), with scope adjusted as necessary. Time pacing is considered a success factor especially when developing products for turbulent markets. In contrast, event pacing drives evolution according to occurrences, such as moves by the competitors, changes in technology, customer demands, or the completion of a specific set of features. [5] Of the software engineering triangle of time, scope and cost [4], event-paced development decision-making emphasises (in order of relative importance) meeting scope, cost and time, whereas in time-paced development, this is reversed. While time-paced development requires careful prioritisation and examination of dependencies, it seems a more feasible approach for software product development with fairly limited resources.

Unfortunately, the examples of portfolio management provided in literature either imply event-paced development processes [11], or simply lack detail on how to implement the portfolio reviews in the context of time-paced product development (for example, [24,25,36]). To provide an example, according to Cooper, Edgett and

Kleinschmidt ([11], p.359), even reviews of the entire portfolio are triggered by a single development project reaching its phase review. This in turn requires having a prescribed set of deliverables ready for examination, implying that the portfolio management process is event-paced regardless of whether the emphasis is on portfolio reviews or in-depth reviews of individual projects.

4. Towards an approach for portfolio management

To address the problems facing practitioners and overcome the limitations of existing literature, we are in the progress of developing an approach for implementing portfolio management. The approach is designed to be suitable for managing a mixture of time- (e.g. release-based and customer-specific product development) and event-paced (e.g. consulting and deliveries) activities in the context of small software product companies.

The approach consists of four steps. First, the development activity types that comprise the portfolio are identified. Then targeted spending levels are set. Third, suitable ways to manage the different types of development activities are identified in terms of rhythm and control points, and the increment control points are synchronised where possible, thus forming the basis for the portfolio management process. Last, the portfolio management process itself is set up through defining control points to govern the portfolio. Figure 1 below depicts an example of using this approach to manage a development portfolio. It identifies five development activity types, their rhythm and control points, target spending levels, and the portfolio management process with three types of meetings (namely, *Roadmap Revisions*, *Portfolio Reviews* and *Fire Brigades*).

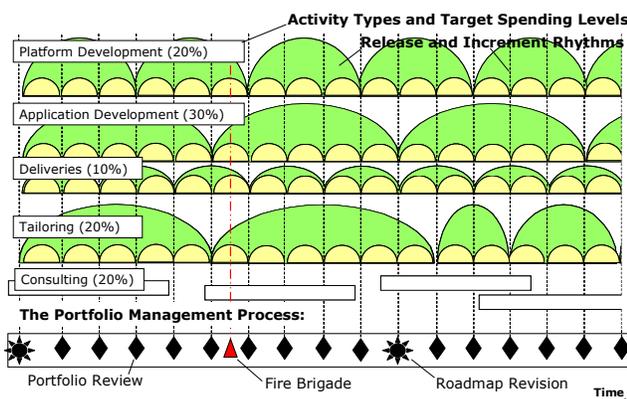


Figure 1. A managed development portfolio

Sections 4.1–4.4 discuss the steps of the approach further, and Section 5 provides examples of how they have been realised in the case companies.

4.1. Identify development activity types

In setting up portfolio management, the first step is to identify the different *types of development activities* the developers are attending to. This creates a framework and terminology for thinking about and discussing the performed activities as an explicit portfolio [36]. In small companies the responsibilities of a single developer may range from consulting assignments, customer tailoring and maintenance to developing new product releases and research of leading edge technology. Categorising the portfolio by identifying the development activity types of the company helps in seeing how each planned or ongoing activity contributes to the big picture, and makes it easier for management to decide what mix is currently appropriate and why. Because of the high degree of resource sharing in small companies, the portfolio should include all of the activities that require attention from the developers, whether or not these activities actually involve “product development” in the strict sense. This step helps in linking the portfolio to strategy and balancing it, as well as ensuring that the total number of activities is feasible (portfolio management goals #2, #3 and #4 from Section 3.2, respectively).

4.2. Set target spending levels

The balance of how the available resources are spent on the different development activities should reflect the acceptable risk level and the strategy of the company [25]. Establishing target spending levels means deciding how much should relatively be spent on each activity type. Once this is done, the specific tasks or projects in each activity type can be prioritised against the level set. Like the previous step, setting target spending levels for effort types and reflecting on the actual resource usage addresses (portfolio management goals #2, #3 and #4 from Section 3.2).

