

Comparison of business processes and agile software development processes

Stefan Eicker, Tim Gottschalk, Annett Nagel

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen
Universitätsstraße 9
45141 Essen

{Stefan.Eicker; Tim.Gottschalk; Annett.Nagel}@icb.uni-due.de

Abstract: Well-defined software development processes are rare established. This especially is true for agile software development processes, which become more and more popular. The paper evaluates how far agile software developing processes can be seen as a kind of business processes. This forms the basis for the selective transferring of methods and tools from the field of business processes to the field of software development processes.

1 Introduction

Software development is a process embedded within a dynamic or even chaotic environment. Additionally the software as the core object of software development becomes more and more complex and comprehensive (Ha07, p.89f). Providing success of software development projects, enterprises have to optimize the development processes by using effective software development tools on the one hand and agile approaches on the other hand (MR08, p. 1). The first step is the integration of agile software development processes within the organization. Those agile processes do not only target a higher quality of the software itself, but also a higher effectiveness and efficiency of the development teams. Furthermore agile software development processes provide approaches to cope with increasing complexity and dynamic of different kinds of software development projects. These approaches are based on agile management and method frameworks (LM04, p. 29ff).

The paper evaluates how far agile software developing processes can be seen as a kind of business processes. This evaluation forms the prerequisite for the transfer of methods and tools from the field of business processes to the field of agile software development processes. Scrum and eXtreme Programming (XP) are exemplary chosen for the presentation of agile software development processes as they are very popular and often used in the field of agile software development (Ve08, p. 5). The research method used is a combination of an analysis of related literature and argumentative-deductive reasoning.

The goal of using those methods is the warranty of a systematic analysis of commonness's and differences concerning the fields of business processes and software development processes. On this base methods can wittingly be transferred from one field to the other.

The software development process can be separated into three different levels: The first one is the project management level, the second one the process level and the third one the level of practices and activities. Each agile approach addresses one or more of these levels (ASR02, p. 95). Also different approaches addressing different levels can be combined to an overall approach for all levels. An example is XP@Scrum, where the approaches of Scrum for the levels of (project) management and process (process definition) (1st and 2nd level) are combined with practices and activities defined by XP for the 3rd level.

2 Added value in software development processes

Porter has strongly differentiated between primary activities and supporting activities concerning the added value of an organization (Po00, p. 1ff). His approach has to be extended in the context of software development as there are additional secondary activities (cf. Figure 1). The primary activities are corresponding to the actual software development process and therefore to the added value. The secondary activities on the one hand are supporting the primary added value, but on the other hand they are even a part of it. Thus the supporting activities not only provide a service to the primary but also to the secondary activities - not creating a part of the added value directly.

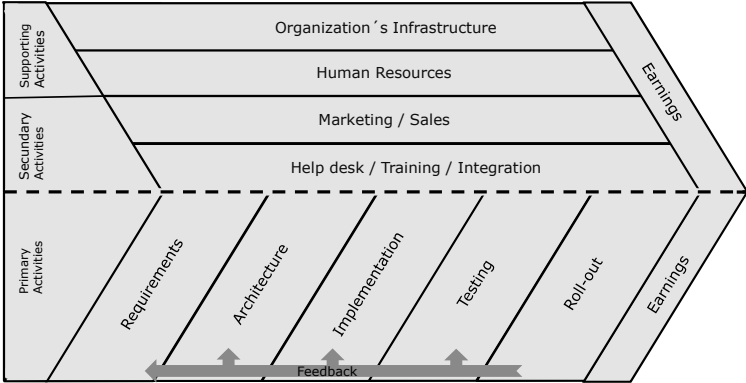


Figure 1: Added value of software development processes (According to Po00)

Therefore the actual development of the software within the development process is the primary activity. Secondary activities are the help desk, training of the customers, support or activities by the sales department by analyzing the market in form of studies and at the end sell them or using the results within the own organization. Those activities are supported by activities of the departments of infrastructure or human resources.

The structure of the activities and business processes of a software development organization therefore is comparable to those of a classic organization: The processes are separated into primary and secondary business processes supported by standardized processes (e.g. human resources). The actual development - the primary business processes - instead is individual in every software development project. That is useful as the development process can fulfill the specific requirements and conditions of each individual project. This aspect - the flexibility of processes - is called "Tailoring" and is the crucial factor of success within a dynamic environment since a non-dynamic process model is pointless in any kind of software development processes (MSV97, p. 1f).

