Integrating Agile Modeling with Participative Enterprise Modeling

Janis Stirna[1], Marite Kirikova[2]

[1] Jönköping University, PO Box 1026, SE-551 11, Jönköping, Sweden
janis.stirna@jth.hj.se

[2] Riga Technical University, 1 Kalku Street, Riga, LV 1658, Latvia
marite.kirikova@cs.rtu.lv

Abstract. Agile Modeling (AM) provides a set of best practices of "light-weight" modeling to support the modeling process on a macro level within the agile development teams. At the core of AM is close collaboration with stakeholders which is similar to participative Enterprise Modeling (EM). Hence, the objective of this paper is to analyze the potential of using EM in agile development projects to address some of the existing challenges of agile projects. We analyze the objectives and compatibility of artifacts of AM and EM, compatibility of Agile Model Driven Development and the EM process, as well as the specifics the EM process and the tool support.

Keywords. Enterprise modeling, agile modeling, agile development

1 Introduction

The Information System (IS) development community has been trying out and adopting various agile development approaches such as, e.g., eXtreme Programming (XP) [1], SCRUM [2], and Dynamic Systems Development Method (DSDM) [3]. One of the strengths of the agile development approaches is their flexibility and ability of dealing with change efficiently. Agile approaches typically do not prescribe which methods, languages, and tools to use. Instead, the main emphasis is on choosing the simplest, most effective and, therefore, the most cost effective ones.

To support the modeling process on a macro level within the agile development teams Agile Modeling (AM) [4] was developed. AM provides a set of best practices of "light-weight" modeling and suggests active stakeholder involvement. This is similar to Participative Enterprise Modeling [5].

However, gathering requirements in agile methods is targeted exclusively to software development needs [6, 7, 8]. The relationship between knowledge of enterprise stakeholders and software artifacts is tacit and contributes only to the software development process, not to the enterprise knowledge development on a larger scale. This phenomenon does not permit to utilize all possible benefits of requirement gathering. Nowadays agility is needed not only in software development

but in all aspects of organization's performance. Hence, a method to transparently relate software development to other organizational processes is needed.

Enterprise Modeling (EM) has proven to be a practicable instrument for creating an integrated and negotiated model describing different aspects of an enterprise. An Enterprise Model comprises a number of related "sub-models", each focusing on a particular aspect of an organization, e.g. processes, business rules, concepts/information/data, goals, actors, as well as requirements.

In this paper we analyze the potential of using EM in agile development projects. The research approach is conceptual and argumentative based on findings of a number of qualitative research studies [9, 10, 11, 12, 13].

EM's overall approach to development is similar to AM because models are developed in a series of facilitated modeling seminars with a group of stakeholders. Such a participative way of working improves quality, consensus, acceptance and agreement on the business decisions made and organizational designs produced during the seminar. EM developers have also suggested that EM is applicable for a variety of purposes, e.g. business process standardization, reengineering, strategy planning, knowledge sharing, enterprise integration, as well as IS development [11, 14], (c.f. [5, 15, 16, 17, 18, 19, 20, 21] for examples of EM approaches).

A combination of EM and Agile approaches thus suggests a new paradigm in agile projects where relevant knowledge is derived from the overall organizational development knowledge, captured in an Enterprise Model. Such an approach utilizes all knowledge created in the requirements gathering process and ensures the transparency of relationships between the IS development process and other organizational development processes.

The rest of the paper is structured as follows. Section 2 provides background to EM. Section 3 presents a brief background to agile development approaches while section 4 outlines the current issues and challenges of agile projects. Section 5 discusses the potential of the use of EM in agile projects in terms of modeling objectives, artifacts, process and tool support. Section 6 briefly presents concluding remarks.

2 Background to Enterprise Modeling

EM is a method for developing, acquiring, and communicating early, enterprise knowledge, such as strategies, goals, or requirements, by a structured, iterative, working and modeling approach [5]. The Enterprise Model consists of a set of structured, goal/problem driven models to be used for structuring and representing organizational knowledge – the modeling product. The modeling process consists of a set of guidelines for the knowledge acquisition, analysis, and representation process. Knowledge acquisition and modeling is strongly participatory because multiple stakeholder views need to be consolidated. Furthermore the resulting view represents a consensus about the decisions made thus contributing to their implementation in reality. A variety of EM approaches have been suggested and validated in practice, see, for instance, [5, 15, 16, 17, 18, 20, 21, 22, 23, 24].

