

Supporting Scaling Agile with Portfolio Management: Case Paf.com

Kristian Rautiainen
Aalto University School of
Science and Technology
kristian.rautiainen@tkk.fi

Joachim von Schantz
Paf
Joachim.vonSchantz@paf.com

Jarno Vähäniitty
Aalto University School of
Science and Technology
jarno.vahaniitty@tkk.fi

Abstract

This paper is a descriptive case study of how one department at Paf, Paf.com, introduced portfolio management to help support scaling agile software development. Paf.com had experienced problems with long time-to-market due to thrashing, which was caused by frequently changing priorities due to an ad-hoc prioritization process and handovers. Also, there was lack of visibility into projects entering and progressing in the development pipeline. No structured way of starting projects was enforced company-wide, and too many parallel projects got started. As a result of introducing a structured portfolio management process, the number of ongoing projects has dramatically reduced, from over 200 to 30, reducing thrashing. Listing all projects in priority order in the Paf.com backlog provides visibility into what is currently ongoing, helping coordinate the work of multiple Scrum teams. The portfolio follow-up function provides progress data on the projects, helping managers make more informed decisions, considering the whole portfolio.

1. Introduction

During the past 10 years, agile software development methods have gained acceptance in the software industry. Originally, the sweet spot for agile software development was one co-located team of 3-8 persons working on one product [3]. Lately, the trend has been in scaling agile to the enterprise level and applying lean philosophy to software development, see e.g. [14-16, 22, 23]. Some amazing results have been reported about applying fully distributed Scrum [28, 29], but not many can boast such impressive results.

The transition from the agile sweet spot to enterprise-scale agility is not easy. Kalliney [9] reports “portfolio problems” when scaling Scrum, specifically regarding product management and enacting the company vision, managing cross-team risks and dependencies, and handling the silos of knowledge and skills.

The process for achieving balanced resource allocation in terms of value maximization, strategic alignment, risk level, and the number of ongoing projects is called new product development portfolio management [5], or portfolio management for short. It has been mostly addressed in literature on new product development (NPD), but lately also in literature on scaling agile and lean. However, experience reports and case studies in the context of lean or agile are scarce. This paper aims to add to that body of knowledge by providing a descriptive case study of how one company, Paf, introduced portfolio management to help support scaling agile software development. We also report on initial results and challenges met.

The paper is structured in the following way. First we present related work on portfolio management from literature on NPD and especially literature on agile and lean software development. Second we describe the research methodology and provide case company background information. In Section 4 we provide the results of the case study. Finally, we sum up the paper with its contributions and suggestions for future work.

2. Related work

2.1. Portfolio management in new product development

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 has been in the area of new product development (NPD) and on developing quantitative techniques and methods for project evaluation, selection and prioritisation. [4]

Portfolio management has been adopted in the industry, but not without problems. A recent study of 30 companies [2] shows that while the companies have adopted portfolio management practices, they still struggle with completing projects within schedule and lack a broad overview of ongoing projects. The main reasons behind this were; (1) very different types of

projects are included in the managed portfolio, and (2) not all projects and smaller activities are managed as part of the portfolio. Therefore as much as 50% of the product developers' time can be allocated to work that is not seen by the portfolio managers. This creates a gap between required and available resources that can go unnoticed. Since portfolio management is essentially about resource allocation, the inherent complexity of the issues involved keeps it far from being a mechanical exercise.

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. [5, 17]

Setting up proper portfolio management has been recognized as challenging even for the best of organizations [19]. In practice, portfolio management is often realised through integrating techniques for project evaluation, selection and prioritisation with a phased review process for ongoing development projects [5, 17, 34]. 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. [4, 6]

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* [5] and *model I funnel* [34]). The second is top-down, with decisions based on looking at the portfolio as a whole (e.g. *portfolio reviews dominate* [5] and *model II funnel* [34]). 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. [5, 34]

2.2. Portfolio management in agile and lean software development

Advocates of agile methods tend to see much of the literature on managing new product development as fundamentally incompatible with agile software development because it tends to view development as a separate "phase" in the cycle of realizing an idea into an actual working solution [12]. For example, phased review processes as described in Section 2.1 are by some seen as incompatible with the basic principles of agile development [14, 15]. Whatever the case, integrating agile software development with phased review processes may not be straightforward due to attitudes, as well as a number of actual contradictions in e.g. level of abstraction in planning and feedback cadence [10].

Not much has to date been written on how to set up portfolio management so that the result would be compatible with agile and/or lean principles. Still, the ideas and practices of portfolio management have begun to emerge also in agile software development, mainly through the introduction of lean principles to software development, see e.g. [11, 12, 14, 15, 22, 24, 25]. However, the use of the term portfolio management differs between the authors.

