

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

Jarno Vähäniitty

Helsinki University of Technology  
Software Business and Engineering Institute  
P.O.Box 9210, FI-02150 TKK  
+358 50 386 2777

firstname.lastname@tkk.fi

Kristian Rautiainen

Helsinki University of Technology  
Software Business and Engineering Institute  
P.O.Box 9210, FI-02150 TKK

firstname.lastname@tkk.fi

## ABSTRACT

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.

## Categories and Subject Descriptors

K.6.3 [Management of Computing and Information Systems]: Software Management—*software development, software process.*

## General Terms

Management

## Keywords

Agile, Product and business planning, Development portfolio, Portfolio management, Software development governance, Tool support, Backlog management

## 1. INTRODUCTION

For a software company, it is essential to understand how to link business management and software development [11] and employ a solid, business-oriented approach in its development decision-making. While initiatives for integrating the perspectives of business and development have started to appear in the software engineering literature (e.g. [3,6,7,11,17]), there is ample room for extending the body of knowledge in this area of research.

Most iterative and incremental development methods—such as agile methods—emphasize building releasable software in short, fixed time periods. Since agile methods originally adhere to the viewpoint of single development teams dedicated to individual projects, multi-project management has mostly been left unaddressed [15]. In practice, poorly governed fast-paced development with multiple teams may lead to fragmentation, with the big picture of the ongoing work and its link to the company's overall strategy becoming unclear. In Scrum, for example, business decision-making and prioritization based on the company's strategy is handled by the product owner role, but proper practical guidance to what it all entails in multi-team, multi-project environments has only started to emerge [4,12]. We believe there is a need for a common language through which the big picture of software development—including the link between business and development—can be analyzed, communicated and discussed.

This position paper contributes to the understanding of software development governance by presenting our work-in-progress conceptual framework of the possible links between agile software development and long-term business, product and release planning. With the help of the framework people can identify needed roles, responsibilities and decision structures ranging from top-management business decisions to reprioritizing the developers' daily work. We also introduce Agilefant, an open source proof-of-concept support tool based on the presented conceptual framework.

In section 2 we describe our framework and the current status of the related tool support. Section 3 concludes the paper with discussion and future work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SDG '08, May 12, 2008, Leipzig, Germany.

Copyright 2008 ACM 978-1-60558-035-7/08/05...\$5.00.

## 2. FRAMEWORK AND TOOL SUPPORT

Sections 2.1-2.5 describe the current version of our conceptual framework for linking long-term product and business planning with agile software development. The framework is an augmentation of the *Cycles of Control* framework (see e.g. [8-11,13,16]) that uses time pacing as the coordinating mechanism for linking business and software development decision-making. The levels of the conceptual framework (Figure 2) discussed in this paper are: *Business unit*, *Product and service*, *Development portfolio*, *Project*, and *Iteration* management. Section 2.6 describes the current state of tool support we have developed based on the framework. For the sake of simplicity and due to space limitations, only the most relevant concepts and relationships are shown and discussed. Also, as the framework at its current state is intended as a basis for analysis and discussion, it does not attempt to prescribe how the levels should be operationalized.

### 2.1 Backlogs, Epics and Backlog Items

Central to the framework are Product, Project and Iteration level *Backlogs* and their relationships to regular *Backlog items* and *Epic backlog items* (*Epics*, for short, see Figure 1). We have chosen the terms Epic and Backlog item instead of e.g. Requirement to indicate that often not everything that requires the developers' attention (see section 2.4) are in fact software requirements [15]. Adapting from Scrum [12], Products, Projects and Iterations have their own Backlogs, which are lists of prioritized items that need to get done. Backlog items and Epics are anything that needs attention from the developers: new features, change requests, bug fixes, service requests, or something else. Epics differ from regular Backlog items in that 1) they may consist of other Epics, 2) they are too big to be reasonably estimated in terms of effort, and 3) they may not reside in Iteration-level backlogs (see Section 2.3). Example Backlog items could be "The user must be able to change the priority for multiple items simultaneously" and "Set up Agilefant demo instance", whereas example Epics could be "Support for roadmapping" and "Agilefant must be able to handle 1000 simultaneous users". In our framework regular Backlog items are allowed to reside in the Product and Project backlogs—side by side with Epics—to avoid unnecessary hierarchy.

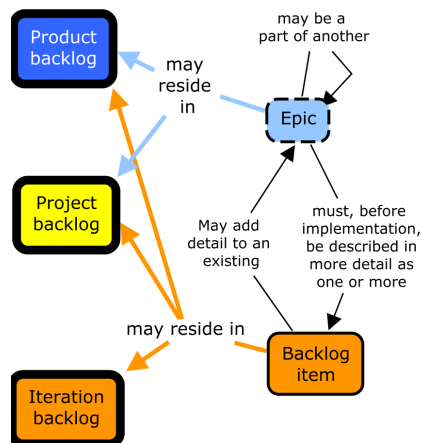


Figure 1 Backlogs, Epics and Backlog items

Sections 2.2-2.5 describe the levels of the framework in more detail by going through the rest of the concepts and relationships shown in Figure 2 below.