4.3. Set the development rhythm and synchronise increment control points

Besides the fact that different development activities compete for the developers’ attention, these development activities should in many cases also be managed differently. For example, the process for developing a new major release is likely to have different emphasis than the process for conducting customer-specific tailoring, not to mention making customer deliveries or training the customers. In our earlier work we have identified *rhythm* and the resulting *control points* as the backbone for managing software product development [28]. Rhythm supports persistence and forces convergence

while retaining the flexibility to change plans and adapt to changes at specific time intervals in the control points.

Figure 2 shows an example rhythm for release-based software product development. The time horizon for *strategic release management*, in other words long-term product and release planning, spans two *release projects*, each release is built in three increments and the daily work is coordinated and synchronised with suitable practices in *heartbeats*. Each time horizon begins and ends in a control point to plan it or wrap it up, respectively. Specific agendas of control points should vary depending, e.g., on the time horizon and the development activity type. [28]

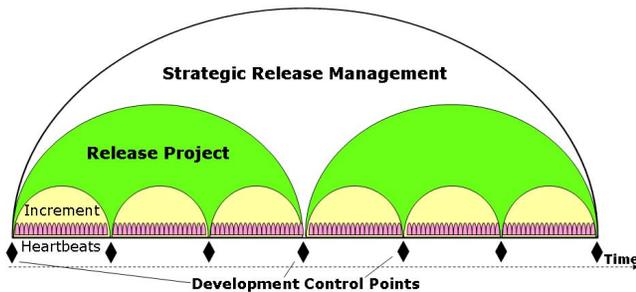


Figure 2. Rhythm and control points in a release-based, incremental development process

In the example portfolio of Figure 1 each development activity has its own rhythm. Setting a suitable rhythm for a development activity entails understanding what kind of rhythm suits the customers of the activity in question. For example, the rhythm of the market directs when product releases should be made. However, the nature of the offering as well as the internal capabilities of the company constrain what is possible. Examples of internal capabilities include the development process effectiveness, personnel skills and how much effort can be spent considering all the other tasks at hand.

Setting a development rhythm creates control points, that (with the exception of the heartbeat level) require the attention of portfolio level decision-making. Thus, as illustrated in Figure 3, there is a danger that the portfolio management process becomes saturated with meetings due to the sheer volume of different kinds of control points required by the development.

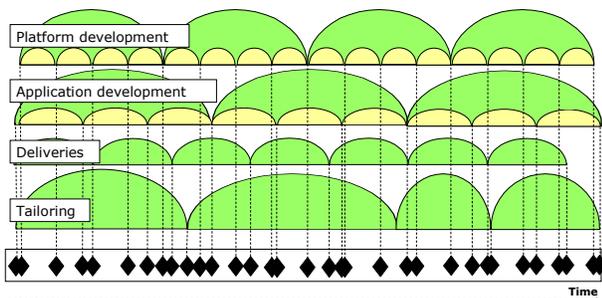


Figure 3. An out-of-sync development portfolio

An out-of-sync portfolio leads to problems in resource planning and allocation. When push comes to shove, the types of development activity with close customer involvement tend to override release-based product development, which in turn makes longer term planning of release-based product development both difficult and frustrating. *Portfolio synchronisation* means organising the control points for different development activities so that the overall portfolio management process becomes as simple as possible. For this to succeed, the rhythm of the entire portfolio should be similar, at least on the increment level. For example, if the longest increment time horizon for any development activity is 4 weeks, the other development activities should have increment time horizons that are 1, 2, or 4 weeks. This way the increment control points of the entire development portfolio are synchronised at least every 4 weeks, which can be used as the time interval for portfolio control points. Even if conducting customer deliveries and doing consulting would not by themselves require control points on the increment level, they should adhere to some kind of rhythm for the benefit of resource planning and allocation for the entire portfolio.

While this step does not directly address any of the four goals of portfolio management in Section 3.2, it is the basis for setting up portfolio management in the discussed context. This is explained further in the next section.

4.4. Set up the portfolio management process

Without a clear process for making portfolio level decisions – in other words, deciding which activities will be resourced and supported – decision-making tends to be reactive, causing shifting priorities, volatile resourcing and short-term fire fighting [17]. Figure 2 illustrated how a certain type of development activity can be managed through control points on different time horizons. In a similar manner, the entire portfolio should be governed with distinct portfolio control points in which the decisions are made, as suggested in Figure 1.