[11] shows that EM can be used for two main types of objectives – (1) developing the business, e.g. developing business vision, strategies, redesigning the way the business operates, developing the supporting information systems, or (2) ensuring the quality of the business, e.g. sharing the knowledge about the business, its vision, the way it operates, or ensuring the acceptance of business decisions through committing the stake-holders to the decisions made.

Enterprise Knowledge Development (EKD) method [5] is a representative of the Scandinavian strand of EM methods. It defines the modeling process as a set of guidelines for participative way of working and the modeling product in terms of six sub-models each focusing on a specific aspect of an organization (see table 1).

The ability to trace decisions, components and other aspects throughout the enterprise is dependent on the use and understanding of the relationships between the different sub-models addressing the issues in table 1. When developing a full enterprise model, these relationships between components of the different sub-models play an essential role because they make the model traceable. They show, for instance, why certain rules, processes and information system requirements have been introduced.

	Goals Model (GM)	**Business Rules Model (BRM)**	**Concepts Model (CM)**	**Business Process Model (BPM)**	**Actors and Resources Model (ARM)**	**Technical Component & Requirements Model(TCRM)**
Focus	Vision and strategy	Policies and rules	Business ontology	Business operations	Organizational structure	Information system needs
Issues	What does the organi-zation want to achieve or to avoid and why?	What are the business rules, how do they support organiza-tion's goals?	What are the things and "phenomena" addressed in other sub-models?	What are the business processes? How do they handle information and material?	Who are responsible for goals and process? How are the actors interrelated?	What are the business requirements to the IS? How are they related to other models?
Com-po-nents	Goal, prob-lem, external constraint, opportunity	Business rule	Concept, Attribute	Process, external proc., information set, material set	Actor, role, organizational unit, individual	IS goal, IS problem, IS requirement, IS component

Table 1: **Overview of the sub-models of the EKD method [25]**

3 Background to Agile Development Approaches

Agile development approaches such as e.g. AM [4], XP[1], Crystal Clear [26], Scrum [2], and DSDM [3] all share the same values. They provide a set of principles, techniques and best practices for IS development. Some of them focus on supporting different IS development phases and some of them can easily be combined with other approaches and practices (c.f. e.g. [27]).

The agile approaches suggest delivering solutions to customer requirements in smaller parts on a frequent basis before the whole product is delivered which allows early customer feedback (c.f. e.g. [3]). The challenge of changing requirements is addressed by closely cooperating with the customer.

According to [28] agile approaches have the following characteristics – (1) iterative, (2) gradually growing (incremental), (3) self-organizing (the team in cooperation with the customer decides and prioritizes the tasks and organizes

themselves), and (4) adaptive (processes, principles, work structures are adapted according to the situation).

Since the agile approaches do not explicitly tackle modeling in terms of how to model and what, to address these issues AM emerged. AM's goal is to integrate the more traditional modeling approaches with the ideas of agile development. AM is best suited for projects with rapidly changing requirements and low demands on security. See [12] for evidence about the suitability of AM in practice.

Some of the main concepts in AM are: agile models and agile documentation. Agile models are models that are easy to understand, fulfill their purpose and are just detailed and advanced enough [4]. In addition to this, agile models should be kept up to date to ensure consistency and accuracy [4]. Agile documentation is about creating the documentation that is really needed, and the focus of agile documentation often lies on how a system is used, not how it was constructed.

The philosophy of AM relies heavily on communication between project members and stakeholders. This puts heavy requirements on the individuals involved in AM, on their ability to communicate and their skills in knowledge acquisition, analysis, and systems engineering.

AM also recommends several practices such as teamwork, simplicity of work, and validation. The modeling in AM should be iterative and incremental. To achieve this, several models should be created in parallel, and the increments should be relatively small. The modeling should be participative and the models displayed publicly. Every artifact that is created should be kept simple and the simplest tools should be used. To validate the work in AM, everything should be testable and proven with code [4].