### 2.2 Business Unit Level

The Business unit level addresses the management of an individual business unit in terms of how to compete in a particular industry or product/market segment [2]. *Product* is the generic term used for an offering. It may be a piece of software or a service that the company is developing, either to be sold, or for internal use. Products should contribute to a *Vision*. A *Vision* describes the "grand plan" for one or more Products, and is concretized as one or more *Business goals*. In our framework, the overall strategy of a business unit is described in terms of the defined *Visions* and *Business goals*.

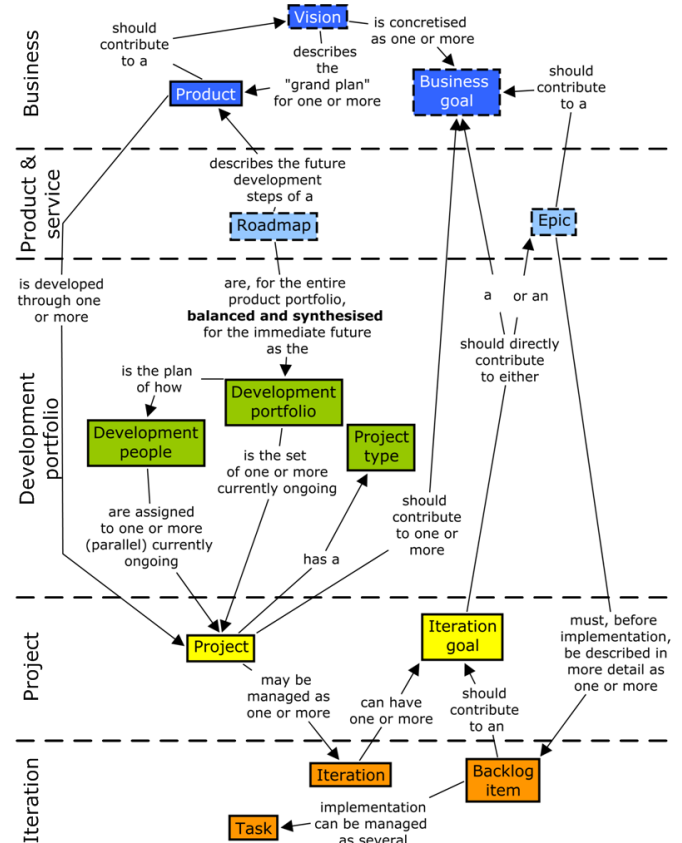


Figure 2 Linking product and business planning with agile development

### 2.3 Product and Service Level

The Product and service level is responsible for product and release planning. Product planning is about creating and updating Epics for a specific product/service proposition to meet the Business goals. The emphasis of planning should be less on constraints such as resourcing, i.e. "If this product had all the resources we wanted, what would be achieved?" In our framework, release planning means describing the future development steps of a product/service through roadmapping [5]. In roadmapping the currently defined Business goals and Epics are mapped to future development Projects.

## 2.4 Development Portfolio Level

Portfolio management is a key governance process for linking business and development. On the Development portfolio level the “hard reality” should step in. In development portfolio management, new projects are evaluated, selected and prioritized, existing projects may be accelerated, de-prioritized or killed, and resources are allocated to and reallocated within active projects based on business priorities and other constraints such as resource or technical dependencies [1,14]. In our framework, the contents of the Roadmaps for different product/service propositions are balanced and synthesized for the immediate future to match the available resources and the most important Business goals as closely as possible. This results in the *Development portfolio*, which consists of (1) the set of currently ongoing Projects that require attention from the *Development people*, i.e., the product development and/or technical personnel available to the business unit [14], and (2) the plan for how the Development people are assigned to these. Development portfolio management is also concerned with the *Project types* (e.g. new releases, customer-specific development, deliveries, maintenance) and governs that resource spending is aligned with the Business unit’s overall strategy. Development portfolio management should also account for possible non-project and other work, which may take up a considerable amount of Development people’s attention.

## 2.5 Project and Iteration Levels

Project management deals with the necessary planning and monitoring of individual Projects. This consists of, e.g., detailed planning of how the Project contributes to the Business goals, and planning the initial goals and high-level content of the *Iterations* of the project [8]. Iteration management aims at developing the product as a series of reasonably stable, working intermediate versions having part of the functionality of the final release to get feedback during development [8]. Those Epics (or parts of them) that are to be implemented in an Iteration must before the start of the Iteration be described in more detail as one or more Iteration Backlog items. *Iteration goals* should be used to provide tangible, Business goal related objectives for the Iteration that go beyond or summarize the individual Backlog items selected for the Iteration. In daily work, the implementation of individual Backlog items can be further detailed as *Tasks*.