Leffingwell provides in his book [16] and its blog companion (scalingsoftwareagility.wordpress.com) a very high-level view to portfolio management. At the portfolio management level the company's executives define *investment themes* that drive the resource allocation and thus investment priorities of the company. *Epics* or epic-scale initiatives are used to express the portfolio vision in practice. These guide upcoming product releases.

Poppendieck & Poppendieck [22] provides another viewpoint to what could be interpreted as an approach to portfolio management. Possible development efforts that take up people's time are first classified by type, for example as strategic business initiatives, feature upgrades, infrastructure upgrades, and maintenance. Then the desired cycle time for each type of effort is decided. The investment levels for the categories are set by determining how many initiatives of each type should be carried out within, e.g. a year. Or, in the case of e.g. maintenance, a reservation is made of how much of the total capacity the activity should be allowed to take. Finally, the slots for the initiatives are laid out in the calendar in advance. When the time slot of a certain initiative approaches, its actual content is decided on the basis of what is most valuable for the company at that point in time.

The most common interpretation of portfolio management in software development follows the definition from NPD literature; portfolio management

is the activity of making resourcing decisions across a portfolio of planned and ongoing projects. Shalloway's concept of *Lean Portfolio Management* [25] means deciding on a frequent basis across a portfolio of projects how the development resources are allocated for delivering the *minimum marketable features* or *business* features that at the moment provide the most business value. Pichler [20] agrees with Shalloway's idea and recommends that "competing backlogs" should be dealt with in a similar fashion.

Larman and Vodde [14, 15] are along the same lines. They note that in organizations with less than 100 people, prioritizing on the level of the portfolio of products and services offered tends to lead to local optimization. Instead, portfolio management can be more effectively carried out by merging the backlogs of different product/service offerings into a single backlog, and then performing backlog management on that single backlog. This view is shared by Krebs [12], who advocates that the ongoing and planned projects should be kept in a list called the "project portfolio backlog". Decisions on which projects should continue, be put on hold, be launched, or be killed, are then made on a per-sprint basis. A similar approach is also mentioned by Rothman [24]. These approaches seem to assume that the iterations are synchronized for the whole organization and that all projects use an agile software process. An off-synch portfolio of projects can indeed lead to problems, as pointed out in [33].

Only a few experience reports seem to exist to date. Karlström and Runeson [10] report experiences on how two large software systems projects attempted to use eXtreme Programming in the context of a phase-gate model. Tengsøe and Noble [30] report how the *Portfolio Management Office* (PMO) helped balance the demand on Capital One Auto Finance's resources from multiple and sometimes inter-dependent projects. Kalliney [9] reports how Ultimate Software transitioned from agile development to an agile enterprise, setting up portfolio management practices to solve some of the problems related to the transition. Thomas and Baker [31] report on what challenges they faced when applying agile methods to IT investment funding, change management, and governance, and how they managed their projects as a portfolio. Steindl [26] reports how IBM manages agility on three levels; project, portfolio, and business. Laanti [13] describes how a large organization implemented portfolio management as an "outer control loop" on top of the "inner scrum control loop" according to the *Sashimi model* from [18].

3. Methodology

3.1. Research methods

This study is a longitudinal, descriptive case study [35], with elements of action research [27]. The first and third author and other researchers have observed the portfolio management process at Eget and later Paf.com from the beginning of 2008. The second author has worked in the company since August 2008 and he has been responsible for defining the **Paf model** for projects (Pamp) presented in Section 4, and for further refining the portfolio management approach.

The researchers arranged a survey and interviews with selected personnel in March 2008 to investigate the state-of-practice of portfolio management at the company. The survey contained 56 statements related to different aspects of portfolio management and it was answered by 21 respondents. The sampling was purposeful. Key personnel from a diverse assortment of roles and responsibilities (developer and tester representatives, R&D management, business representatives, Product Owners (PO's), Scrum Masters (SM's)) were selected to participate in the survey. Of those 21 people, 7 were further interviewed, using the survey statements as an interview guide. The results of the survey and interview were disseminated to the whole company two weeks after the interviews. After the dissemination of the results of the survey and interviews, the researchers participated as observers in different meetings relating to the development process and later the portfolio management process of the company. They also reviewed documents related to the existing and planned processes and provided feedback.

In September 2010 the survey was re-run, with 28 respondents of which 6 had also answered on the first round. However, new interviews based on the second survey round were not made before the publication of this paper.