Our approach features three basic types of portfolio control points: *Roadmap Revisions*, *Portfolio Reviews*, and *Fire Brigades*. Roadmap Revisions deal with issues such as product visions and release strategies, and should involve a procedure for long-term planning, such as product roadmapping [22]. Roadmap Revisions require the attention of people from both Business and Development.

Portfolio Reviews set the rhythm for the development organisation as a whole and are the primary mechanisms for linking operations with strategy. Portfolio Reviews look at the ongoing development activities as defined by

their internal control points, but keep the entire portfolio in mind when dedicating resources and setting the scope for the upcoming set of development increments. In this way, fire fighting is minimised as cross-project tradeoffs are made proactively on a continuous basis [19]. Portfolio Reviews require the attention of both Business and Development, but depending on the exact roles and responsibilities, the representation of Business need not be as extensive as in Roadmap Revisions.

Fire Brigades are event-triggered Portfolio Reviews. While the objective in an ordinary Portfolio Review is to freeze the resources and scope for the upcoming set of development increments (see Figure 1), business realities make absolute adherence to this principle impossible. Defining and allowing Fire Brigades increases the likelihood of systematic and conscious decision-making when making mid-increment changes. In addition to making the needed changes in priorities and resourcing, a Fire Brigade should also analyse and record the root cause that led to the situation. This makes it easier to spot similar situations in advance, as well as provides a baseline for estimating how often Fire Brigades are likely to be needed, which in turn promotes realism in planning. When a Fire Brigade is needed, it may or may not be necessary to call the entire team responsible for portfolio decision-making. While the number of people that need to be involved depends on the size of the “fire”, the dominant factor is whether the smaller group of people is able to solve the problem and account for the trade-offs made.

Since portfolio decision-making is ultimately done in portfolio control points, defining portfolio control points – thus setting up the portfolio management process – addresses all of the goals for portfolio management outlined in Section 3.2.

5. Application experiences

This section describes how the approach has been turned into practice in the case companies, the related experiences and the perceived benefits in May 2004.

5.1. Development activity types

During the co-operation, all three companies identified the development activities that constituted their portfolio. These are summarised in Table 1.

Table 1. Activity types at the case companies

Company	Development activity types
Achilles	Product development, consulting, customer-specific projects
Hector	Platform development, application development (on top of the platform), research-like development, customer deliveries and installations, consulting
Odysseus	Product development, delivery and installation projects, delivery and sales support, technology development

None of the companies have yet set target spending levels for the activities. At Achilles, the impression of the authors is that the portfolio is managed for maximising short-term cash flow. At Hector and Odysseus, the goal is a balanced portfolio in terms of long and short-term cash flow, and they both intend to shift from customer-specific development towards a more product-oriented business.

5.2. Development rhythm

Achilles conducts customer-specific projects in two-week increments. The rhythm of all customer-specific projects is kept in sync, but the interviewed manager perceived that this is due to supporting process compliance rather than helping with resource allocation. Product development has no explicit rhythm, mainly due to its currently very low level of spent effort. Should this increase, it would also be conducted in two-week increments. Consulting has no explicit rhythm.

At Hector, application development, both customer-driven and internal, is conducted in one-month increments with weekly builds. Platform development is done in one-week increments and an internal release, timed to the middle of the ongoing application development increment, is made every month. As application development is effectively the customer to platform development, the faster platform development cycle makes it possible to react to feedback and satisfy the requests for the platform with a short response time. Consulting and deliveries do not have an explicit rhythm.

Odysseus had been using a time-paced process for product development for almost a year before this study began. Odysseus’ product development process features long term planning three times per year and conducting the development in one-month increments. The major change during this study concerns portfolio synchronisation. While previously increment contents and goals were set for each ongoing development activity individually, the emphasis is now on understanding the goals for the entire portfolio for the upcoming set of increments. Also, the processes for the other types of development activities have been defined for the sake of achieving a common vocabulary and a better

understanding of the projects' stage and degree of completion.

5.3. Portfolio management process

Every two weeks, Achilles' developers and management get together to review and update the status of the development portfolio in terms of ongoing activities and planned resourcing. Achilles maintains an overview of the product development portfolio with a spreadsheet, with rows representing different projects, columns representing calendar weeks, and the total amount of resource usage planned for each week at the junction. The spreadsheet also has a separate view for a detailed account of resource usage. Tool support for long-term product planning has been improved by implementing product-specific backlogs using TWiki (<http://twiki.org>).