## 2.6 Tool Support

Although some solutions for linking business and development seem to be emerging (e.g. Rally<sup>1</sup>), the missing conceptual links are still in practice reflected in project management/issue tracking tool support. For example, the solutions for project management and issue tracking we have seen used in the Finnish software industry either 1) do not support agile software development 2) lack the capabilities to link daily work such as tasks/features/backlog items with long-term product and business objectives, and/or 3) do not support managing the developers’ efforts as an explicit portfolio [15].

Agilefant is a proof-of-concept prototype support tool for backlog and development portfolio management. It is an open

source J2EE web application based on the Spring framework<sup>2</sup>. Its development is coordinated by the first author. Agilefant’s vision is to support iterative and incremental development work, help with business-alignment, e.g., through linking daily work items with business level goals, and support managing the developers’ efforts as an explicit portfolio. Currently Agilefant implements the concepts and related functionality described in sections 2.1-2.5 with the exception of *Vision*, *Business goal*, *Epic*, and *Roadmap*. These have been denoted with a dashed line in Figure 1 and Figure 2.

So far, Agilefant has been actively used by three teams. Two of the teams are student groups of nine on the Software development project course at TKK<sup>3</sup>. One of the teams develops Agilefant under the supervision of the authors, while the other pilots Agilefant as a special assignment. The third team consists of four researchers (two of which are the authors of this paper), who have used Agilefant since September 2007 to manage their work. The research team has a diverse ‘development portfolio’, consisting of software development, research & publishing, industrial collaboration, various teaching activities, and consulting. Currently, in Agilefant’s terms, the members of the research team work concurrently on nine Projects concerning three Products. Three of the Projects are managed using one month iterations.

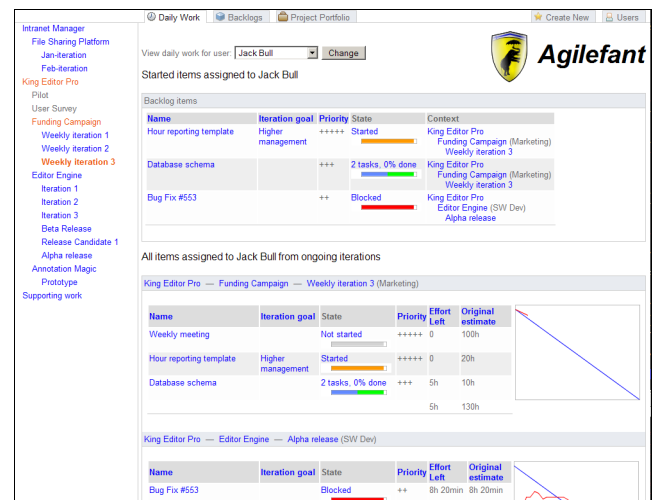


Figure 3 Viewing the Daily Work of a user in Agilefant

We have recently extended Agilefant’s functionality to address Development portfolio management. Agilefant now has a *Project Portfolio* view for prioritizing and resourcing projects. There is also a *Daily Work* view (see Figure 3) to help users figure out what the next job (i.e. backlog item) they should attend to is. The Daily Work view summarizes and presents the backlog items assigned to a single user across all of his projects. The items already started are displayed at the top of the page according to (1) project priority and (2) priority within a project. Below the started items, all of the items in each project assigned to the user are displayed project-wise, from the most important to the least important project. We believe that in today’s typical

<sup>1</sup> See [www.rallydev.com](http://www.rallydev.com)

<sup>2</sup> See [www.agilefant.org](http://www.agilefant.org) for more information and downloads

<sup>3</sup> <http://www.soberit.hut.fi/T-76.4115/>

multiple responsibility and multi-tasking environment [15] this kind of view may help both in selecting the next job that really needs attention as well as in maintaining focus on the work at hand.

### 3. DISCUSSION AND FUTURE WORK

In this position paper, we presented a tentative framework for linking long-term product and business planning with agile software development. We also presented the current state of Agilefant, a proof-of-concept open source support tool based on the framework.

Our initial experiences of using the conceptual framework in companies are encouraging. By discussing the framework and its concepts together with the key personnel in two companies we have been able to identify missing responsibilities, decision-making structures, and poorly defined roles. We have also been able to propose tangible improvement suggestions which the companies have acted on.

