Aalto University
School of Science
and Technology

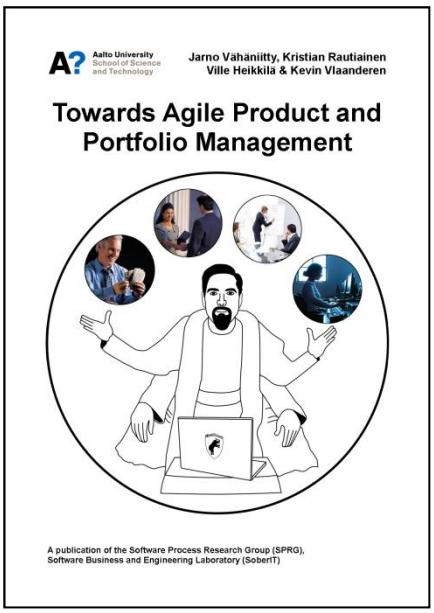
ATMAN deliverables

Aalto University
School of Science
and Technology

BOOKS

Towards Agile Product and Portfolio Management

Download PDF



Success in today's software industry requires integrating long-term product and business planning with technology development, juggling the scarce development resources so that those activities that from a business perspective are the most important get attended to, as well as combining flexibility and control provided by modern, agile approaches to software development, such as Scrum. However, this is not easy, and to be compatible with agile software development, the enterprise level processes of product and portfolio management have to be understood in a new way.

This book summarizes the findings of ATMAN (Approach and Tool support for development portfolio MANagement), a three-year research project conducted by members of the Software Process Research Group (SPRG) of the Software Business and Engineering Institute (SoberIT) at the School of Science and Technology of the Aalto University, Finland. In it we provide a synthesis of guidelines from those relatively few authors out there that deal with the reconciliation of long-term product and business planning, portfolio management and agile software development. Combining these with our own findings from a decade of research collaboration with the top Finnish Software Companies, we hope you find this book a part of the solution.

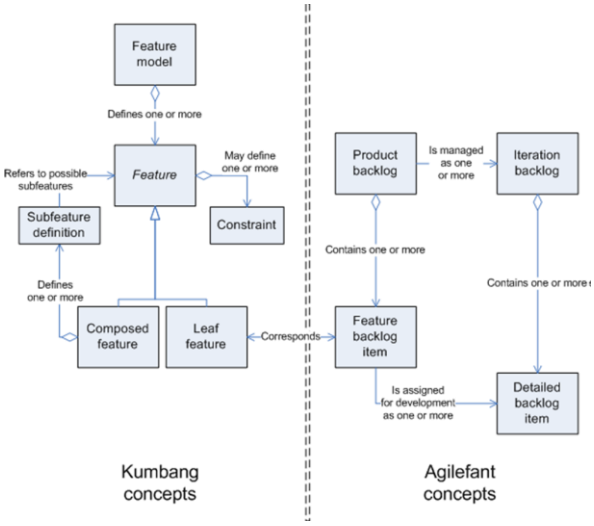
Eds. Heikkilä, V., Rautiainen, K., Vähäniitty, J. 2010, Espoo, Finland: Aalto University.

INTERNATIONAL RESEARCH PUBLICATIONS

Integrating Product Family Modeling with Development Management in Agile Methods

Download PDF

View at publisher



Software product families and agile development have emerged as a popular means in software engineering. In this position paper, we discuss how development management in agile methods can be integrated with software product family structure modeling. The integration aims towards improving software product family development governance by providing technology in terms of concepts and even automated tool support for planning, monitoring and controlling the development work for different stakeholders. For example, integration provides support for prioritization of development tasks and enables monitoring the development status of products in a software product family, for example. The feasibility of integration is shown by combining Kumbang and Agilefant conceptualizations and respective prototype tools.

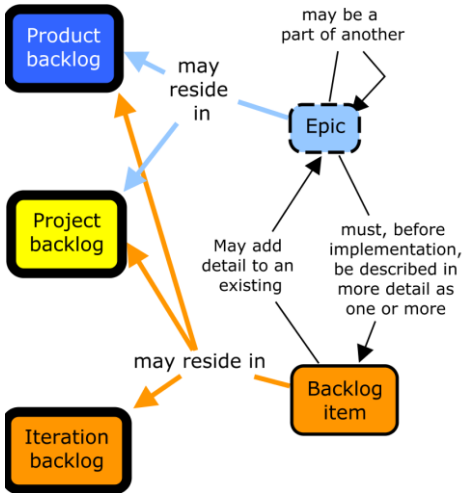
Raatikainen, M., Rautiainen, K., Myllärniemi, V., Männistö, T. 2008, in Proceedings of the 1st international workshop on Software development governance (SDG'08), pp. 17-20.