4. Current Issues in Projects Using Agile Approaches

The agile development approaches primarily focus on the development of a software system. The underlying assumption of agile projects is that the customer is to a some extent certain about what kind of system is needed, what are its features, and who will use the new system and how. In practice, however, IS development is a part of some business development or change management project. Moreover, in the course of the project the developers and the stakeholders may have to face a number of ill-structured or wicked problems [29]. In such a case we need to assess and explore various business development alternatives, which then need to be taken into account when developing the supporting IS, because there often are several means of achieving the same business goal.

A common situation is to set up business goals of the new IS system before the development project is commissioned. Sometimes this is done without involving the developers (e.g. subcontractors) of the new IS. These business goals might be vague and obscure. They might reflect only the official information about the project and leave out valuable knowledge about the existing problems, challenges and hidden agendas, which will further undermine the possibility of achieving the project goals (e.g. develop the functionality that is really needed by the business). This immediately puts the IS development team at a disadvantage because not all information is communicated to them. The agile development approaches try to address this

challenge by active stakeholder involvement and having a customer representative on the development site. However, these practices might not be enough because the agile team has to re-acquire the knowledge about who the stakeholders are and what their intentions are, and then rediscover the real business objectives of the system. The team might also need to access stakeholders who are high level managers and are either too busy or disinterested in engaging in IS development projects.

Agile teams do not usually perform a thorough business analysis, which means that they do not require the business people to think systemically about their business and to connect their business needs to the new IS. In some cases the IS development team might even be unaware of some of the business decisions affecting their work which then leads to unnecessary rework.

A lot of the knowledge about the new IS is scarce and tacit – it lies in the heads of a few stakeholders. They may also have different opinions about the same issue, especially concerning the future plans of the organization, which, therefore, need to be consolidated, made explicit, and transferred to the agile development team. A successful practice, suggested by XP, is "customer on site". In reality, however, only a few stakeholder types can be kept on site. High level managers such as CEOs or CFOs need to be engaged differently.

5. Using EM in Agile Development Projects

This section analyzes the potential of using EM, taking EKD as an example, in agile development projects. We discuss the objectives and compatibility of the artifacts of AM and EM, the compatibility of Agile Model Driven Development (AMDD) [30] and the EKD EM process, as well as the specifics the EM process and the tool support.

The integration of AM and EM suggests new emergent properties of the IS development approach that would be achieved by utilizing dimensions at which both methods are strong and dimensions where one or the other method may be superior. E.g. agile methods handle IS development at both, - theoretical and practical levels of development. EM does that only at a theoretical level. On the other hand, EM can be considered as more powerful than AM with respect to the following dimensions [31]:

- Dimension of holistic learning: AM facilitates software developers' learning about the expected performance of the system [32]; EM facilitates business level learning about strategic and operational alignment and correspondence between business system and the IS. Integration of AM and EM suggests an emerging development property: holistic organizational learning during the IS development process. This is achieved by a better understanding of business goals, idea generation, tests and experiments, as well as by systemic information storage and distribution.
- Dimension of contingency: Expert opinions show that contingency is not a built-in feature of agile approaches. Systemic explicit models provided by EM on the other hand compensate this weakness of AM if we integrate both methods.
- Dimension of disciplined and educated tailoring: An Enterprise Model may serve as a useful map for tailoring IS development efforts and thus bring in more assurance in the development process than in other AM cases.

– Dimension of measurement of output: Expert opinions show that AM does not support this dimension, while EM can support at least partial measurements of the output by measurable objectives and goals in the GM and TCRM.

Consequently, utilization of EM in agile development aims at a systemic enterprise level agility by providing explicit knowledge based means for change management, enterprise and software systems configuration management and other activities, which are essential constituents of enterprise agility, i.e. the ability to change rapidly in terms of the business goals, concepts and processes [33, 34].

5.1 Compatibility of Objectives of AM and EM

Agile projects commonly use AM or at least a part of its recommendations, principles and practices. The purpose of AM is to support the process of analysis and design – to discover, understand, illustrate, and describe some development artifact and to facilitate communication about this artifact between developers and stakeholders. In this process both parties should eventually reach a consensus, so that the development process can go on. The emphasis is on models that are of a manageable size. As one practitioner of AM explains:

"… the purpose [of AM is] to create few and manageable concepts and manageable entities and to develop a language, to create a conception of your project, so that you, when you say "customer", know what you mean by that." – Interviewee i7 quote in [12].