3.2. Case background

Paf (Ålands Penningautomatförening), founded in 1966, is a public association that operates gaming activities on the autonomous Åland Islands, onboard Ålandic and Finnish ships and on the Internet. Gaming began on 1st January 1967 in collaboration with the member associations behind Paf: the public health service on Åland, the Åland branch of Save the Children, the Finnish Red Cross and the child welfare foundation Stiftelsen Dagens Barn. This study is focusing on the software development of the Internet gaming activities. Internet gaming on www.paf.fi was launched on 3rd December 1999. The first form of

gambling was betting. Today, the business also encompasses gaming machines, casino gaming, bingo and lotteries, poker, and skill games.

Paf founded the subsidiary Eget focusing on internet gambling in 1999. Paf held the majority of Eget shares. Eget developed and operated Internet monetary games for Paf and other companies. In the spring of 2008 Paf and Eget merged but the setup of buyer and supplier was still around in the minds of the employees. Later, in the spring of 2009, most of the former Eget was to be known as Paf.com, the Internet department at Paf.

Scrum was introduced at Eget around 2006 to solve problems related to quality, release planning and ways of distributing work. Scrum was introduced bottom-up starting with one team. In the fall of 2008 most teams used Scrum “by the book”. At that time the visibility into teams’ backlogs was established, but the backlogs and other documentation of 10+ geographically distributed teams were spread all over the company wiki and in Excel files with no structure. However, the main problem was the visibility into what PO’s and development teams *actually* did, which was a result of unclear and shifting priorities from the Marketing and Sales department due to congestion of the development pipeline. For example, if the capacity of the development pipeline was 5 projects, 5 projects were started, but before these projects were ready and the deliverables were released into production, 5 new projects were started, and so on.

This congestion and thus shifting priorities caused thrashing, which resulted in long time-to-market. This could also be seen as Development-Business disconnect and inefficient and uncontrolled portfolio-level decision making. These had been identified as the two most pressing challenges in a survey conducted by the first author and his researcher colleagues in the spring of 2008, when the collaboration with the company and researchers began.

In a value chain mapping exercise performed by the developers at Eget, the outcome was that even if a game took 5 months to develop on average, the worst case scenario was that it would go into production in a total of 24 months with 19 months of shelf time and handovers. It was also proven that a simple game can be put out into production in 3 months when expedited by C-level managers. The situation in the fall of 2008 was that releases were planned five weeks ahead as a release train [16] and the practice of joint release planning [8, 16] had been taken into use, to make the teams’ plans visible to all the other teams and stakeholders and to reduce handovers.

The problems found are similar to symptoms that have been identified in literature to be associated with inadequate portfolio management [32]. Therefore

setting up practices for portfolio management was selected as one of the improvement paths.

Thrashing was not the only reason for long time-to-market. A monolithic architecture that had evolved over time was also a major contributing factor. Due to the architecture and history, Paf.com operates with component teams: Slot team, Lotteries & Bingo team, Integration team, BackOffice team, Core team, Report team, CRM team, Web management team and so on. Build automation was introduced and further developed to mitigate the effect of the complex architecture and help coordinate the work of the 10+ geographically distributed teams. However, the impact of architecture and build automation or changes in team structures are excluded from this study that concentrates on the initial experiences of introducing portfolio management practices to alleviate the problem of unclear and shifting priorities and thrashing.

4. Introducing portfolio management

4.1. First steps towards portfolio management

The first steps towards portfolio management were taken when the content of every release was internally prioritized according to business value. In every release there were 3-5 upgrades or new components that were committed to be ready to be put into production by the teams. Some effort was spent on training the Marketing and Sales department in the ways of agile planning and prioritizing, i.e. the most important things need to be completed first, so that less important things can be scoped out at the end of the release time box, if need be. Gradually this began to work but then 4 new problems were discovered:

- lack of visibility on “projects” about to enter the development pipeline
- the maturity of the planning of the “projects” entering the development pipeline
- prioritisation of projects was ad-hoc
- rogue, “business critical” projects were induced “under the hood” by some managers

These problems were discovered in the fall of 2008, when the newly created Project Management Office (PMO) made an inventory of all the active projects in Paf development and technical operations (later Paf.com). As a result, 214 “projects” were found, including duplicates and sub-projects. To get visibility into the situation, the PMO started to issue unique project ID’s to existing and new projects. However, it was difficult to get grasp of the existing projects, since they were unstructured and very little documentation was available.

In the spring of 2009 the management of Paf was still not happy with the inefficiency and handovers in the organization inherited from the merger of Paf and Eget. Cooperation negotiations were held to get a competence shift towards the Internet business. The guiding star of the new organization was less handovers. In the new organization the departments System & Services, Games, Sales, Customer Experience and Customer Relations were formed into Paf.com with responsibility for the Internet gaming business. The component/development teams were divided under the appropriate departments. The idea was that problems and opportunities discovered in a department could be handled by its development teams. Problems and opportunities that are dependent on resources outside the own department were to be handled in a structured manner according to the suggested new project framework Pamp, which is discussed in the next section.