Towards a Conceptual Framework and Tool Support for Linking Long-term Product and Business Planning with Agile Software Development

Download PDF

View at publisher

Presentation slides



For a software company it is essential to understand how to link business management and software development decision making. Agile methods adhere to the viewpoint of individual development projects, leaving business concerns such as long term product and release planning and multi-project management mostly unaddressed. With poorly governed fast paced development, the big picture of the ongoing work and its link to the company's overall business goals and strategy may become unclear. The difficulties in linking business and development are also reflected in current project management/issue tracking tool support. In this paper we present a conceptual

framework of the links between long-term business, product and release planning and agile software development. The framework aims to provide a common language through which the big picture of software development—including needed roles, responsibilities and decision structures—can be analyzed, communicated and discussed. We also present Agilefant, a proof-of-concept tool based on the framework.

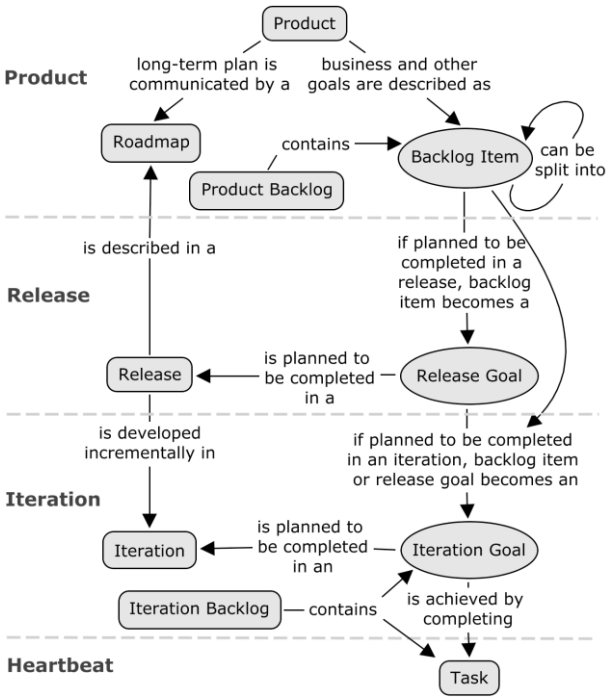
Vähäniitty, J., Rautiainen, K. 2008, in Proceedings of the 1st International Workshop on Software Development Governance (SDG '08), pp. 25-28.

Software Development Governance Challenges of a Middle-Sized Company in Agile Transition

Download PDF

View at publisher

Presentation slides



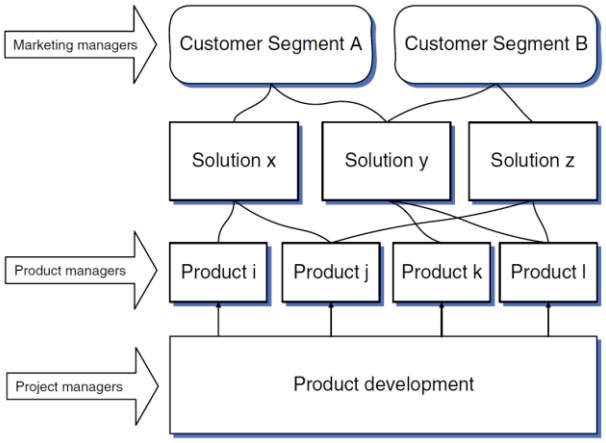
We studied how a middle-sized Finnish company employing agile methods governs its software product development. Through observations and interviews we followed the trace from strategic plans in the form of roadmaps to various backlogs and all the way to daily work. The governance roles, responsibilities and deliverables seemed to be in place on different organizational levels. However, closer inspection revealed challenges in the practical implementation. There were too many roles and hierarchy levels with information consistency problems in between. Prioritization of the high-level goals was unclear and made it difficult to plan and organize development work based on business value. The trace from high-level goals to more detailed plans was easily corrupted due to poor planning practices. Progress monitoring of daily work was poorly done and not linked to high-level plans. Consequently, the required feedback loops were inadequate, making it impossible for management to take corrective actions in time.

Lehto, I., Rautiainen, K. 2009, in Proceedings of the 2009 ICSE Workshop on Software Development Governance (SDG '09), pp. 36-39.

Linking business and requirements engineering: is solution planning a missing activity in software product companies?