Similarly, in the context of IS development, the objectives of EM are to capture business or stakeholder requirements. Since the EM process is participative, only those aspects of the problem domain that are relevant and important to the project and the product are modeled. This directly contributes to the core principles of AM "travel light" and "model with purpose".

Enterprise models are developed participatively which increases the understanding and commitment of the team, thus contributing to the values of AM "communication", as the following citation indicates

"Active participation leads to commitment. So, by creating active participation you make it impossible for people to escape commitment." – Interviewee i5 quote in [11].

In both approaches, AM and EM, the group work is used to achieve a consensus, understanding and commitment. AM differs from EM regarding the following issues:

– "Project stakeholders do not know what they want. Project stakeholders are unable to see beyond the current situation" [35]. AM just presents these issues as challenges of the agile development projects. EM addresses these problems by providing helpful means for discovering the requirements not only with respect to the current situation but also with respect to further organizational situations.
– AM uses tacit enterprise knowledge and the tacit relationship between the enterprise level knowledge and the IS development knowledge. EM supports knowledge externalization with explicit knowledge models to ensure transparency between the enterprise, requirements and software design knowledge [36, 37].
– AM only emphasizes the requirements prioritization in change management [35] while EM supports reusable explicit domain knowledge, i.e., it enables to create an engineered vision of a new way of working [38].

Similarities between AM and EM suggest that EM is suitable for agile projects. Moreover, the differences in both approaches suggest that using EM in agile projects can contribute to achieving a new level of agility, where agility is based on the use of both tacit and explicit knowledge and is explicitly addressed not only at the level of software product development but also at the level of enterprise development per se.

5.2 Compatibility of Modeling Artifacts of AM and EM

Relationships between AM artifacts and EKD sub-models are shown in table 2. Only those artifacts that are suggested for long term keeping [8] are addressed here. We distinguish four relationship types between AM artifacts and EKD sub-models:

- The AM artifact is an element or a subset of an Enterprise Model sub-model.
- The AM artifact is an aggregate of the Enterprise Model's elements belonging to one or several sub-models – derivable from the EKD activities.
- An AM artifact can be partially represented with Enterprise Model elements.
- An Enterprise Model element can serve as a reference to a particular agile requirement modeling artifact.

Table 2 shows that relevant AM artifacts are related to at least one EM (EKD) modeling artifact. Therefore, we can argue that Enterprise Model is able to serve as a reference framework for organizing and maintaining AM artifacts.

AM artifact	**Is an element or a subset of the Enterprise Model sub- model**	**Is an aggregate of elements of the Enterprise Model sub- model**	**Partially overlaps with the Enterprise Model sub- model**	**Has a reference point in the Enterprise Model sub- model**
Business rule definition	BRM			GM, BPM, ARM
Component diagram (UML)		TCRM	CM	TCRM
Constraint definition		GM, TCRM		
Data model	CM, TSRM		BRM	GM
Deployment diagram (UML)			CM, TCRM	TCRM, GM
Essential use case			BPM, TCRM, ARM	GM, ARM, BPM, CM, TCRM
External interface				TCRM, ARM
Features			BRM, TCRM	TCRM, BRM
Glossary			CM	CM
Network Diagram			TCRM, AM	TCRM
Organization chart	ARM			
Package diagram			TCRM	TCRM
Specification language			BPM, BRM	
Table	GM, BRM, TCRM	GM, BRM, TCRM		
Technical Requirement	TCRM			
Use case diagram (UML)				GM, ARM, BPM, CM, TCRM
User interface flow diagram			ARM, CM	
Workflow diagram	BPM			

Table 2. **Correspondence between the candidate artifacts of AM for modeling requirements and EKD enterprise model's components**

5.3 Compatibility of AMDD and the EM Process

Agile development projects using EM should focus on fulfilling the business vision and requirements. This can be done by integrating prototyping approaches with the business analysis to explore various alternatives of supporting the business goals and processes by IS components and features. Using EM to capture the business knowledge pertinent to the IS development project is not the same as BRUF (big requirements up front) which contradicts with the principle of iterative and incremental development, argued against, for example, by [7], because EM does support interactive and incremental development.

