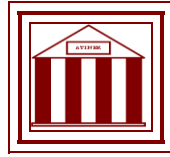


Athens Institute for Education and Research
ATINER



ATINER's Conference Paper Series
COM2012-0015

Comparing Agile and PMBOK® – Time Management

Nikolay Todorov

PhD Student

Bulgarian Academy of Sciences

Bulgaria

Tsvetelina Kovacheva

PhD Student

Bulgarian Academy of Sciences

Bulgaria

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209
Email: info@atiner.gr URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm

Printed in Athens, Greece by the Athens Institute for Education and Research.
All rights reserved. Reproduction is allowed for non-commercial purposes if the
source is fully acknowledged.

ISSN **2241-2891**

An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. The papers published in the series have not been refereed and are published as they were submitted by the author. The series serves two purposes. First, we want to disseminate the information as fast as possible. Second, by doing so, the authors can receive comments useful to revise their papers before they are considered for publication in one of ATINER's books, following our standard procedures of a blind review.

Dr. Gregory T. Papanikos
President
Athens Institute for Education and Research

This paper should be cited as follows:

Todorov, Nikolay and Kovacheva, Tsvetelina (2012) "Comparing Agile and PMBOK® - Time Management" Athens: ATINER'S Conference Paper Series, No: COM2012-0015.

Comparing Agile and PMBOK® - Time Management

Nikolay Todorov

PhD Student

Bulgarian Academy of Sciences

Bulgaria

Tsvetelina Kovacheva

PhD Student

Bulgarian Academy of Sciences

Bulgaria

Abstract

The paper compares the processes and practices defined by the internationally recognized standard - Project Management Body of Knowledge (PMBOK® Guide) and the agile methodologies for managing software development projects (becoming extremely popular and attractive nowadays). The goal is to show that there is a considerable mapping between mentioned approaches for software projects management. The paper emphasizes on the knowledge area of Time Management, following the PMBOK® defined processes and comparing them to the Agile techniques and practices.

It is a common opinion that agile and PMBOK® ideologies for managing a software development projects are quite contradictory. Agile values and focuses on the final results and collaboration and is often criticized for being non-disciplined, whereas PMBOK® relies on the well documented project planning and its strict monitoring and control.

The PMBOK® Guide defines nine knowledge areas within the project management lifecycle. Each of them consists of several processes comprising a full set of 42 processes in the standard. In this paper we focus on one of the areas – Project Time Management. For this area we go through its underlying processes, inputs, tools, techniques and outputs and look for corresponding practices in agile software development methodologies which actually implement the items defined in the PMBOK® process.

Keywords: Agile, PMBOK®, Time Management, Scrum

Acknowledgements: This work was sponsored by the Bulgarian National Science Research Fund through contract ДДЛОКФ 02/04/2010

Contact Information of Corresponding authors:

Nikolay Todorov – nikolay.todorov@gmail.com

Tsvetelina Kovacheva - tskovacheva@abv.bg

1. Introduction

Agile methodologies are intended to be used in software projects' development. Several major methodologies exist - Extreme Programming (XP) (Beck, 1999), Scrum (Scwaber, 2004), Feature-Driven Development (FDD) (Palmer and Felsing, 2002), Adaptive Software Development (ASD) (Highsmith, 2002) and others. They are trying to reduce the risks by developing in short periods of time called iterations, usually lasting between one and four weeks. Each iteration is like a separate software project including all of the phases necessary to develop and deliver new functionality – planning, analysis, design, coding, testing and documentation.

The Guide to the Project Management Body of Knowledge (PMBOK® Guide, 2008) is a recognized standard for the project management profession. As is well known, a standard is a formal document that describes established norms, methods, processes and practices. As with other professions such as law, medicine, and accounting, the knowledge contained in this standard evolved from recognized good practices of project management practitioners who contributed to the development of this standard.

Today's project manager in software development projects faces many challenges. Demands and pressures have increased due to competitive environments, complex solutions and changing technology—further complicated by economic conditions.

To deal with these challenges, a project manager needs to rethink traditional approaches and consider a more flexible one. Effective project management not only requires a mastery of traditional techniques but also the knowledge, wisdom, and ability to bend, throw out or rewrite the rules when the situation requires it.