Download PDF

View at publisher



A strong link between strategy and product development is important, since companies need to select requirements for forthcoming releases. However, in practice, connecting requirements engineering (RE) and business planning is far from trivial. This paper describes the lessons learned from four software product companies that have recognized the need for more business-oriented long-term planning. The study was conducted using the action research approach. We identified five practices that seem to strengthen the link between business decisions and RE. These are (1) explicating the planning levels and time horizons; (2) separating the planning of products' business goals from R&D resource allocation; (3) planning openendedly with a pre-defined rhythm; (4) emphasizing whole product thinking; and (5) making solution planning visible. To support whole-product thinking and solution planning, we suggest that companies create solution concepts. The purpose of the solution concept is to provide a big picture of the solution and guide RE activities.

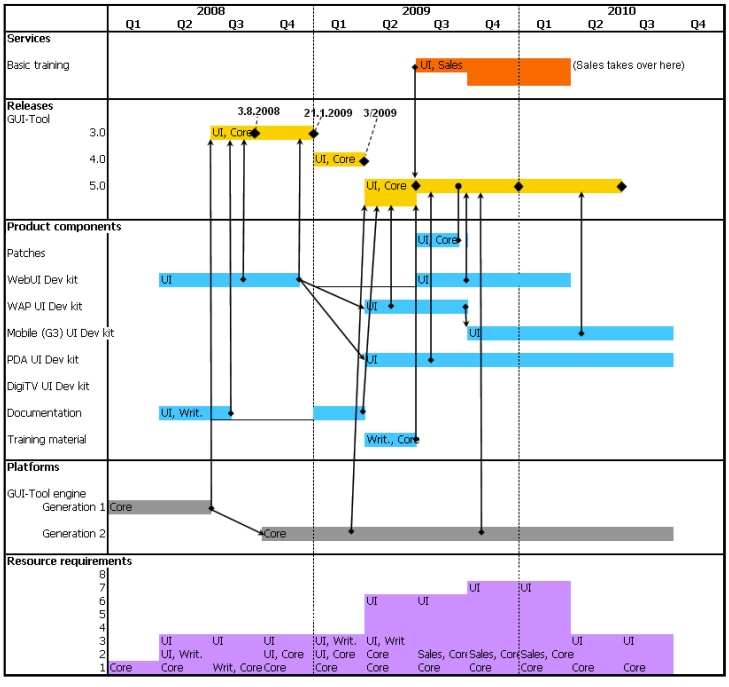
Lehtola, L., Kauppinen, M., Vähäniitty, J., Komssi, M. 2007, Requirement Engineerin, vol. 14, no. 2, pp. 113-128.

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

Download PDF

View at publisher

Presentation slides



Success in the software product business requires timely release of new products and upgrades with proper quality and the right features. For this, a systematic approach for managing the contents, timing and roles of future product releases as well as the product architecture is needed. In practice, such an approach is often missing, especially in small companies, due to inexperience, unclear priorities, time-to market pressures, or the lack of suitable process infra-structure. In this paper, we present

a model to visualize product roadmaps developed together with three small software companies and experiences from its use. The model depicts release and development schedules, the composition of individual releases, services that require attention from the developers, changes to the underlying technology and the planned resource usage. We also present lessons learned from the case companies and outline directions for future research.

Vähäniitty, J., Lassenius, C., Rautiainen, K., Pekkanen, P. 2009, in Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA '09), pp. 300-305.

Rigorous Support for Flexible Planning of Product Releases — A Stakeholder-Centric Approach and its Initial Evaluation

Download PDF

View at publisher

Presentation slides

ID	O1	O2	O3	O4	O5	M
F01	2	2	3	2	2	3
F02	3	3	2	2	3	1
F03	2	2	1	3	1	2
F04	3	1	2	3	3	2
F05	1	3	2	1	1	3
F06	1	1	1	1	1	1
F07	1	1	1	1	2	1
F08	1	1	3	1	2	2
F09	1	1	1	1	1	1
F10	1	1	1	1	1	1
Opt. (%)	100,0	98,9	98,6	97,1	96,8	93,4

This paper addresses the problem of product release planning in iterative product development. We propose a method which combines decision, process, and tool support. The method, which is called SCERP, facilitates the active involvement of stakeholders in the different stages of the planning process. SCERP is flexible in the number of stakeholders involved, in the number of releases, in the number and definition of planning criteria, and in the selection of the best plan out of a set of optimized alternatives. A proof-of-concept of the method is given by a case study of release planning for a tool called Agilefant, which is developed with a process partially based on Scrum. The benefits of the method as demonstrated by the case study are: (i) better decisions by the product manager by relying on more objective information, (ii) more transparency of release decisions, and (iii) efficient tool support accompanying the whole process.