[30] introduces AMDD as a framework for iterative and incremental modeling to be used in agile projects. AMDD presents the overall way of working and thinking without specifying what artifacts should be modeled. Table 3 takes the EKD modeling process as basis and shows how it contributes to AMDD. Additional contribution to Iteration 0 is achieved by the preparatory phase of the EKD process which includes activities such as identifying the project objectives and pre-interviewing the stakeholders, more about this is in [5, 9, 25].

AMDD stage	EM support
Iteration 0: Envisioning	
Initial Requirements modeling (identify high level scope and an initial requirements stack)	An EM seminar with all key stakeholders to establish the business goals of the system, to explore the business requirements and to set the overall strategy of the project. The intangible benefit is the consensus about these issues.
Initial Architecture Modeling	An EM seminar to identify architecture components of the IS on a crude level.
Iteration 1-n	
Iteration Modeling: Thinking Through What You'll Do This Iteration	EM to elaborate detailed issues concerning the iteration. E.g. elaboration of the business process that needs to be supported.
Model storming (work through specific issues, just in time (JIT) modeling, stakeholders actively participate)	Short EM events in the development team to resolve specific modeling issue that they have involving stakeholder representatives that are available on site. Involving other stakeholders would have to be planned in advance.
Executable Specification via Test Driven Development	EM supports this task by providing explicit linking to business goals, rules, and requirements which can serve as measurable constraints.

Table 3. **Combining AMDD with the EKD modeling process**

5.4 Specifics of Integrating the Agile Way of Working with EM

Guidelines for conducting EM projects [25] are also applicable in agile projects. This section discusses specifics of integrating the agile way of working and EM.

Elaborating Multiple Perspectives Iteratively

Developing an enterprise model that answers various questions and stakeholder viewpoints ensures that the agile team has a repository of explicit knowledge. Such a repository/model should address questions such as what are the goals of the customer organization, what are the business and/or organizational problems that the new IS

attempts to solve, what are the business processes and actors that need to be supported, what are the business rules and policies that affect the new IS, what resources are available/necessary, what are the IS goals and problems, etc. To address these issues we recommend holding a series of participative EM seminars in the early stages of agile projects. The agile team should not however aim at developing a complete enterprise model first and only then begin developing software as the following citation warns against:

"…if you see [the model] as our mutual opinion right now, in this question, then you are working agile. As soon as you start talking about freezing stuff and to itemize … by that you mean that (something) is decided … but that would almost be seen as failure and not as … the right way to get the knowledge." – Interviewee i8 quote in [12]

The initial Enterprise Model serves as a starting point for IS development, but the team has to keep in mind that changes will most likely occur.

"You do more modeling in the beginning of project, because when you start implementing, the model has to have reached certain maturity, otherwise you have nothing to work towards. Then, of course, if you are working really agile, you have to be prepared to remodel and do more modeling, because it might be so that you are not doing all the modeling at one time." – Interviewee i9 in [12]

The tangible benefit of this is having a repository/model of explicit business knowledge about the system and its intended usage. The intangible benefit is a better commitment to the use and acceptance of the new IS by the stakeholders.

Involve Different Stakeholder Types in Collaboration

The agile team should involve various stakeholder types in order to consolidate their opinions about the requirements and the future application of the system. They need to involve end-users and occasional users as well as the stakeholders that have indirect relation to the system, e.g. high level managers who will benefit from the system in terms of greater work efficiency of their subordinates. The agile team is to be involved in the EM process to become familiar with the models and with the stakeholders. The modeling facilitator should be a part of the agile team.

The tangible benefit is the discovery and integration of various views and opinions about the requirements thus giving a more complete knowledge about the IS to be built. This makes the iterative and incremental development more efficient, because less redesign and rework is needed. The intangible benefit is the promotion of the system and increased acceptance of the IS by various stakeholder types.

Link Other Models and Designs with the Enterprise Model

In agile projects the IS development activities should begin as soon as the team has enough knowledge to identify the overall system architecture and set targets for the first iteration. To do this, the Enterprise Model does not need to be complete. Artifacts, such as models and designs produced as part of the agile IS development process, should be linked with the Enterprise Model. This will allow identifying which aspects of the domain knowledge or which requirements expressed in the model are supported by the current version of the IS. Table 2 shows how some of the AM artifacts can be linked to EKD sub-models.