Being more flexible in your mindset helps you adhere to the philosophies behind the agile approach to project management. Gartner (Murphy and Norton, 2009) predicts that this approach will be used on 80 percent of all software development projects by the end of 2012.

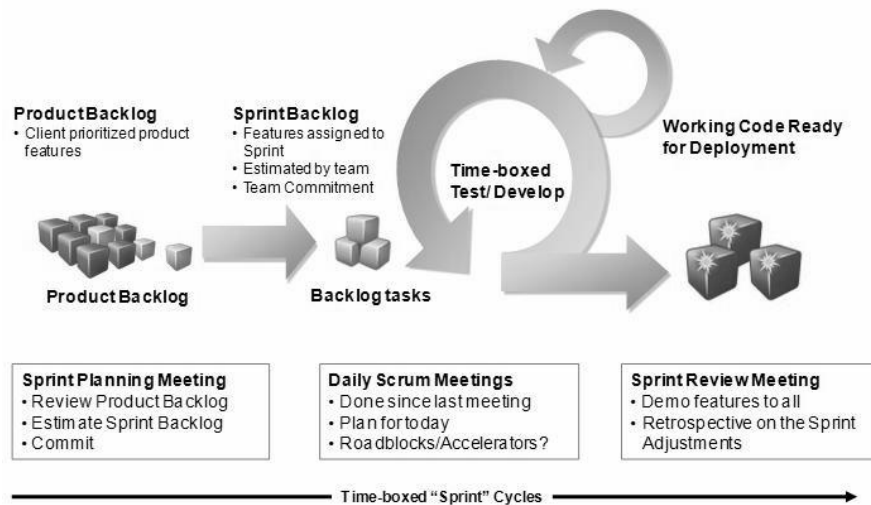
It is important to state that in this paper we do not select a particular agile methodology (e.g. XP or Scrum) but consider them as a whole because of the following reasons:

- The latest trends in software development are to use the term agile in general, emphasizing on the iterative approach and agile practices we use as a natural response to the current pressing business needs and expectations.
- Many of the latest agile practices are not considered as a part of a concrete methodology but generally they refer to a general notion (e.g. planning poker, agile retrospective, continuous integration, etc.)
- Different methodologies offer different sets of practices and using a combination of them will help us to better map to the PMBOK® processes.

2. Project lifecycles

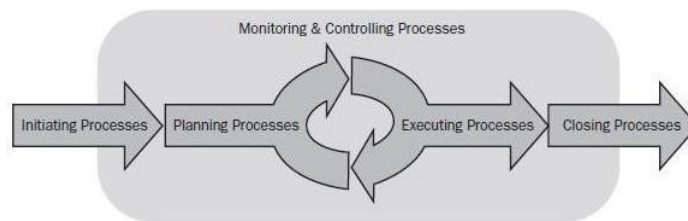
Agile software methodologies are well known with their iterative approach for delivering project's or product's valuable increments. An example of the Scrum lifecycle can be summarized using the following diagram:

Figure 1. Scrum lifecycle (Focus on Agile, 2012)



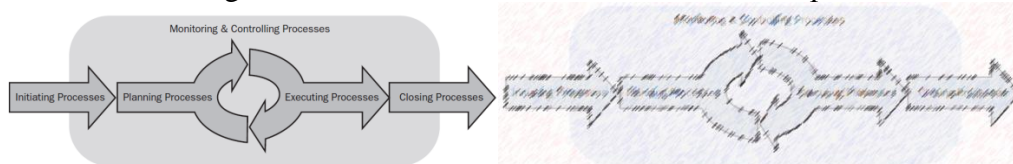
A Guide to PMBOK® (2008) defines five concrete project phases – Initiation, Planning, Execution, Monitoring and Control, Closing.

Figure 2. PMBOK® (2008) Project Phases



But it does not restrict only to this sequence of phases for the whole project as it also defines an iterative relationship, where only one phase is planned at any given time and planning for the next is carried out as the work progresses on the current phase and deliverables. This approach is useful in largely undefined, uncertain or rapidly changing environments such as research, but it can reduce the ability to provide long term planning. The scope is then managed by continuously delivering increments of the product and prioritizing requirements to minimize project risks and maximize product's business value (PMBOK® 2008).