4.2. Paf model for projects (Pamp)

In the spring of 2009, the PMO and QA proposed a model for managing the portfolio of projects in a structured way. It was called Pamp and it combined elements from Stage-Gate models [7], the Open Unified Process (<http://epf.eclipse.org/wikis/openup/>), and PMBoK [21].

The main goal of Pamp was to *clarify a business need and increase the visibility* for each project in Paf.com. The idea was to tackle the problems identified earlier and described in the previous section. Pamp was not to include guidelines for the actual project execution process, which was left for the

team(s) and project manager to decide. Typically, the execution was done using the existing Scrum process.

Figure 1 shows an overview of Pamp. Pamp controls and monitors project planning, prioritization, and execution through 4 project demos (Proposal, Planning, Design, Closing), where the person having the role of project manager presents the status of planning and execution. The time units in Figure 1 are indications of calendar time (the size of which may differ, depending on project size); most of the effort is done in the Execution phase by the development team(s). The level of required documentation and planning is governed by a complexity classification of the projects into complex, normal, or simple. The demos are held as a review and workshop with representatives from the project organization and senior project and process managers reviewing the documentation. The emphasis is on clarifying risks, dependencies, stakeholders, and business value.

In the project proposal demo, the project manager presents, e.g. one structured PowerPoint slide including information on:

- Business issue/opportunity
- Primary project deliverable(s) or Epic statement
- Business benefits of the project outcome
- Project overview
- Main risks and dependencies

If the proposal is accepted, the project is put into the Paf.com backlog, which is further explained in the following section. The first draft of the project plan is prepared and presented in the project planning demo, addressing the work that needs to be undertaken:

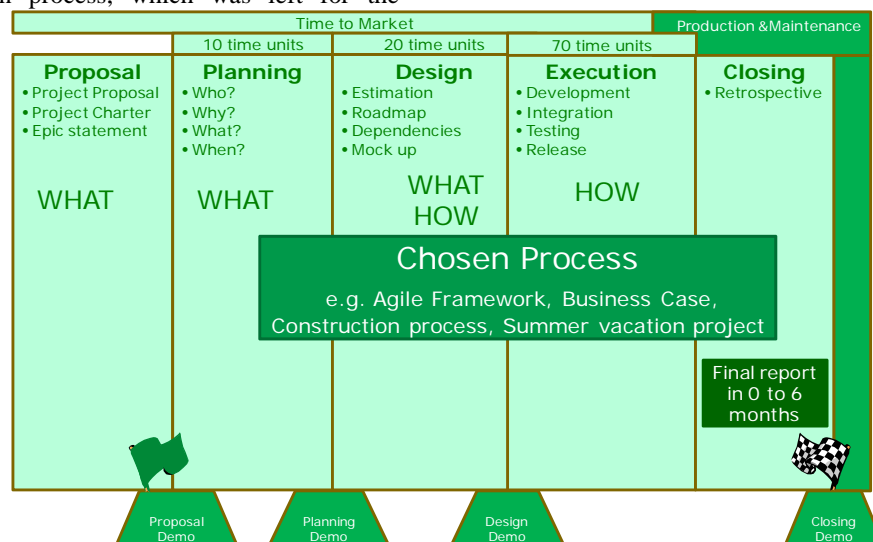


Figure 1 Paf model for projects (Pamp)

- Who will primarily be doing the work
- What are the main deliverables and project scope
- When is the target release date
- Why the work is done, including more refined business and other benefits compared to the project proposal

After the planning demo, project preparations move on to the design demo, where the plan for how the project will be executed is presented, including:

- Roadmap of preliminary epics and sprints
- Initial estimations of the epics
- Resource needs
- Dependencies
- Mock-up to show technical feasibility in more complex projects

The final, closing demo is arranged after the project has ended. The project's success is reviewed and a retrospective is held.

During the project execution period, the follow-up is governed by the approved project plan, which is a living document that is updated and "re-approved" throughout the project. While this sounds like a plan-driven approach, it is still similar to the roll-out planning of Scrum. The extra planning is needed for coordination when scaling agile to development projects involving several teams, especially if the teams are component teams and a switch to feature teams is not feasible in the short run. The teams still do the actual value-adding work using Scrum and burn-down charts are used to evaluate progress. If the burn down shows problems in reaching the desired scope, corrective actions are taken as deemed appropriate, and a re-evaluation of project value can be performed. In this way the ranking of the project might fall (or rise) in the portfolio review, arranged once a month.