3 Aspects of the business process management-oriented organization in the context of agile software development

Schmelzer and Sesselmann specified an organization based on business processes upon eight aspects (SS08, p. 83). These aspects can also be found in the basic approaches of agile software development processes. Principally the agile processes are based on the approaches of the agile manifest (Ma01): Individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. The approaches of the agile manifest are obviously an antagonism to the basic process orientation respectively process organization. The following presentation of the core aspects in the context of agile software development indicates, that the structure of agile processes is explicitly based on the same approaches as business processes are. The aspects are referenced to the three levels of the software development process in the context of agile software development.

3.1 Stronger customer orientation

One of the main goals of an organization is to solve the problems of their customers by providing solutions, i.e. products or services, to fulfill their needs and finally satisfy the customers. Depending on their grade of satisfaction customers buy the products or services and therewith the consistency of an organization can be assured.

Using the mechanisms and methods of business process management (BPM) those aspects become more important. Especially in the field of software development the focus is on the requirements of the customers and the ability to react quickly and flexible to changes of requirements. The term "customer" is general for a person, an organizational unit or an organization who/which ordered the development of software. The aspect of a stronger customer orientation is being accounted in the different agile development approaches in miscellaneous ways.

On the management and process level Scrum defines a product owner who is part of the development team. He always is available in case of questions or ambiguities on the part of the software developers. (Pi08, p. 9; Sc07, p. 114) On the level of practices and activities XP defines different approaches for this aspect.

E.g. the on-site customer is similar to the product owner defined by Scrum, the planning game as well as short releases. The customer is directly involved in many of the activities which results in a high customer orientation.

3.2 Less interfaces

One of the core elements of business process management is the reduction of interfaces within an organization. This is done by using a process-oriented organization instead of a function-oriented one. Focusing the process-oriented organization means that an organization is no longer diverted into vertical functions where everybody fulfills only one task without knowledge what goes on behind or before his working station. So there won't be bottlenecks anymore and everyone has the final product or service in mind.

The reduction of interfaces actually relates to a function-oriented organization. The basic thought of agile development is that the development team iterates through all process steps several times until the software is finished. Due to that fact that there is e.g. no dedicated requirements manager or architect there is no function-oriented but a process-oriented organization. Thus there are less interfaces than there would be if the development process would be organized in a function-oriented manner.

Furthermore this aspect has to be seen in two different contexts: On the one side the interfaces apply to the different tools used in the process of software development like the integrated development environment (IDE) or a source code management system. On the other hand interfaces between the stakeholders are referenced here. The amount of interfaces between the heterogeneous systems is a question of the integration scale.

Scrum as an approach on the management and process level defines a daily scrum meeting. This meeting is compulsive and ensures that information is transported directly from the source to the target and not passing any interstations in form of other people which generally distort information. Therefore on the level of practices and activities the agile development approaches require a physical adjacency of the workplaces of the developers and the customer representative (product owner / on-site customer). Together with the required daily meetings this ensures a face to face communication which is the most efficient way of communicating for human beings (VS07, p. 9).

3.3 Less coordination effort

In the context of process-oriented organizations the existence of coordination should be rather less. Not only there are fewer interfaces needed but also communication is much more efficient, because everybody knows the whole process and therefore knows who and when to ask for special concerns.

This aspect deals with the necessary coordination in processes. Agile software development approaches are based on a self coordinating team.

The teams run through the iterations several times and thus a concrete coordination is only necessary for a short period of time which is on the one side easier and requires less effort than coordinating a long period on the other side, e.g. due to less complexity. This aspect focuses the management and the process level of the agile development approach. Regarding the practices and activities level this aspect takes care of the fact that teams plan and discuss the next iteration together with the customer and thus the teams coordinating themselves. Another aspect on this level is that there is not more than one customer representative supposed to be. The teams exactly have one contact person. Hence on the side of the customer there is no necessity for coordination between several contact persons. These practices and activities reduce the coordination effort of the agile software development process (Pi08, p. 17f).

3.4 Clear responsibilities for the process result

The result of a process is based on the working results of all involved people. The goal of business process management is that every one of them shows responsibility and interest in the final product or service. Therefore everybody involved has to reach a certain stage of quality to satisfy the customer and even shows a responsibility for this.