The tangible benefit of such a linking is the possibility to identify how different features of the system contribute to business goals, business process and requirements. The intangible result is a reduced need for redesign and rework.

Agile EM in Requirements Process

When EM becomes a part of an agile project it must be done in an agile manner, thus we can introduce the concept of Agile EM. Following the underlying principle of agile development, the Agile EM process must add value to the project. The main purpose of using EM in agile projects is to provide the team with high quality requirements in terms of their fit to organization's needs.

[6] distinguishes between the following three requirements methods that mitigate risks in agile development projects – Extreme requirements method, Agile requirements method, and Robust requirements method.

In the extreme requirements method the vision of the system is only verbal and the unit of requirements gathering is the user story, which describes the functionality chunk that provides value for the user. User stories are written by customers "on site". Using this method little attempt is made to understand or document future requirements. This method recommends having a use case model that, at least loosely, structures the meta-knowledge about user stories. We suggest using EM instead of writing use cases, by reflecting meta-information about user stories in Enterprise Model's sub-models. In this case the EM process consists of meta-knowledge extraction from user stories and its reflection in appropriate sub-models. A tangible benefit of this is a more structured meta-knowledge in comparison to use cases and time savings because there is no need to write the use cases for getting a systemic view on the requirements. An intangible benefit is that developers do not have to restructure their knowledge several times according to different modeling formalisms.

Using the Agile requirements method the vision is no longer verbal and its development method is incremental. Use cases have specifications that elaborate the sequence of events, the pre- and post-conditions, and the exceptions and alternative flows. In this method the vision could be developed at the beginning of requirements gathering using EM. Later the use cases could be attached to the corresponding sub-models of the Enterprise Model. The tangible benefit of using EM is a shared vision instead of just a vision and a systemic overview of use cases. The intangible benefit is a smoother software development process due to a shared vision.

The Robust requirements method utilizes all tools of the Agile requirements method but on a larger scale and in a more robust manner, including product planning and validation of requirements. Additional modeling techniques such as activity diagrams, message sequence diagrams are also used. In this case EM can be applied iteratively until a consensus among all stakeholders is achieved concerning the vision, concepts and requirements. Enterprise Model elements may be linked to the requirements artifacts amalgamated in requirement management tools if there is a possibility to document the link. Tangible and intangible benefits in this case are the same as using the Agile requirements method.

To achieve a real agility in EM we have to develop tools that support not only representation of the Enterprise Model, but also provide means for effective linking of Enterprise Model elements and other artifacts of agile development projects.

Tool Support of Agile EM

Agile teams are used to simple and effective tools. EKD similarly suggests using simple tools such as a large plastic sheet on the wall and colorful post-it notes to develop an enterprise model participatively. In participative EM projects the rule of thumb seems to be to use simple tools like a whiteboard or the plastic sheet for brainstorming and to use a computerized tool and a beamer for "polishing" – refining the existing models. This principle is also applicable in agile teams. The main motivation for using simple tools is that every stakeholder is able to contribute to the model at any time. On the contrary, if a computerized modeling tool is used during a modeling seminar, then the stakeholders take turns to channel their input through a tool operator, which usually slows down the creative process. Using a synchronous collaboration tool also seems too cumbersome and immobile for agile developers as the following citation shows:

"It [i.e. modeling with simple tools] is an efficient way to get an overview, there can be several people, you can jump in and modify the picture for each other. The model becomes dynamic and something you build together. …there certainly are such (collaboration) tools, but it feels a little strained to say – let's sit down and work via network in this program, so that we all can draw in this textbox at the same time. It becomes complicated to talk to each other simultaneously." – Interviewee i6 quote in [12].

Only those models that will be kept "alive" during the development project are to be redrawn into a computerized tool. Others may just as well be discarded, because their value has been consumed once the developers resolved the issue at hand [10].