Heikkilä, V., Jadallah, A., Rautiainen, K., Ruhe, G. 2010, in Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS-43), pp. 1-10.

Small software organizations need explicit project portfolio management

Download PDF

View at publisher

The concept of managing new product development projects as an explicit portfolio originates from the context of large organizations. However, the question as to whether explicit portfolio management is relevant for small organizations is rarely discussed. We conducted a qualitative multiple-case study of six small organizations (with 15–40 people) that developed software and provided related services. Five of the organizations did not practice explicit portfolio management. They also seemed to suffer from problems that, in the literature, are considered symptomatic of inadequate portfolio management, such as having too many simultaneous projects, over-commitment in terms of workload, and ineffective executive decision making. In one of the studied organizations, the management personnel had recognized the need for explicit portfolio management and introduced portfolio management practices such as regular reviews of the project

portfolio, appointing specific people for resolving cross-project conflicts, and limiting the number of concurrent projects to which a person can be assigned. The personnel we interviewed perceived clear improvements with respect to various challenges since the introduction of these practices. Our preliminary study suggests that explicit portfolio management is relevant for small software organizations, at least in cases in which the development personnel possess multiple roles and responsibilities and are concurrently performing many different types of activities.

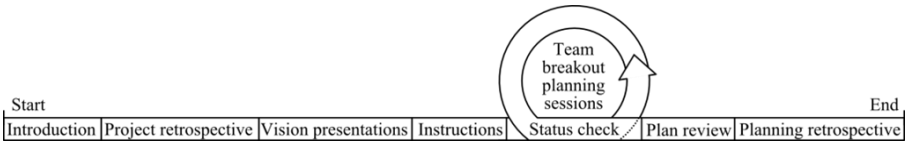
Vähäniitty, J., Rautiainen, K., Lassenius, C. 2010, IBM Journal of Research and Development, vol. 54, no. 2, pp. 1:1-1:12.

A Revelatory Case Study on Scaling Agile Release Planning

Download PDF

View at publisher

Download presentation

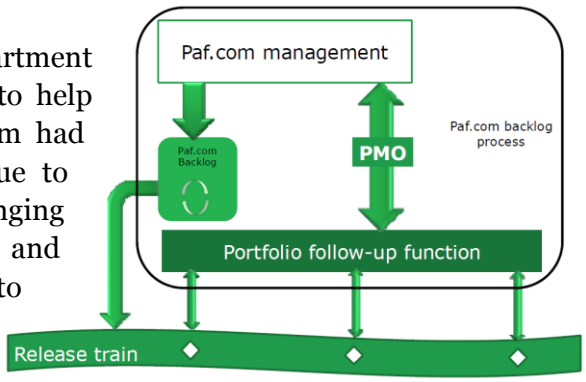


A way to scale up agile release planning to meet the requirements of multi-team agile development is a practice called joint release planning. A software product company piloted the joint release planning method. The aim of the company was to improve coordination of work of multiple agile development teams who develop a large legacy software product. Another aim was to improve communication between product management and development. We conducted a case study to explore how the new release planning method was executed. We gathered data by observing two release planning events, observing event planning meetings, and by conducting surveys. The events were attended by approximately 140 stakeholders, including over 10 development teams, who spent several days in a common space. The participants liked the method and considered it efficient. This revelatory case study provides the first detailed empirical description of this emerging method for multi-team agile release planning.

Heikkilä, V., Rautiainen, K., Jansen, S. 2010, in Proceedings of the 36th Euromicro on Software Engineering and Advanced Applications (SEAA '10), pp. 289-296.

Supporting Scaling Agile with Portfolio Management: Case Paf.com

Download pdf



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.

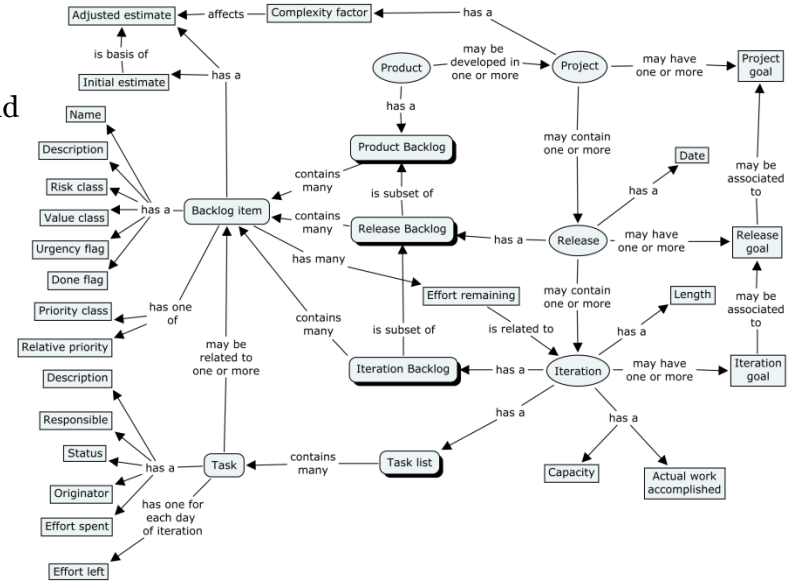
Rautiainen, K., von Schantz, J., Vähäniitty, J. forthcoming 2011, in Proceedings of the 44rd Hawaii International Conference on System Sciences (HICSS-44).