The project manager (PM) is a role that a PO can have during a project. When a project needs multiple teams, a project manager is appointed to handle project-related matters not related to the PO work, such as budgeting for multiple teams, handling reporting to the steering group, and facilitating the communication and problem solving between the PO's in the project. For a cross-departmental project the final say is with the PM unless it is escalated to the project owner (usually a department head). In a typical setup the PO that is most central in the project and has the biggest need to get the project done is made PM for the project and through the Paf.com backlog prioritisation (described in the following section) (s)he gets the other PO's support and resources. Usually the PO appointed as PM does not have full knowledge about the domain that the supporting teams are developing. Therefore the

PO responsible for that domain is handling the backlog for that supporting team.

The effort of using Pamp for giving visibility to the work and helping in the planning is small compared to the situation of not having information for portfolio management. In general, writing the proposal takes a couple of hours and preparing the plan for the project a couple of days. Feedback has shown that it is faster if the project manager has a clear view on the work that has to be undertaken. The proposal and the plan for the project are based on templates that also function as checklists for planning the work and presenting possible business value for Paf.com management. The rule of the thumb is that the project manager has a lot of freedom in designing the work as long as the highest risks are mitigated to a certain extent and the way of working is documented. Statements like "according to company process X", "will be clarified in week 5 of the project" or "according to team estimates" are valid.

The demos take approximately 1 hour or less, if the project manager has prepared herself properly. The senior experts invited to the demo ask the project manager to clarify questions around the main points of the project plan. This is done to make sure everybody understands what the project is about, so that the portfolio decisions are based on the best possible data.

Next we take a look at the Paf.com backlog process, which is the portfolio management process of Paf.com.

4.3. Paf.com backlog process

The backlog process was first drafted in 2008 and in the beginning of summer 2009 the current version shown in Figure 2 was taken into use. The process addresses the problem of managers inducing "business critical" projects "under the hood". All Pamp projects are in the Paf.com backlog. Also projects that are not using Pamp are visible. Projects not using Pamp are, for instance, new games, security patches, or general updates, which are done from the team's or department's own backlog. The Paf.com backlog is a simple table placed in the intranet of Paf.com. It shows project priority, ID, name, sponsor (Paf.com management representative), manager (person taking the role of project manager for that particular project), status, estimated completion date, and capitalization information. Finance is monitoring the Paf.com backlog to follow up on costs from projects and released value-adding content.

The 5-10 most important projects are prioritized, out of roughly 30 (which is considerably less than 214 in the fall of 2008), by Paf.com management (consisting of department heads and the Paf.com director) on a monthly basis in a portfolio review

meeting, with the PMO owning the prioritization backlog called Paf.com backlog. The reason for Paf.com management to prioritize the projects is that they control the resources and need to commit to assigning their resources to a specific project. The person having the role of project manager can and is obliged to ask for resources from projects ranked lower in the Paf.com backlog to speed up project completion. The driving force is to concentrate on a few things at a time so that projects can be done fast to deliver value and thus help achieve shorter time-to-market. This is in accordance with lean principles [23].

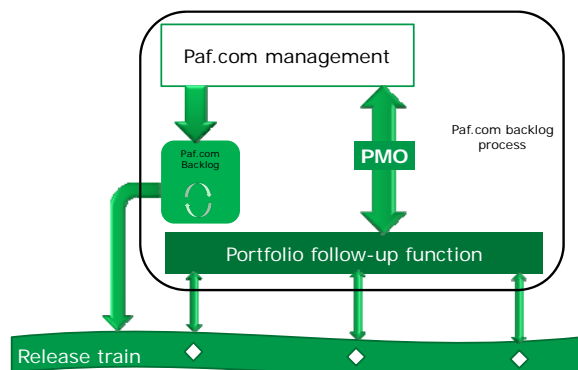


Figure 2 Paf.com backlog process

Resourcing can be done by giving work to other teams or by taking individuals into the team working with the project with a higher priority. Company-wide changes to platforms or processes are prioritized in the Paf.com backlog in the same way so teams know what to focus on in the company sprint of 5 weeks (release train) and what is expected from them when the release is done. Joint company sprint planning is used as a practice to make it possible to more easily negotiate the resources for each company sprint. The teams and individuals can easily block requests that are not connected to a higher priority project. If there are no requests, the team can work on their assigned project even if it is not prioritized. The teams' sprints (typically 2 weeks) are not synchronized and every team has the freedom to plan their work within the company sprint as they see fit. Only those teams that have their own component in an upcoming release participate in the releasing activities. On average during 2009, 8 out of 16 teams were involved in making a production release. The smallest amount of teams participating in making a production release was 4 and at one occasion 12 teams participated.

Figure 3 shows the backlog hierarchy at Paf.com. Pamp and the Paf.com backlog process are only concerned with the two left-most backlogs, which provide input to the teams' backlogs. You could also

say that the Paf.com backlog and the project backlog show business priorities and high-level plans, whereas the teams' backlogs contain the actual development work.