At Hector, a Portfolio Review similar to that described in Section 4.4 is held every four weeks. Between these meetings, a brief status check meeting is held, making the portfolio rhythm two weeks. The product development manager maintains a list of ongoing activities and their respective resource commitments in a spreadsheet, and updates it at least for the status checks and portfolio reviews. Feature lists for the products are maintained in a separate application built on Hector's own platform. Roadmap Revision meetings have been planned to be held every six months to review the different solutions and products offered by the company. In addition to these three types of management team meetings, all of the developers participate in a weekly half-an-hour meeting where everyone quickly goes through what they have done, what they are going to do next, and whether they have encountered any problems. While not a part of the portfolio management process as such, these meetings are used to get an overview of the portfolio, as well as to communicate the decisions made in the management team meetings to the developers.

At Odysseus, the management team reviews the portfolio every four weeks and plans the contents of the upcoming set of increments in a Sprint Planning meeting. The development manager discusses the business priorities with the Chief Executive Officer (CEO) of the company in advance, and thus represents the business perspective in the Sprint Planning meeting. The central artefact is a table drawn on a blackboard. The rows of the table represent each of the seven developers, and the columns represent the four weeks of the increment. Chalks of different colours are used to mark different modules or products, with the text indicating the task in question. Connector lines represent possible dependencies between the tasks. The aim is to keep the blackboard constantly up-to-date with each developer responsible for his own row. In addition to Sprint Planning meetings,

Odysseus employs Fire Brigade and Release Planning meetings. Release Planning is essentially a Sprint Planning that looks at the goals and scope of the entire upcoming four-month time horizon. It takes up to an entire day and the CEO and sales managers participate. Fire Brigades are similar to those discussed in section 4.4. By keeping the blackboard up-to-date, trade-offs that are possible and subsequent ripple effects become visible.

5.4. Experiences and perceived benefits

We found that advances in managing new product development from large companies and other industries can be of use for small software companies. As the result of our joint improvement efforts, the case companies have succeeded in implementing aspects of portfolio management and are satisfied with the initiatives taken. However, it seems that our contribution was significant in helping to interpret the lessons from existing literature to the context. The interviewed product development managers felt that despite the improvements' importance, it was unlikely that the companies would have fared as well or even undertaken such work by themselves. A quote from one of the product development managers suggests an explanation for why these issues might be overlooked in small companies: *"Many of the problems we face seem to have already been solved, but the solutions masquerade as something that looks fit for companies very different from ours."*

At Achilles, the main benefit from the improvement work is an explicit, documented understanding of what the development resources are being spent on. Together with the two week portfolio rhythm, this has alleviated the most significant problems and challenges identified at the start of the study. As the volume of Achilles' product business has halved during the last year both in terms of revenue and effort spending, the new process for long-term product planning has not yet seen much use and the original challenges regarding the Business-Development dialogue (or rather, the lack thereof) remain.

The main benefit at Hector is having an explicit and documented view of what the developers are attending to. While the product development manager admits that there is still work to do with respect to structuring and preparing for the portfolio management meetings, he considers the improvements made an absolute necessity for Hector's operation. Overall, the manager feels that much of the work is now more goal-oriented, and prioritisation is possible. He also feels that having a common understanding of the process for making product development decisions has had a positive effect on the atmosphere at Hector.

At Odysseus, improvement efforts have resulted in forming an understanding of the big picture, as well as providing the managers with a better position to make

informed decisions and trade-offs when necessary. Quoting the interviewed development manager: “*Last week we had to call the Fire Brigade twice, but were able to handle both situations quite easily and conscious of the trade-offs. On the whole, I feel that we have been able to do things that might in the past have caused considerable anxiety without actually feeling too much stress.*” Additionally, while the demand for the attention of certain key personnel remains high, it is now quite difficult to overbook resources without seeing the consequences in advance.

6. Discussion and conclusions

We see the main contribution of this paper as threefold. First, a topic central to strategic software engineering, portfolio management, is identified and discussed. Second, the applicability of portfolio management is examined in the context of *small* product-oriented software companies. Third, an initial approach for integrating portfolio management basics with modern, time paced software development processes in small software product companies is presented, along with preliminary application experiences.