THESES

Tool Support for Development Management in Agile Methods

Download pdf

Presentation slides



The goal of this research was to find out what requirements the most prominent agile methods and the case company have for a development management tool and to determine if an existing tool that sufficiently fills the requirements can be found. Many tools have been developed to facilitate the management of work and requirements in agile software development projects since the mainstream breakthrough of agile software development methods following Agile Manifesto in 2001. However, it is not known whether the tools that are publicly available actually meet the requirements that the agile methods have for managing work and requirements. Pre-existing research on the topic has concentrated on generating a set of requirements and building a new tool which is based on those requirements.

Based on an analysis of popular articles on agile software development, Extreme Programming and Scrum were found to be the most prominent methods. Practitioner guidebooks on each method were selected and then reviewed for requirements. Interviews were performed in the case company and the results were analyzed to identify the requirements. The resulting requirements belonged to two groups: conceptual requirements, which describe what kind of information needs to be saved in the tool, and functional requirements, which describe what kind of functionality the tool must have. The open source tool Agilefant and the commercial tools Mingle, Rally Enterprise Edition and ScrumWorks Pro were selected. Each tool was then separately reviewed against the requirements.

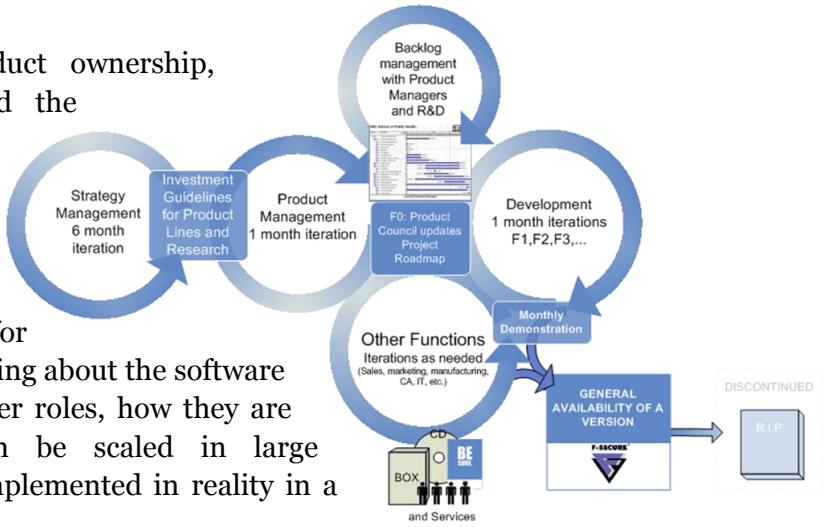
Agilefant had severe deficiencies concerning conceptual requirements in both requirement sets. These conceptual deficiencies also resulted in severe deficiencies in functional requirements. Mingle filled well the conceptual requirements of both sets. It also successfully met the functional requirements from the books. However, Mingle failed to fulfill the important functional requirements from the case company that concerned work-hour recording and reporting. Rally Enterprise Edition adequately fulfilled the conceptual and functional requirements from the book review, but the tool had severe conceptual and functional deficiencies regarding the work-hour recording and reporting requirements from the case company. ScrumWorks Pro had many deficiencies concerning the conceptual and functional requirements from the book review and the case company.

The results of this research show that although tools for agile software development by-the-book exist, real world software development companies have additional requirements which may not be fulfilled by the current tools. In this case, the majority of the additional requirements concerned work-hour recording and reporting. Tool developers should take into account the real world needs of software development organizations when creating tools. In order to enable the development of tools that better match the real world requirements, further research into other agile software development organizations should be conducted to identify what kind of additional requirements the organizations have.

Heikkilä, V. 2008, Helsinki University of Technology, Department of Computer Science and Engineering, master's thesis.