In the context of agile software development the whole team is responsible for the result of the process - in this case the developed software. On the management and process level this means that the members of the team act like a team e.g., they support each other. This aspect requires team building which is a process of learning. It is described in detail under the aspect “organizational learning”. Due to matter of fact that they are planning their activities themselves as well as how they accomplish these activities for the entire process they are responsible for the output. On the practice and activity level the practice of collective ownership is defined (LRW02, p. 94 ff).

3.5 Systematic increase of the process performance

This aspect can be inherited from the previous ones. As the involved people not only try to save the quality of the products and services but also try to satisfy the customers, they are also interested in an increasing process outcome.

The increase of the process performance is closely linked to the adoption of the required and suggested practices in the context of agile software development. This means the team has to develop skills in using the practices and especially has to modify the practices to the requirements of the organization and the special needs of individual projects respectively project kinds. Due to the fundamental role of the team an important aspect is to optimize the composition of the team and their skills in different areas. To identify where this is necessary and how this can be done the agile principles suggest reflecting on “*how to become more effective, then tunes and adjust its behavior accordingly*” (Manifesto 2001).

Another aspect is the integration regarding the interaction of the used systems to support the development process like Computer Aided Software Engineering (CASE) tools.

Increasing process performance can be reached by optimizing the use of the suggested practices and also by increasing the scale of integration towards the optimum (SA05, p. 201).

3.6 Multidimensional control based upon the target figures customer satisfaction, time, quality and costs

Besides the primary goal of the satisfaction of the customer's needs, financial aspects shall not be forgotten. Also the time of the production is important, because a customer can't wait to long - even for a "big" product. And the costs of course have to be taken into account, too. So these factors all together have to be matched to the big picture: assuring the organization's consistency.

The management of agile software development regarding the figures customer satisfaction, time, quality and costs is supported by several practices on the three levels of the development process. The first level project controlling is based upon classical approaches of project management like milestones. Regarding the process level and the customer satisfaction one aspect is, that the customer is directly involved into the development due to the on-site customer. Another aspect is that the customer is able to "see" the iteration (process) result very early due to the short time frame of an iteration and the short release cycles. The figure time on the process level focuses on the fact to not exceed the planned time for the project on the one hand for the entire project and on the other hand for each iteration. This is reached by the fact that the team, including the customer, are planning the next iteration and thus the team knows which functionalities can be implemented in the given time. The daily meetings required by the process level where everybody gives a status uncover possible delays in a matter of days.

Quality has to be ensured on the level of practices and activities. The focus here is on quality which is visible to the customer and not on internal quality attributes like code quality (nevertheless XP provides a practice for this aspect too). The quality is mainly ensured by two different activities. One is the automated testing of the code and the other one is a short release cycle. The automated testing helps to avoid errors in releases because a wide range of tests can be accomplished within a short time. Short release cycles ensure the quality regarding the usability and functionality of the software in its actual or final using environment (LEW02, p. 41ff).

Based upon the project triangle the costs are depending on the quality and the time and thus are not explained here in detail (OF03, p. 17).

3.7 Decentralized controlling

Not only will every process be controlled by another department using e.g. activity-based costing, due to the fact that everybody is involved in the process and has to have the end quality of the product or service in his mind, the controlling can be done on different stations and levels. So besides the central controlling there is always a decentralized one e.g. using the results of a special service hotline.

The controlling of the software development process is decentralized because the development team defines at the beginning of an iteration itself, which functionality has to be implemented. How far this target is going to be reached is monitored by the daily status meeting. Possible delays due to problems can be identified early and necessary arrangements can be made to avoid any kind of delay. If the defined targets are not reached, the reasons are analyzed after the iteration but before starting the next one. The results are used to improve the process based upon organizational learning. This controlling approach is described by Scrum on the process level. Nevertheless it is not contemplated how this should be done. (Sc04, p. 9)

3.8 Organizational Learning

The continuous process improvement (CPI) is one of the most important aspects within the BPM. Reaching CPI special approaches like the BPM cycle consisting of four phases (strategic process management, process design, process implementation, process controlling) and corresponding methods can be used to enhance the business processes (Al05, p. 1ff). To provide “better” processes - that are cheaper and faster and have a higher quality - all people involved in the process will therefore increase their knowledge by learning to use special methods and even find new ways on their own.