Although to date largely overlooked by SE research, portfolio management holds potential benefits for the field. Our findings – although merely indicative of the situation rather than conclusive – suggest that managing product development as an explicit portfolio is rare in small software product companies, resulting in a number of problems. It seems that these kinds of companies benefit from managing their development activities as an explicit portfolio, but existing guidance to implementing portfolio management is based on assumptions that do not hold for small companies and have an unclear fit with modern software development processes.

The research presented in this paper is a first step in developing an approach for supporting portfolio management in small product-oriented software companies that want to employ modern development processes. Our findings suggest that using the proposed approach increases awareness of what projects and other development activities are underway, and how these are resourced. It also helps in making informed decisions and trade-offs when necessary.

There are limitations to our study. First, the reported usefulness of our approach is based on the perceptions of one person at each case company (the interviewee) and the interpretations of the researchers, not on explicit measures of success or failure. Second, the observation period is too short for evaluating how lasting the changes are, and due to the limited sample size and unknown representativeness of the sample, generalisation of the results is difficult at this point. On the other hand, it seems obvious that progress has been made in each

company because none were managing their product development as an explicit portfolio before the co-operation. Each company has set up an explicit portfolio management process, which is both being used and refined. This is supporting evidence of the practical utility of portfolio management in small companies, and we hope our findings encourage practitioners in similar companies to employ portfolio management.

Because we employed participative action research, researcher bias is a factor to be considered. Both the authors and the interviewed product development managers have a vested interest in producing good results, and negative evidence may have been accidentally overlooked. Also, at this point it is difficult to evaluate the value of the presented approach as such, eliminating the contribution of the authors – for this, a longer observation period and more cases, combined with a different research setting are needed. However, to provide access for further validation of the findings, we have during the study kept records of all discussions, archived all relevant documents and taped all interviews.

In the future, we continue our work in developing the approach. In addition to observing the three original case companies, we are starting co-operation with four new case companies to further deploy and refine the approach. Currently, the approach provides guidance on how to make flexible but controlled resourcing decisions on the portfolio level. However, it does not provide details on how to do long-term product and release planning, which is a crucial aspect of portfolio management. In our future work, we look at why planning for the long term is important even when it seems difficult and identify the time horizons involved, that is, what is meant by short or long term plans as well as how this depends on the business environment. We will also look at the *emphasis* of planning in these time horizons, including how much planning is “enough”. These issues are of importance for managing software product development, but have to our knowledge received little attention in SE research. Based on our findings from literature and experiences from case studies, we are formulating a process for long-term software product and release planning, and will use it to extend the portfolio management approach described in this paper.

7. References

- [1] Artto, K., M. Martinsuo, and T. Aalto, *Project Portfolio Management - strategic management through projects*, Project Management Association Finland, 2001.
- [2] Berry, M., "Strategic planning in small high-tech companies", *Long Range Planning*, vol. 31, no. 3, 2002, pp. 455-466.
- [3] Bosch, J., *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley, 2002.
- [4] Brooks, F. P., Jr., *The Mythical Man-Month: Essays on Software Engineering*, 20th anniversary edition ed., Addison-Wesley, 1995.