Product Management and Product Owner Role in Large-Scale Agile Software Development

Download pdf



This study focuses on the product ownership, namely product management and the Scrum product owner role, in large-scale commercial software product development environments where agile and lean principles and practices are applied in engineering. The goals for this research are to gain understanding about the software product manager and product owner roles, how they are connected, how these roles can be scaled in large organizations, and how they are implemented in reality in a case study company.

In the theoretical part, the key concepts of product management, related agile methods and lean software development are summarized. This is followed by an overview of current discussion about the role of product owner and scaling Scrum to environments with multiple development teams. The selected viewpoints come from Pragmatic Marketing, Ken Schwaber, Craig Larman and Bas Vodde, and Dean Leffingwell. Based on these views, the study presents a synthesis that seeks the synergies and combines these into such. The case study describes the iterative evolution of the product ownership domain of F-Secure Corporation during the period of agile adaptation, after which the latest situation is compared with the theoretical synthesis.

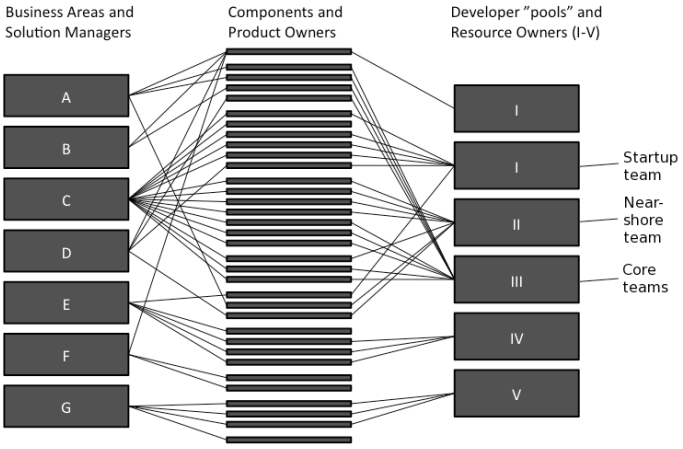
The results of the study show that product manager role(s) operating within the product management domain are not the same as the Scrum product owner role(s), although they are connected. The synthesis presents the relationships between them, and identifies possible means to share the responsibilities between several people in large environments. The case study shows support for this hypothesis although some deviations from the synthesis are identified. These deviations can possibly be an opportunity for improvement in the organization.

Parkkola, M. 2010, Aalto University, School of Science and Technology, Faculty of Information and Natural Sciences, master's thesis.

Aalto University
School of Science
and Technology

Using backlogs for linking long-term product plans and development tasks in agile software development

Download pdf



Agile software development has been used for more than a decade in industry. Scrum, a popular agile method, introduces backlogs as a lightweight approach for managing software requirements and work tasks. The agile methods were originally intended for managing software development of a single team. However, large companies have started to adopt agile methods in large-scale development i.e. in a context where multiple teams and contributing to the same product simultaneously.

In this thesis the applicability of communicating long-term product plans to software developers and communicating progress information back to management, was studied in the context of large-scale agile software development. Existing research of large-scale agile software development is scarce and the role of backlogs has gained even lesser attention. However, some initial suggestions on how to manage backlogs in a multi-team setting were found.

The research problem was studied by conducting a case study in a large Finnish software product company. Research data was collected by conducting interviews and observations. The case company was given constant feedback and the case company reacted to some of the identified challenges. A framework that describes one possible way to link long-term plans with daily tasks, was constructed based on literature. The framework was used to analyze the results and also to collection of the research data.

Challenges were identified within the case company in the organizational structure, planning and progress monitoring practices and also in the tools they used. The case company took a number of corrective actions: An organizational rearrangement cleared off communication barriers, which in turn facilitated better planning practices; The backlogs were also renewed, which stressed the tools and that they had to be also changed.

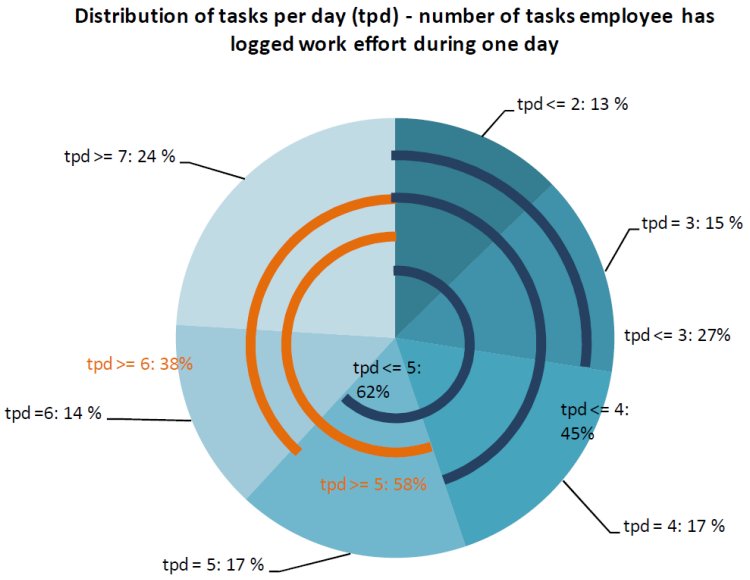
Successful backlog management cannot exist without a well functioning organization, effective work practices and proper tools.