Figure 3. PMBOK® Iterative Phase Relationship



This could be used as an initial testimony that the analyzed approaches (agile and PMBOK®) have a common base in their lifecycle ideology.

3. PMBOK® knowledge areas and processes

The PMBOK® Guide defines nine knowledge areas within the project management lifecycle. Each of them consists of several processes comprising a full set of 42 processes in the standard. As this is a huge set of processes to analyze in this paper we focus on one of the areas – Project Time Management. For this area we go through its underlying processes, tools, techniques and outputs and look for alternative practices in agile software development methodologies which actually implement the items defined in the PMBOK® process.

According to PMBOK®, Project Time Management includes the processes required to manage timely completion of the project. Together with their associated tools and techniques they are documented in the schedule management plan.

In agile, the iterations are always time boxed to a certain period (could be between 1 and 4 weeks). Therefore the time is managed within this frame using the corresponding practices. This helps to keep the team focused on the most important goals and tasks.

We will go through each of the PMBOK®'s processes defined in the knowledge area of Time Management and for each of their inputs, tools and techniques and outputs will try to identify equivalent agile artifacts of practices. For these purposes we will use the main agile definitions in The Scrum Guide (Scrum.org, 2011), XP (Beck, 1999) and other.

3.1. *Define Activities*

Define Activities in PMBOK® is the process of identifying specific actions to be performed to produce the project deliverables. The Create WBS process identifies the deliverables at the lowest level in the Work Breakdown Structure (WBS), the work package. Project work packages are typically decomposed into smaller components called activities that represent the work necessary to complete the work package. Activities provide a basis for estimating, scheduling, executing, monitoring and controlling the project work.

In agile methodologies the technique to collect requirements is called developing user stories. User stories provide us with a way of having just enough written down that we do not forget and that we can estimate and plan while also encouraging this time of communication (M. Cohn, 2009). In Scrum we have the two part sprint planning meeting in the beginning of the iteration. The Sprint Backlog defines the work, or tasks, that the team defines for turning the Product Backlog it selected for that Sprint into an increment of potentially shippable product functionality. The team compiles an initial list of these tasks in the second part of the Sprint planning meeting (Schwarber, 2004).

We can use the following Table 1 to define a mapping between the PMBOK®'s inputs/tools and techniques/outputs and agile practices. For each element based on the analysis we will mark the level of mapping that we have by equivalent agile artifact(s) that implement it. The mapping scale will use one of the three options – Yes, Partially and No. We will use the same comparison tables for each of the next processes:

Table 1. Define Activities

PMBOK®	Agile	Mapping
Inputs		
Scope baseline	Product Backlog	Yes
Enterprise Environmental factors	Issue tracking systems (e.g. Jira)	Yes
Organizational process assets	User stories definition process, development practices, application architecture	Yes
Tools and techniques		
Decomposition	Splitting the user stories to tasks	Yes
Rolling wave planning	It is enough to identify tasks only enough to start the work	Yes
Templates	Issue tracking system's items are used for tasks definition	Yes
Expert judgment	It is expected that agile team members are skilled enough for their job	Yes
Output		
Activity list	Sprint Backlog defines tasks for the iteration	Yes
Activity attributes	These are managed in the issue tracking system used	Yes
Milestone list	The iteration itself is the only milestone	Partially

Based on this analysis we see that we have 9 full and 1 partial mapping between Define Activities process area and agile methodologies.

3.2. *Sequence Activities*

Sequence Activities in PMBOK® is the process of identifying and documenting relationships among the project activities. Activities are sequenced using logical relationships. Every activity and milestone except the first and the last are connected to at least one predecessor and one successor. It may be necessary to use lead or lag time between activities to support a realistic and achievable project schedule.

After identifying the necessary tasks to accomplish the sprint goals the agile team also defines the dependencies between them. These dependencies are however more loosely coupled and it is not mandatory that all of them have a strict order as the team is self-organized and progressively plans their work during daily stand up meetings. It is worth to note that dependencies supported in agile issue tracking systems are usually only Finish-to-start.