- [5] Brown, N. and K. M. Eisenhardt, *Competing on the Edge : Strategy as Structured Chaos*, Harvard Business School Press, 2002.
- [6] Carmel, E. and S. Becker, "A process model for packaged software development", *IEEE Transactions on Engineering Management*, vol. 42, no. 1, 1995, pp. 50-61.
- [7] Clements, P. and L. Northrop, *Software Product Lines: Practices and Patterns*, Addison-Wesley, 2001.
- [8] Condon, D., *Software product management - managing software development from idea to product to marketing to sales*, Aspatore Books, 2002.
- [9] Cooper, R. G., S. J. Edgett, and E. J. Kleinschmidt, *Portfolio Management for New Products*, 2nd ed., Perseus Books, 2001.
- [10] Cooper, R. G., *Product leadership: creating and launching superior new products*, Perseus Books, 1998.
- [11] Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J., "Portfolio management: fundamental to new product success," in Belliveau, P., Griffin, A., and Somermeyer, S. (eds.), *The PDMA Toolbook for New Product Development*, John Wiley & Sons, Inc., New York, 2002, pp. 331-364.
- [12] Crowne, M., "Why Software Product Startups Fail and What To Do About It", *Proceedings of the International Engineering Management Conference (IEMC)*, pp. 338-343, IEEE, 2002.
- [13] Cusumano, M. A. and R. W. Selby, *Microsoft Secrets*, The Free Press, 1995.
- [14] Cusumano, M. A. and D. Yoffie, *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft*, The Free Press, 1998.
- [15] Dver, A. S., *Software product management essentials - a practical guide for small and mid-sized companies*, Anclote Press, 2003.
- [16] Fayad, M. E., M. Laitinen, and R. P. Ward, "Software Engineering in the Small", *Communications of the ACM*, vol. 43, no. 4, 2000, pp. 115-118.
- [17] Gill, B., Nelson, B., and Spring, S., "Seven steps to strategic new product development," in Rosenau, M. D. (ed.), *The PDMA handbook of new product development*, John Wiley & Sons, 1996, pp. 19-34.
- [18] Glass, R. L., I. Vessey, and V. Ramesh, "Research in software engineering: an analysis of the literature", *Information & Software Technology*, vol. 44, no. 8, 2002, pp. 491-506.
- [19] Harris, J. R. and McKay, J. C., "Optimizing product development through pipeline management," in Rosenau, M. D. (ed.), *The PDMA handbook of new product development*, John Wiley & Sons, 1996, pp. 63-76.
- [20] Hohmann, L., *Beyond software architecture: creating and sustaining winning solutions*, Pearson Education, Inc., 2003.
- [21] Jennings, P. and G. Beaver, "The Performance and Competitive Advantage of Small Firms: A Management Perspective", *International Small Business Journal*, vol. 15, no. 2, 1996.
- [22] Kappel, T., "Perspectives on roadmaps: how organisations talk about the future", *IEEE Engineering Management Review*, vol. 29, no. 3, 2001, pp. 36-48.
- [23] Larman, G. and V. R. Basili, "Iterative and incremental development: a brief history", *Computer*, vol. 36, no. 6, 2003, pp. 47-56.
- [24] McGrath, M. Setting the PACE in product development. McGrath, M. 1996. Butterworth-Heinemann.
- [25] McGrath, M., *Product Strategy for High Technology Companies*, McGraw Hill Text, 2000.
- [26] Mello, S., *Customer-centric product definition - the key to great product development*, American Management Association, 2002.
- [27] Nambisan, S. and D. Wilemon, "Software Development and New Product Development: Potentials for Cross-Domain Knowledge Sharing", *IEEE Transactions on Engineering Management*, vol. 47, no. 2, 2000, pp. 211-220.
- [28] Rautiainen, K. and C. Lassenius (eds.), *Pacing Software Product Development: A Framework and Practical Implementation Guidelines*, Helsinki University of Technology, Software Business and Engineering Institute, Technical reports 3, 2004.
- [29] Royce, W., "Managing the Development of Large Software Systems: Concepts and Techniques", *Proceedings of the Wescon*, pp. 1-9, IEEE, 1970.
- [30] Susman, G. I. and R. D. Evered, "An Assessment of the Scientific Merits of Action Research", *Administrative Science Quarterly*, vol. 23, 1978, pp. 582-603.
- [31] Vähäniitty, J., "Key Decisions in Strategic New Product Development for Small Software Product Businesses", *Proceedings of the 29th EUROMICRO CONFERENCE, Software Process and Product Improvement track*, pp. 275-383, IEEE, 2003.
- [32] van Zyl, J. and A. J. Walker, "Strategic product development - a strategic approach to taking software products to market successfully", *Proceedings of the First Software Product Line Conference (SPLCI), August 28-31*, Denver, Colorado, USA: Kluwer Academic Publishers, 2000.
- [33] Wallin, C., F. Ekdahl, and S. Larsson, "Integrating business and software development models", *IEEE Software*, no. 6/2002, 2002, pp. 28-33.
- [34] Weiss, D. M. and C. T. R. Lai, *Software product-line engineering - a family-based software development process*, Addison-Wesley, 1999.
- [35] Wheelen, T. L. and D. J. Hunger, *Strategic management and business policy*, 8th ed., Pearson Education, 2002.
- [36] Wheelwright, S. C. and K. B. Clark, *Revolutionizing Product Development*, The Free Press, 1992.
- [37] Yin, R. K., *Case Study Research - Design and Methods*, Second ed., Thousand Oaks: Sage, 1994.