Lehto, I. 2010, Aalto University, School of Science and Technology, Faculty of Electronics, Communications and Automation, master's thesis.

Aalto University
School of Science
and Technology

Designing a Framework for Linking Company Goals with Daily Tasks in a Small Software Company

Download pdf



Action of a company should be based on company's strategic objectives and strategy. Strategy is the plan to fulfill the strategic objectives. Fluidity of resources is important: resources should be easily allocated to strategically important areas. Today, strategy work in companies is more dynamic and practical than it used to be earlier. Also a small company can take a major advantage of the practical strategy tools and models.

Agile software development methods have made it more effective to manage the development work through stories selected into iterations and broken into tasks for development to be completed. This model working effectively to provide a method for effective resource allocation.

In this thesis, the possibility to improve a company's functions and management by linking the strategic, company-level goals with day-to-day tasks is studied. The target of the study was a small software company, which employs less than 50 employees and is using agile software development methods. The business and management models of the case company were studied profoundly. The model was perceived to be functioning well and serving the purposes of the company. The company uses a software tool called Agilefant.org for managing its software development projects. Agilefant is very suitable for operational management purposes. All the work efforts (i.e. also other than software development) are logged into Agilefant.

The activity logged into Agilefant was analyzed for a 13 month period. The analysis of the data showed that the personnel of the company have notably too many projects in-process at the same time. The company suffers from the loss of effectiveness due to task switching. The most important reason for too many projects running at the same time was noted to be the inability to plan future activated. The company did not have any functional method or tool to do the planning and prioritization of future goals and therefore it did not have a functional model to manage future work.

A framework was designed to link future goals with day-to-day activities in the studied company.

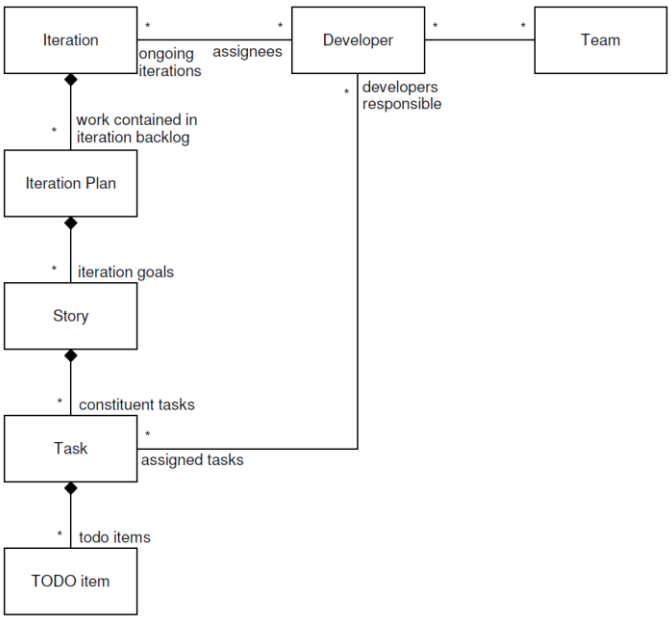
Norja, T. 2010, Aalto University, School of Science and Technology, Faculty of Information and Natural Sciences, master's thesis.

Aalto University
School of Science
and Technology

Tool Support for Managing Daily Work in Agile Software Development

Download PDF

This thesis examines the features of a software tool that is needed for managing efficiently the daily work in agile software development methods, using the constructive research approach. Although agile software development is becoming more and more popular, agile software development management tools are lacking necessary capabilities for managing daily work in these methods. A part of current agile software development guidebooks are critical of existing software tools, and view them as inappropriate and inflexible for daily work management, compared to index cards or spreadsheet programs. Other sources emphasize that information systems are essential in some working environments.



The most popular agile software development methods, Scrum and Extreme Programming are examined. In the theoretical part the daily work practices of these methods and recommended tools are identified from the practitioner guidebooks. Based on the results of literature review, the support for identified daily work management practices in the open source Agilefant tool is reviewed prior to enhancing the support.