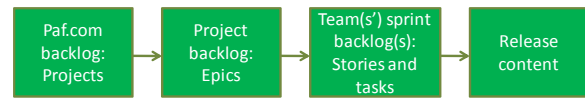


Figure 3 The backlog hierarchy

The PMO monitors project progress based on the teams' story point estimates and burn down charts, and reports to Paf.com management as part of the portfolio follow-up function. Story point estimates reflect the complexity of the story to be implemented and do not account for how long it takes to implement the story. The estimation is done by the teams using standard planning poker with Fibonacci series numbers. Story points per sprint, normalized by sprint length gives an approximation of velocity from which a forecast of project completion can be derived, when the story points of unfinished stories are known. The forecast is calculated based on the mean value of the story points in the last three completed sprints.

When several teams work on the same project in different sprint schedules with different velocity and differently sized backlogs, the forecasts are made on team level. The team with the completion forecast furthest away is used as the current forecast for completion of the project. The forecast gets more uncertain as the complexity of the project grows, i.e. more teams, new technology, or complex solutions are involved. Forecasted date and targeted date are not to be mixed. The difference is basically how a project was planned to go and how is it going time wise. The forecast can be used like a project burn down, signalling when there is a need to re-scope the project or extend the schedule of the project.

A web-based backlog management tool was introduced in early spring 2009 for all the teams to gradually take into use as their primary reporting tool to show the team's stories and their progress in respect to the release train. In the fall of 2009 all teams were using the tool. The aim was to provide a unified way of measuring and visualizing progress of work and effort left undone in the projects. The information is used as background in the portfolio review. From a development project's viewpoint all necessary data is available at a glance; amount of story points done and not done, velocity, and forecast. Additionally, the size of releases of the release train can be estimated based on complexity, not time spent on coding. SM's and PO's are the main users of the tool. The PO's keep their backlogs in the tool and SM's update progress information, taking them an average of about 5 minutes

per day. The main tool for internal communication and information is the team's own Scrum wall in the team room. One of the main reasons for additional, web-based reporting is that the teams are geographically dispersed to seven offices in four countries. The information that is needed, e.g. to plan and coordinate the release train, cannot thus be collected just by walking through all the team rooms and reading the Scrum walls.

4.4. Initial results from introducing portfolio management

While explicit portfolio management has only just been introduced at Paf.com and the investigation to its possible drawbacks and benefits is still ongoing, we can provide some initial results. As already mentioned, the amount of ongoing projects has gone down with one order of magnitude from 214 to 30. Even if the 214 projects included sub-projects and duplicates, the reduction is considerable and can be attributed to the introduction of Pamp, the restructuring of the backlog process, and improved planning practices, which also have helped in reducing handovers. Pamp is working as a gate for the paf.com backlog, guarding it from too many or too immature projects.

Additionally, from those 30 projects, only 5-10 at a time are indicated as high-priority, making the priorities clear to the whole organization. This drives the way work is organized and accepted by the teams, and the team members have already shown that they now can refuse to take work of lower priority, instead of ending up in a thrashing mode of constantly changing priorities. Otherwise Pamp has not directly affected the way teams work, and they can still use Scrum as defined in the company.

The portfolio follow-up function and the Paf.com backlog provide visibility into what is currently ongoing at Paf.com. The process enforces a clearer and more specific definition of project goals and business value. This helps managers understand what is going on in other departments and the business value of the work. New opportunities can still be utilized fast enough. It takes at the longest 5 weeks to get a new project accepted into teams' sprint planning. Even though the process gives room for expediting, there is no record of terminating a sprint due to demands from a higher-ranked project. Making things more visible has also helped with this, because now managers can negotiate with each other in preparing a business case for a new opportunity and they can make an informed decision based on the facts of all the ongoing projects. In this way, a less important project can be put on hold and a new project of higher business value can replace it in a controlled manner.

The structured process for proposing new projects has also dramatically reduced the amount of "Do it, it doesn't matter what it costs and how much time it takes, because I want it!" projects in the portfolio. These "pet projects" could be seen 2 years ago (if one knew where to look for them), but today none exist as far as we know.

Still, not everything is run as a project and thus is not part of the Paf.com backlog or the portfolio management process. For this, there is a clearly defined and accepted, company-wide prioritization based on work types. All production problems, i.e. problems in the customer-facing, live products, get first priority. Second priority is on all work that needs to be done to be able to make the next release. During 2009 10 production releases were made and more than 20 patches, meaning a response to production issues, were deployed into production.