In the context of agile software development this aspect primary refers to the specific practices and proceedings and how they are used and learned by the developers working in the team. One of the basic thoughts in agile developing is naturally spoken that the developer team works as a team. Based on the fact that no defined interfaces between different processes do exist (e.g. there are no subassemblies), another aspect is the knowledge about the characteristic of the team members, i.e. which capabilities and qualifications do they have. According to Tuckman (Tu65, p. 76ff) a team hereby passes through four phases Forming, Storming, Norming and Performing. Many years experience show, that new teams - who’s members worked separately before - usually need 2-3 months reaching the phase Norming (Pi08, p. 18f). Changes of the team - equal if small or big ones - result in a re-cycling of all four phases. Therefore the composition of the team ideally should not be changed (Tu65, 78f).

4 Conclusion

The comparison of agile software development processes and the classical primary business processes using their central aspects shows the similarity of the software development process and the classical product development process. Thus the software development process is one of the primary business processes of software development organizations. Although software development processes differ from classical business processes, their macro structures (core processes and supporting processes) are equal (SS08; TW07). Therefore it can be followed - regardless to the named differences - that special approaches of BPM are transferable to software development processes.

In a further step it will be researched if and in which way the approach of flexible BPM is transferable to agile software development processes. Connected to this it will be analyzed in which way the approaches of flexible BPM need to be tailored to the software development environment. Furthermore an evaluation of classical business processes and agile development processes with the focus on the paradigms of flexibility and service-orientation will be done.

References

- [ASR02] Abrahamsson, P.; Salo, O.; Ronkainen, J.: Agile Software development methods - Review and analysis. Research report, VTT Publications, Espoo, 2002.
- [Al05] Allweyer, T.: Geschäftsprozessmanagement. Strategie, Entwurf, Implementierung, Controlling. W3L GmbH, Herdecke, 2005.
- [Ha07] Hamilton, P.: Dynaxity - Management von Dynamik und Komplexität. Springer, Heidelberg, 2007.
- [LM04] Lindvall, M.; Muthig, D.: Agile Software Development in large Organizations. IEEE - Computer 12, 2004, pp. 26-34.
- [LRW02] Lippert, M.; Rook, S.; Wolf, H.: Software entwickeln mit eXtreme Programming. d.Punkt, Heidelberg, 2002.
- [Ma01] Manifesto for Agile Software Development: Principles behind the Agile Manifesto, 2001, <http://www.agilemanifesto.org/principles.html>. Retrieved 2010-01-07
- [MR08] Meisinger, M.; Rittmann, S.: A comparison of service-oriented development approaches. Institut für Informatik der Technischen Universität München, 2008.
- [MSV97] Münch, J.; Schmitz, M.; Verlage, M.: Tailoring großer Prozeßmodelle auf der Basis von MVP-L. In: Montenegro S et al. (publ.): Vorgehensmodelle - Einführung, betrieblicher Einsatz, Werkzeugunterstützung und Migration. Beiträge zum 4. Workshop. Sankt Augustin: Forschungszentrum Informationstechnik, GMD-Studien (311), 1997, pp. 63-71.
- [OF03] Osterloh, M.; Frost, J.: Prozessmanagement als Kernkompetenz. Wie Sie Business Reengineering strategisch nutzen können. Betriebswirtschaftlicher Gabler, Wiesbaden, 2003.
- [Pi08] Pichler, R.: Scrum-Agiles Projektmanagement erfolgreich einsetzen. dPunkt, Heidelberg, 2008.
- [Po00] Porter, M. E.: Wettbewerbsvorteile. Campus, Wiesbaden, 2000.
- [SA05] Salo, O.; Abrahamsson, P.: Integrating agile software development and software process improvement: a longitudinal case study. Empirical Software Engineering, 2005, pp. 10ff.
- [SS08] Schmelzer, H. J.; Sesselmann, W.: Geschäftsprozessmanagement in der Praxis. Carl Hanser, München, 2008.
- [Sc07] Schwaber, K.: The Enterprise and Scrum. Microsoft Press, Redmond, 2007.
- [Sc04] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Redmond, 2004.
- [TW07] Thränert, M.; Werner, A.: A Process Family Approach for the reuse of development processes. In: Sobh T (publ.) Innovations and Advanced Techniques in Computer and Information Sciences and Engineering. Springer, Netherlands, 2007.
- [Tu65] Tuckman, B. W. Developmental Sequence in Small Groups. Psychological Bulletin: 384-399, 1965.
- [Ve08] VersionOne: 3rd Annual Survey: 2008 "The State of Agile Development", 2008, http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf. Retrieved 2010-01-07
- [VS07] Vigerschow, U.; Schneider, B.: Soft Skills für Softwareentwickler. dpunkt, Heidelberg, 2007.