The requirements of case companies for daily work management were elicited in the user story format using the requirements workshop method. Acquired user stories were grouped into consistent features: these features were prioritized by the representatives in case companies using modified 100-dollar method. The results of prioritization were used to select a top priority features for deeper analysis and possible implementation.

10 practices and 5 different tools for daily work management were identified in the guidebooks. According to analysis, the support for most practices was insufficient in Agilefant. The workshop resulted in 26 different features that were not present in Agilefant; the “team view” was considered the most important in the prioritization. 14 most important features, representing 80 % of the votes cast are analyzed further: the analysis evaluates the candidate features based on the identified practices; a design for their implementation in the tool is proposed. Of these features, “consolidated story and task list”, “work queue”, “task splitting” and “strategy-to-action (bottom-up)” were implemented in Agilefant.

The results of this research can be utilized for example as a roadmap when creating a new software tool for agile software development management, but also to review the support for essential daily work management practices in existing tools.

Haapala, A. 2010, University of Oulu, Faculty of technology, Department of Industrial Engineering and Management, master’s thesis.

OTHER PUBLICATIONS

Automatic Task Update in Agilefant

Download PDF

The past few years have been a revolution to the mainstream of software development when even the largest IT companies have realized the benefits of agile development. Responsibility and code ownership has been handed back to the developers and teams from oppressive management and middle-layer of the companies' organizational structure. Software is developed in short iterations instead of long development steps, and the customer representatives have become a vital part of successful projects.

The purpose of this study is to define different workflow patterns in individual developer's software development practices, and to find the most suitable combination of tools for the development work and automatic feature verification. This work first clarifies the roles and expected behaviors for team members, and then identifies the automatic and manual steps in software development. This work presents four different workflow models for an individualdeveloper, after which the models are discussed and evaluated.

Klementti, M. 2010, March 23.

Työt hallintaan ilmaisella ohjelmalla

Download PDF

Aallon SoberIT tarttui univeraaliin ogelmaan tehdä liikaa yhtä aikaa.

Yritykset ja vapaaehtoiset koodaajat ovat Teknillisen korkeakoulun Ohjelmistoliiketoiminnan ja –tuotannon instituutti SoberITL:n tutkimusprojektissa kehittäneet työkalun nopeatempoisen ohjelmistokehityksen hallintaan.

Agilefant on avoimen lähdekoodin työkalu ohjelmistokehityksen hallintaan. Sen avulla voi seurata projektien ja tuotteiden kehittymistä ja käytettävissä olevia resurrseja. Agilefant huomio niin yksittäiset työtehtävät, välitavoitteet kuin taustalla vaikuttavat suuremman strategisen päämäärän. Tarkoitus on auttaa tekijöitä pysymään paitsi aikataulussa, myös oikeassa suunnassa.

Repo, H., 2009, Kauppalehti, September 19, Helsinki, Finland: Kauppalehti Oyj/Alma Media Oyj, pp. 26.

A short article presenting the usage and concepts of Agilefant 2

Download PDF

Agilefant is an open source backlog management tool enhanced with hierarchical requirements handling capabilities. Agilefant provides three different planning levels for backlog management: product, project and iteration backlogs. Besides traditional flat-list backlogs, requirements can be expressed in tree form, in order to maintain traceability and transparency.

In addition to backlog and requirement management, Agilefant contains different composition

views to the items in different backlogs, which can be useful for small to mid-sized development organizations. These composition views include time tracking features, personal work management and portfolio handling.

Pekkanen, P., Jokelainen, R. 2010, August 26.

PRESENTATIONS

Tekemissalkun hallinnan lyhyt oppimäärä

Presentation slides

Vähäniitty, J. 2008, TKK Software processes course (T-76.5631), April 17, Espoo, Finland

Agilefant: Open Source Tool Support for Managing an Agile Development Portfolio

Presentation slides

Vähäniitty, J. 2009, Scandinavian-Agile Conference 2009 (Scan-Agile 2009), October 15, Helsinki, Finland

ATMAN Health Barometer summary

Presentation slides

Heikkilä, V., Rautiainen, K. 2010, ATMAN Health Barometer summary seminar, September 14, Espoo, Finland

Agile Product and Portfolio Management

Presentation slides

Vähäniitty, J., Norja, T., Heikkilä, V. 2010, ATMAN Expert seminar, November 9, Espoo, Finland

SOFTWARE

Agilefant 2.0

Download installation package

Health Barometer software

Download installation package