These two work types are allowed to thrash work of the third priority, work based on the priority order of the Paf.com backlog. If time still remains, other work can be performed. In general, one team is working on one project backlog with additional stories or tasks from company-wide improvement work, such as upgrading or migrating infrastructure or improving the logging functionality of the component they are responsible of. If somebody tries to circumvent this priority order of work, they are stopped or the issue is escalated. However, as noticed in [2], this could be a source of problems because the availability of resources at any moment is unknown, and thus the situation needs to be monitored in the future.

To summarize the initial positive results, it seems that the problems that were tackled by introducing Pamp and portfolio management have been solved quite satisfactorily. However, it is still too early to conclusively say how much time-to-market has improved. The second survey results show improvement in the areas identified as the biggest challenges. Also, the areas of over commitment and multitasking show improvement, which is in line with the goals for introducing portfolio management.

4.5. Improvement needs and ideas

While the initial feeling and reaction has been mostly positive, some improvement needs and ideas have already surfaced. According to the second author, one of the most challenging things related to the introduction of portfolio management and Pamp has been to make people understand the separation of the value-adding work in a project from the planning of the project. The conceptual mix of having an overall agile framework for development work together with more "traditional" project management practices and

portfolio management for planning the work can admittedly be confusing. The planning is needed to help prioritize and coordinate the work, especially when several teams work on the same project. From an agile standpoint one could argue that the “right” way to go would have been to create true feature teams. But the agile ideal is far from easy to gain in a multi-team environment. Portfolio management has been shown both in theory and practice to be one good alternative for helping in scaling agile. Therefore it was a valid choice to be tried out in our opinion.

Developing a training package for project management in the context of Pamp and the Paf.com backlog process could help mitigate the lack of understanding. Furthermore, a forum for PO’s is being started, where they can interact, learn from each other, and solve problems together. This forum will later be officially added to the Paf.com backlog process after the test-run has been completed and the retrospective has found it useful in practice.

There is still work to be done to get good enough progress data out of evaluation projects, or spikes in XP-terminology [1]. Currently the data is a subjective opinion communicated in progress reports distributed to the whole organization by e-mail. Demos are also used, but mainly for the stakeholders of the evaluation project, which makes it hard to see the impact to other projects and overall resourcing when making portfolio-level decisions.

5. Contribution and future work

The contribution of this paper is twofold. First, we describe how Paf.com has introduced portfolio management to help them scale agile software development. We provide a rich description of the introduction of portfolio management and initial results. While we cannot generalize the results based on only one case company, the results may still inspire other companies into considering portfolio management as an option to help in scaling agile. The descriptive case study adds to the body of knowledge of portfolio management

Second, the condensed section on related work, especially Section 2.2, provides a good starting point of references for anyone interested in adopting or improving portfolio management in their agile organization. While the list of references is by no means exhaustive, we feel that it is a good sampling of the current trends in portfolio management, especially regarding agile ways of working.

Since this work only reports on the initial results and experiences, we intend to continue monitoring and improving the portfolio management practices and process. Part of this work is conducting the interview

for the second round of the survey and analyzing and disseminating the results.

We are also planning to write a case study on the biggest project in Paf.com history, ending in the fall 2010. That would also help fill the gap left in this paper as to describing in more detail the whole development process, including the project and portfolio management processes presented in this paper.

While specific techniques for portfolio management, such as valuation of projects/epics, were left out of scope of this paper, there is still a need to study which of these exist or could be applicable for portfolio management in a lean or agile context. This could include, for example, studies of IT investment analysis, real options, and other IT governance models. Also, other competing models for scaling agile should be reviewed and a comparative analysis of different models would provide value in helping companies choose suitable scaling strategies and practices. In general, more case studies and industry experience reports on portfolio management are needed to help industrial actors find better ways of working.

6. References

- [1] K. Beck, *EXtreme Programming eXplained*. Addison-Wesley, 2000.
- [2] B. S. Blichfeldt and P. Eskerod, "Project portfolio management - There's more to it than what management enacts," *International Journal of Project Management*, vol. 26, pp. 357-365, 2008.
- [3] A. Cockburn, *Agile Software Development*. Pearson Education, 2002.
- [4] R. G. Cooper, S. J. Edgett and E. J. Kleinschmidt, *Portfolio Management for New Products*. Perseus Books, 2001.
- [5] R. G. Cooper, S. J. Edgett and E. J. Kleinschmidt, "Portfolio management: Fundamental to new product success," in *The PDMA Toolbook for New Product Development*, P. Belliveau, A. Griffin and S. Somermeyer, Eds. New York: John Wiley & Sons, Inc, 2002, pp. 331-364.
- [6] R. G. Cooper, *Product Leadership: Creating and Launching Superior New Products*. Perseus Books, 1998.
- [7] R. G. Cooper, "Perspective: Third-Generation New Product Processes," *JPIM*, vol. 11, pp. 3-14, January, 1994.
- [8] V. T. Heikkilä, K. Rautiainen and S. Jansen, "A revelatory case study on scaling agile release planning," in *Proceedings of the 36th Euromicro/SEAA*, Lille, France, 2010, pp. 289-296.