Table 2. Sequence Activities

PMBOK®	Agile	Mapping
Inputs		
Activity list	As described in the previous section	Yes
Activity attributes		Yes
Milestone list		Partially
Project scope statement	Product Backlog	Yes
Organizational process assets	Project layers and architecture	Yes
Tools and techniques		
Preceding diagramming method	Dependency relation in issue tracking system (only FS)	Partially
Dependency determination	Usually only mandatory (blocking) dependencies are defined.	Partially
Applying leads and lags	One of the agile goals is to eliminate waste, therefore also leads and lags	No
Schedule network template	Sprint Backlog is the template with a prioritized list of tasks	Yes
Output		
Project schedule network diagram	Defined in Sprint Backlog as a prioritized list of tasks	Yes
Project document updates	Sequencing may result in task description updates or identify risks	Yes

Here we have 7 full, 3 partial and 1 element with no mapping.

3.3. Estimate Activity Resources

Estimate Activity Resources in PMBOK® is the process of estimating the type and quantities of material, people, equipment, or supplies required to perform each activity.

In software engineering one of the main resources to plan is the human productivity in terms of their intellectual effort. Therefore estimations in almost all kind of projects (agile or not) are done in person hours/days. One of the most commonly used techniques in agile projects for estimation is the consensus based planning also called planning poker game. It is a consensus-based approach using a Fibonacci like set of numbered cards. Planning Poker can be used with story points, ideal days, or any other estimating unit. For tasks it is usually ideal hours.

Table 3. Estimate Activity Resources

PMBOK®	Agile	Mapping
Inputs		
Activity list	As described in the previous sections	Yes
Activity attributes		Yes
Resource calendars	People availability (e.g. vacations) is considered while calculating the	Yes

	available ideal hours	
Enterprise Environmental factors	Planning poker cards	Yes
Organizational process assets	Planning poker process, ideal hours concepts (e.g. 6 per day)	Yes
Tools and techniques		
Expert judgment	The first round of poker planning for each task is based on the expert judgment of each team member	Yes
Alternatives analysis	Estimations may differ between team members based on their alternative approach to the task. This is discussed until consensus is reached.	Yes
Published estimating data	Tasks are so granular that they cannot rely on generally published data	No
Bottom-up estimating	Task based estimations are bottom-up as they sum to the whole story estimation	Yes
Project management software	Issue tracking systems are used to collect and sum estimations which is manually compared to available resources	Partially
Output		
Activity resource requirements	Based on estimation sums per story and as a whole it could be decided if sprint goals are achievable. As the iteration is time-boxed stories can be added or removed to the sprint.	Yes
Resource breakdown structure	We have only one type of resource – hours and the breakdown can be done by components (UI, server, testing, etc.)	Partially
Project document updates	Estimating of resources may result in sprint goals and backlog updates	Yes

In the Estimate Activity Resources process there are 10 full, 2 partial and 1 missing equivalences.

3.4. *Estimate Activity Durations*

Estimate Activity Durations in PMBOK® is the process of approximating the number of work periods needed to complete individual activities with estimated resources. Estimating activity durations uses information on activity scope of work, required resource types, estimated resource quantities, and resource calendars. The inputs for the estimates of activity duration originate from the person or group on the project team who is most familiar with the nature of the work in the specific activity. The duration estimate is progressively elaborated and the process considers the quality and availability of the input data.

In agile processes we have the very important concept of ideal days. Although you usually have a regulated 8 hour working day it is considered that you can productively work on new tasks only a limited time during this period as you also spend time on

meetings, discussions with other team members, configurations issues, etc. Usually it is considered that the ideal day is 6 hours. Based on this concept, team members availability, sprint goals (backlog) and tasks estimations it is decides what can be fit in the iteration's fixed duration.