5 Concluding Remarks

We have shown that EM has a potential to be useful in agile development projects. More specifically, the benefits of using EM in agile projects are the following:

- Enterprise model explicitly documents as-is and to-be situation of the organization and helps to elicit IS requirements.
- The EM process connects the agile IS development team with the management level where all strategic decisions concerning the project are made.
- Enterprise model explicitly documents dependencies between the real situation in the organizations, the future state of business, in which the new IS will be used, and the business decisions made during the IS development process.
- Enterprise model allows the agile team to analyze the business impact and the consequences of various design alternatives.
- Configuration management of enterprise model artifacts on a business level allows a more efficient development of business rules which to be incorporated in the IS.
- Use of EM may reduce the need for redesign and rework in agile IS development process due to availability of explicit knowledge.

Agile development approaches provide a set of general guidelines, most of which are independent of the application context. It is efficient to combine agile development with EM methods, e.g. with EKD, because they support participative discovery and integration of multiple stakeholder perspectives and knowledge. Our

assumption is that EM is useful not only for small and medium scale agile development projects but also for achieving agility on a large scale and in complex enterprises and projects. However, we do not suggest using EM in all situations. In fact, EM usefulness depends on the organizational and situational context it is applied in (c.f. [5, 9, 11, 25]).

The proposed integration of AM and EM has been partly applied at two IS development projects at Riga Technical University, namely "Professional Orientation Information Base in Computer Science and Information Technology" and research project "Development of the Prototype for the Support of Inter-Institutional Flow of Knowledge". The EKD process contributed to the consensus building between the different stakeholders while the resulting models established the project's "backbone" of knowledge. This gave the team an opportunity to be agile at different project and organizational activities. The latter project followed the EM process guidelines of [25] closer than the former project, which resulted in smoother work overall. This suggests that not only explicit artifacts of EM and AM influence project's agility, but also the growth of participants' tacit knowledge is to be taken into consideration. A deeper analysis of the tacit knowledge dimension in integrated AM and EM activities is a goal for future research aimed at developing methods for IS engineering of agile enterprises.

References

1. Beck, K. (2004). Extreme programming explained: Embrace change. Addison-Wesley.
2. Schwaber, K., & Beedle, M. (2002). Agile software development with SCRUM. Prentice Hall.
3. Stapleton, J. (2003). DSDM business focused development. Addison-Wesley.
4. Ambler, S. (2002). Agile modeling: Effective practices for extreme programming and the unified process (1st ed.). John Whiley & Sons Inc.
5. Bubenko, J. A. Jr., Persson, A., & Stirna, J. (2001). User guide of the knowledge management approach using enterprise knowledge patterns, deliverable D3, IST project "Hypermedia and Pattern Based Knowledge Management for Smart Organisations", Royal Institute of Technology, http://www.dsv.su.se/~js/ekd_user_guide.html.
6. Leffingwell, D. (2002). Agile requirements methods. Rational Software.
7. Ambler, S. (2007). Examining the "Big Requirements Up Front (BRUF) Approach". Retrieved November 27, 2007, from http://www.agilemodeling.com/essays/examiningBRUF.htm
8. Ambler, S. (2007). Artifacts for agile modeling: The UML and beyond. Retrieved November 27, 2007, from http://www.agilemodeling.com/essays/modelingTechniques.htm.
9. Persson, A. (2001). Enterprise modeling in practice: Situational factors and their influence on adopting a participative approach, PhD thesis, Stockholm University, Sweden
10. Stirna, J. (2001). The influence of intentional and situational factors on enterprise modeling tool acquisition in organisations PhD thesis, Royal Institute of Technology, Sweden
11. Persson, A., & Stirna, J. (2001). An explorative study into the influence of business goals on the practical use of enterprise modeling methods and tools. In G. Harindranath et al. (Eds.), New perspectives on information systems development: Theory, methods, and practice. Kluwer.
12. Jönsson, M. (2004). Agile modeling in Sweden – from practices to principles, MSc thesis, Stockholm University and Royal Institute of Technology, Stockholm, Sweden