Currently, we are working towards improving and validating the framework based both on existing literature, practitioner case studies, and piloting Agilefant in Finnish software companies. In terms of Agilefant, our next major effort will be to implement the Business and Product & Service level concepts and functionality described in Sections 2.2 and 2.3. We are also looking at existing solutions for project/backlog/issue management in order to abstract a core set of 'feature families' needed to link long-term product and business planning with daily work.

### 4. ACKNOWLEDGMENTS

We would like to thank Tekes (the Finnish Funding Agency for Technology and Innovation) and the participating companies (F-Secure, EGET, eCraft, IPSS, Mipro, Napa and Tekla) for funding and participation in the ATMAN research project.

### 5. REFERENCES

- [1] Cooper, R. G. 1998. Product leadership: creating and launching superior new products. Perseus Books
- [2] Hitt, M., Ireland, D., and Hoskisson, R. 1997. Strategic Management: Competitiveness and Globalization: Concepts, 2 edn. West Publishing Company
- [3] Hohmann, L. 2003. Beyond software architecture: creating and sustaining winning solutions. Addison-Wesley Professional.
- [4] Leffingwell, D. 2007. Scaling software agility: best practices for large enterprises. Addison-Wesley Professional
- [5] Lehtola, L., Kauppinen, M., and Vähäniitty, J. 2007. Strengthening the link from business decisions to requirements engineering: Long-term product planning in software product companies. In Proceedings of the 15th IEEE International Requirements Engineering Conference (New Delhi, India, 2007). IEEE, pp. 153-162. DOI=<http://doi.ieeecomputersociety.org/10.1109/RE.2007.59>
- [6] Nocks, J. 2006. Multiple Simultaneous Projects with One eXtreme Programming Team. In Proceedings of the Agile Conference 2006 (Minneapolis, Minnesota, USA, 2006). IEEE Computer Society, pp. 170-174. DOI=[10.1109/AGILE.2006.39](http://doi.ieeecomputersociety.org/10.1109/AGILE.2006.39)
- [7] Qumer, A. 2006. Defining an Integrated Agile Governance for Large Agile Software Development Environments. In Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming (Como, Italy, 2007). Springer, pp. 157-160. DOI=<http://www.springerlink.com/content/p40365m2n327vp24/>
- [8] Rautiainen, K. 2004. Cycles of Control: A Temporal Pacing Framework for Software Product Development Management. Licentiate Thesis. Helsinki University of Technology. Available online at <http://www.soberit.hut.fi/kqr/>
- [9] Rautiainen, K., Lassenius, C., and Sulonen, R. 2002. 4CC: A Framework for Managing Software Product Development. Engineering Management Journal. 14, 2 (2002), pp. 27-32.
- [10] Rautiainen, Kristian et al. 2004. Pacing Software Product Development: A Framework and Practical Implementation Guidelines (2<sup>nd</sup> printing, electronic edition). Helsinki University of Technology, Software Business and Engineering Institute, Technical reports 3.
- [11] Rautiainen, K., Vuornos, L., and Lassenius, C. 2003. An Experience in Integrating Strategic Product Planning and Agile Software Development Practices. In Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering (Rome, Italy, 2003), pp. 28-37
- [12] Schwaber, K. 2007. The Enterprise and Scrum. Microsoft Press.
- [13] Vähäniitty, J. 2005. A Tentative Framework for Connecting Long-Term Business and Product Planning with Iterative & Incremental Software Product Development. In Proceedings of the 7th International Workshop on Economic-Driven Software Engineering Research (EDSER-7) (St. Louis, USA, 2005). IEEE, pp. 1-4. DOI=<http://doi.acm.org/10.1145/1083091.1083097>
- [14] Vähäniitty, J. 2006. Do small software companies need portfolio management, too? In Proceedings of the 13th International Product Development Management Conference (Milan, Italy, 2006). EIASM, pp. 1471-1486. Available online at <http://www.soberit.hut.fi/jvahanii/>
- [15] Vähäniitty, J. and Rautiainen, K. 2005. Towards an Approach for Development Portfolio Management in Small Product-Oriented Software Companies. In Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS-38) (Big Island, Hawaii, USA). IEEE, DOI=[10.1109/HICSS.2005.636](http://doi.ieeecomputersociety.org/10.1109/HICSS.2005.636)
- [16] Vanhanen, J., Itkonen, J., and Sulonen, P. 2003. Improving the Interface Between Business and Product Development Using Agile Practices and the Cycles of Control Framework (Salt Lake City, Utah, USA). Agile Development Conference 2003, pp. 71-80, <http://portal.acm.org/citation.cfm?id=942789.942817>
- [17] Wallin, C., Ekdahl, F., and Larsson, S. 2002. Integrating business and software development models. IEEE Software. 6/2002 (2002), pp. 28-33.