Table 4. Estimate Activity Durations

PMBOK®	Agile	Mapping
Inputs		
Activity list	As described in the previous sections	Yes
Activity attributes		Yes
Activity resource requirements		Yes
Resource calendars		Yes
Project scope statement		Yes
Enterprise Environmental factors	Velocity from previous sprints	Yes
Organizational process assets	Ideal hours concepts (e.g. 6 per day)	Yes
Tools and techniques		
Expert judgment	It is the team's expert judgment that decided if goals and tasks are achievable within the iteration	Yes
Analogous estimating	Similar tasks from the same and previous iterations can be used as a base	Partially
Parametric estimating	Metrics like number of methods to be implemented can be used as a base	Yes
Three-point estimate	This is actually replaced by the planning poker's all team estimations	Yes
Reserve analysis	Ideal hours are the main reserve concept. Other than this the goal is to eliminate all wastes and inefficiencies	Partially
Output		
Activity duration estimates	Duration is dependent on the actual work estimation and ideal hours coefficient	Yes
Project document updates	Estimating of duration may result in sprint goals and backlog updates	Yes

The mapping here shows that 12 PMBOK® elements have their corresponding agile practice and for 2 of them this is partial.

3.5. *Develop Schedule*

Develop Schedule in PMBOK® is the process of analyzing activity sequences, durations, resource requirements and schedule constraints to create project schedule. Entering the activities, durations and resources into the scheduling tool generates a schedule with planned dates for completing project activities. Developing an

acceptable project schedule is often an iterative process. It determines the planned start and finish dates for project activities and milestones. Schedule development can require the review and revision of duration estimates and resource estimates to create an approved project schedule that can serve as a baseline to track progress. Revising and maintaining a realistic schedule continues throughout the project as work progresses, the project management plan changes and the nature of risk events evolves.

In agile methodologies the Sprint Backlog is the actual schedule during the next iteration. It contains the list of tasks necessary to be implemented, ordered by priority, having the dependencies defined between them and the estimated ideal time to complete them. However, as the team is self-organized the backlog and schedule may evolve during the sprint based on team’s progress reported on daily stand up meetings.

Table 5. Develop Schedule

PMBOK®	Agile	Mapping
Inputs		
Activity list Activity attributes Project schedule network diagram Activity resource requirements Resource calendars Activity duration estimates Project scope statement	As described in the previous sections	Yes Yes Yes Yes Yes Yes Yes
Enterprise Environmental factors	Sprint Backlog is defined in the issue tracking system	Yes
Organizational process assets	Priorities definition	Yes
Tools and techniques		
Schedule network analysis	The Sprint Backlog is rather a flat structure of consecutive tasks based on priorities	Partially
Critical path	The critical path is actually the full set of tasks defined for the iteration. The team is self-organized to achieve the goal eliminating waste	Yes
Critical chain	Team members (as main resources) should be fully committed to the project	Yes
Resource leveling	Team is self-organized and ideal day concept also helps to level their effort	Yes
What-if scenario analysis	The iteration itself is a relatively short and visible period and it is not necessary to plan alternative scenarios	No
Applying leads and lags	The team’s goal is to optimize their work and eliminate waste	No
Schedule compression	Overtime is rarely acceptable but for sure not allowed in two consecutive	Partially

	weeks	
Scheduling tool	The issue tracking system is used to define and track progress on the Sprint Backlog	Yes
Output		
Project schedule	The Sprint Backlog	Yes
Schedule baseline	Rarely backlog is changed within the sprint so we have it also as a baseline	Partially
Schedule data	The issue management system where tasks from the backlog are maintained supports all the necessary attributes	Yes
Project document updates	Schedule development finalizes backlog tasks definition	Yes

For Develop Schedule we have 16 full mappings, 3 partial and 2 missing.

3.6. Control Schedule

Control Schedule in PMBOK® is the process of monitoring the status of the project to update project progress and manage changes to schedule baseline. The process is concerned with:

- Determining the current status of the project schedule
- Influencing the factors that create schedule changes
- Determining that the project schedule has changed
- Managing the actual changes as they occur

In agile process the team tracks on a daily basis the actual effort spent on a task and an updated expectation of the remaining hours in order to complete it. This data is used to generate the Sprint Burndown Chart which shows the actual progress and if there are any deviations and corrective actions are needed. The team reports the status also on the daily standup and this may incur changes to the Sprint Backlog but only after a discussion and agreement with the Product Owner.