13. Lagerquist, I., Lindmark, M., Stirna, J., & Nyfjord, J. (2006). Adoption of agile development in practice: A qualitative inquiry. In Industrial proc. of EuroSPI'06, Finland.
14. Fraser, J. (Ed.). (1994). Enterprise state of the art survey (Part 5. Technologies supporting enterprise modeling, DTI ISIP Project Number 8032). AIAI, The University of Edinburgh. Retrieved August 8, 2004, from http://agilealliance.org/home.
15. Yu, E. S. K., & Mylopoulos, J. (1994). From E-R to "A-R" – Modeling Strategic Actor Relationships for Business Process Reengineering. In Proc. ER'94, Manchester, England.
16. Fox, M. S., Chionglo, J. F., & Fadel, F. G. (1993). A common-sense model of the enterprise. In Proceedings of the 2nd Industrial Engineering Research Conference. Norcross GA: Institute for Industrial Engineers.
17. Zorgios, Y. (Ed.). (1994). Enterprise state of the art survey (Part 3. Enterprise modeling methods). AIAI, The University of Edinburgh.
18. Dobson, J., Blyth, A., & Strens, R. (1994). Organizational requirements definition for information technology. In ACM (Ed.), Intern. Conf. on Requirements Engineering 1994.
19. Goldkuhl, G., Lind M. (2004) Developing e-interaction - a framework for business capabilities and exchanges, in proceedings of ECIS 2004
20. Castro, J., Kolp, M., & Mylopoulos, J. (2001). Tropos: A requirements-driven software development methodology. In Proc. CAiSE 2001, Springer.
21. Bajec, M. and Krisper, M. (2005). A methodology and tool support for managing business rules in organisations. Information Systems, 30 (6), pp423-443.
22. Rolland , C., Prakash N. (2000) Bridging the gap between Organizational needs and ERP functionality, Requirements Engineering Journal
23. Krogstie, J., Jørgensen, H. D., (2004), Interactive Models for Supporting Networked Organizations. Proceedings of CAiSE'2004, Springer LNCS
24. Van Lamsweerde, A., Dairmont R., Massonet P., (1995) Goal Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt, Proc. of RE'95, IEEE
25. Stirna, J., Persson A., Sandkuhl K., (2007) Participative Enterprise Modelling: Experiences and Recommendations, in proc. of CAiSE'07, Springer LNCS, ISBN 978-3-540-72987-7
26. Cockburn, A. (2005). Crystal clear: A human-powered methodology for small teams. Addison-Wesley.
27. Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. In Proc. of Conf. on Software Engineering, IEEE
28. Boehm, B., & Turner, R. (2004). Balancing agility and discipline. A guide for the perplexed. Addison-Wesley.
29. Rittel, H. W. J., & Webber, M. M. (1984). Planning problems are wicked problems; Developments in design methodology (N. Cross, Ed.). Chichester: John Wiley & Sons.
30. Ambler, S. (2007). Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development. November 27, 2007, from http://www.agilemodeling.com/essays/amdd.htm
31. Conboy, K. A. (2006). Framework of method agility in information systems development, PhD thesis, University of Limerick
32. Nerur, Sr., & Balijepally, VG. (2006). Theoretical reflections on agile development methodologies: The traditional goal of optimisation and control is making way for learning and innovation. Communications of ACM, 50(3), 79-83.
33. Browning, T. R., Fricke, E., & Negele, H. (2006). Key concepts in modeling product development processes. Systems Engineering, 9(2), 104-128.
35. Ambler, S. (2007) Overcoming Requirements Modeling Challenges, November 27, 2007, http://www.agilemodeling.com/essays/requirementsChallenges.htm
36. Bubenko, J. Jr., & Kirikova, M. (2004). Improving the quality of requirements specification by enterprise modeling. In A. G. Nilsson, et al. (Eds.), Perspectives on business modeling: Understanding and changing organisations. Springer-Verlag.

34. Conboy, K., & Fitzgerald, Br. (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. In Proc. of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research.
37. Kirikova, M. (2004). Interplay of tacit and explicit knowledge in requirements engineering. In H. Fujita & V. Gruhn (Eds.), New trends in software methodologies, tools and techniques: Proceedings of the Third SoMet_W04 (pp. 77-86). IOS Press.
38. Kirikova, M. (2000b). Potential role of enterprise models in organizational knowledge processing. In E. Kawaguchi, H. Kangassalo, H. Jaakola, & I. A. Hamid (Eds.), Information modeling and knowledge bases XI.