- [9] M. Kalliney, "Transitioning from agile development to enterprise product management agility," in *Proceedings of AGILE 2009*, Nashville, USA, 2009.
- [10] D. Karlström and P. Runeson, "Integrating agile software development into stage-gate managed product development," *Empirical Software Engineering*, vol. 11, pp. 203-225, 2006.
- [11] P. Kettunen, "Adopting key lessons from agile manufacturing to agile software product development—A comparative study," *Technovation*, vol. 29, pp. 408-422, 7, 2009.
- [12] J. Krebs, *Agile Portfolio Management*. Redmond, WA, USA: Microsoft Press, 2009.
- [13] M. Laanti, "Implementing program model with agile principles in a large software development organization," in 2008, pp. 1383-1391.
- [14] G. Larman and B. Vodde, *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*. Boston, MA, USA: Addison-Wesley, 2010.
- [15] G. Larman and B. Vodde, *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Boston, MA, USA: Addison-Wesley, 2009.
- [16] D. Leffingwell, *Scaling Software Agility: Best Practices for Large Enterprises*. USA: Addison-Wesley, 2007.
- [17] M. McGrath, "Setting the PACE in product development," 1996.
- [18] I. Nonaka and H. Takeuchi, "The New New Product Development Game," *Harv. Bus. Rev.*, vol. 64, pp. 137-146, 1986.
- [19] P. O'Connor, "Spiral-up implementation of NPD portfolio and pipeline management," in *The PDMA Toolbook for New Product Development*, P. Belliveau, A. Griffin and S. Somermeyer, Eds. New York: John Wiley & Sons, Inc., 2004, pp. 461-492.
- [20] R. Pichler, *Agile Product Management with Scrum: Creating Products that Customers Love*. Boston, MA, USA: Addison-Wesley, 2010.
- [21] PMI, *A Guide to the Project Management Body of Knowledge : PMBOK Guide*. Newtown Square, PA, USA: PMI, 2008.
- [22] M. Poppendieck and T. Poppendieck, *Leading Lean Software Development: Results are Not the Point*. Boston, MA, USA: Addison-Wesley, 2010.
- [23] M. Poppendieck and T. Poppendieck, *Implementing Lean Software Development: From Concept to Cash*. Boston, MA, USA: Addison-Wesley, 2007.
- [24] J. Rothman, *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects*. USA: Pragmatic Bookshelf, 2009.
- [25] A. Shalloway, G. Beaver and J. R. Trott, *Lean-Agile Software Development: Achieving Enterprise Agility*. Boston, MA, USA: Addison-Wesley, 2010.
- [26] C. Steindl, "From agile software development to agile business," in *Proceedings of 31st Euromicro/SEAA*, Porto, Portugal, 2005, pp. 258-265.
- [27] G. I. Susman and R. D. Evered, "An Assessment of the Scientific Merits of Action Research," *ASQ*, vol. 23, pp. 582-603, December, 1978.
- [28] J. Sutherland, G. Schoonheim and M. Rijk, "Fully distributed scrum: Replicating local productivity and quality with offshore teams," in *Proceedings of the 42nd Hawaii International Conference on System Sciences*, Hawaii, USA, 2009.
- [29] J. Sutherland, A. Viktorov, J. Blount and N. Puntikov, "Distributed scrum: Agile project management with outsourced development teams," in *Proceedings of the 40th Hawaii International Conference on System Sciences*, Hawaii, USA, 2007.
- [30] A. Tengshe and S. Noble, "Establishing the agile PMO: Managing variability across projects and portfolios," in *Proceedings of AGILE 2007*, Washington D.C., USA, 2007.
- [31] J. C. Thomas and S. W. Baker, "Establishing an agile portfolio to align IT investments with business needs," in *Proceedings of Agile 2008*, Toronto, Ontario, Canada, 2008, pp. 252-258.
- [32] J. Vähäniitty, K. Rautiainen and C. Lassenius, "Small software organisations need explicit project portfolio management," *IBM J. RES. & DEV.*, vol. 54, pp. 1-12, March/April, 2010.
- [33] J. Vähäniitty and K. Rautiainen, "Towards an approach for managing the development portfolio in small product-oriented software companies," in *Proceedings of the 38th Hawaii International Conference on System Sciences*, Hawaii, USA, 2005.
- [34] S. C. Wheelwright and K. B. Clark, *Revolutionizing Product Development*. New York: The Free Press, 1992.
- [35] R. K. Yin, *Case Study Research: Design and Methods*. London: SAGE Publications, 1994.