Table 6. Control Schedule

PMBOK®	Agile	Mapping
Inputs		
Project management plan	The plan for the iteration is the Sprint Backlog together with the Product Backlog.	Yes
Project schedule	The Sprint Backlog	Yes
Work performance information	Actual effort spent and tracked per task together with the updated remaining estimation.	Yes
Organizational process assets	Sprint burndown chart, Kanban whiteboard	Yes

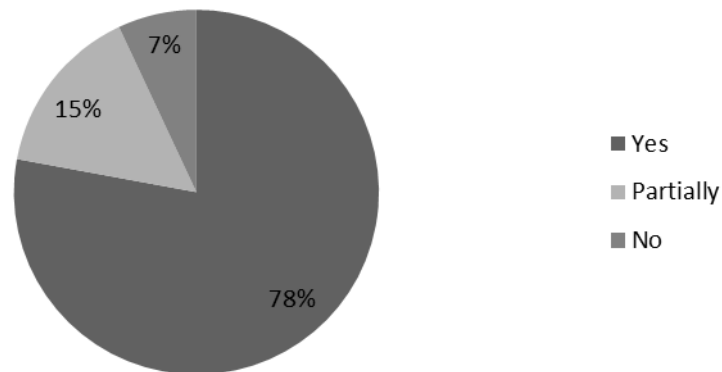
Tools and techniques		
Performance review	Status and performance is reviewed on daily stand up meetings	Yes
Variance analysis	Variance is calculated in the burndown in comparison to the ideal line	Yes
Project management software	Most issue tracking systems support burndown visualization of the progress	Yes
Resource leveling What-if scenario analysis Applying leads and lags Schedule compression Scheduling tool	As described in the previous section	Yes No No Partially Yes
Output		
Work performance measurements	The actual difference in the burndown from the ideal line is the performance measurement indicator	Yes
Organizational process assets update	Retrospective meetings at the end of a sprint may bring scheduling process improvements	Yes
Change requests	Changes are generally not desirable during a sprint. At the review meeting at the end new stories may be identified	Partially
Project management plan updates	The output of a sprint may result in Product Backlog and project roadmap update	Yes
Project document updates	Sprint Backlog is constantly updated during the sprint based on the progress	Yes

In this last process there are 13 PMBOK® elements finding their equivalent in agile, for 2 of them they are partial and 2 are missing at all.

4. Results

Based on the afore mentioned comparison of the processes defined in PMBOK®'s knowledge area of Time Management and their corresponding practices in agile software methodologies we can summarize the mapping between them in the following figure:

Figure 4. PMBOK® and Agile - Time management



It is visible that most of the PMBOK® inputs/tools and techniques/outputs in the processes of Time Management have their corresponding practices and artifacts in agile methodologies. The majority of incompatibilities are based on the fact that PMBOK® takes in mind also long term scheduling where more alternative scenarios and network of activities and their dependencies should be considered. While this is not applicable to the short time boxed iterations in agile where the level of unpredictability is decreased to a minimum.

5. Conclusions and future work

Our goal in this paper was to refute the understanding that Project Management Body of Knowledge processes contradict the agile software development methodologies. We choose the PMBOK® knowledge area of Time Management and for each of its processes we tried to show in a systematic way the similarities between its inputs, tools, techniques and outputs and the existing and well known agile practices and artifacts.

The results showed that in most of the cases the two approaches for managing software projects have much in common and they perform the same, however using different terms and templates.

As future work in this area we plan and have already started to extend the PMBOK® knowledge areas to include not only the Time Management processes. Similar analysis technique could be applied in order to further identify potential broader mapping between PMBOK® and Agile. This detailed elaboration is expected to lead to an efficient and effective agile implementation of the defined and established PMBOK® processes, which is also the subject of our future research interests.

This work was sponsored by the Bulgarian National Science Research Fund through contract ДДОКФ 02/04/2010

6. References

- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Reading, Mass., Addison-Wesley, ISBN 0201616416
- Schwaber, K., (2004). *Agile Project Management with Scrum*, Microsoft Press, ISBN 073561993
- Palmer, S.R. and Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development*. Upper Saddle River, NJ, Prentice-Hall
- Highsmith, J. A. (2000). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, NY, Dorset House Publishing
- Project Management Institute (2008), *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*, 4th Edition, 2008
- Murphy Th., Duggan J., Norton D., Prentice B., Plummer D., Landry S. (2009), *Predicts 2010: Agile and Cloud Impact Application Development Directions*, Gartner Focus on Agile (2012), available at <http://www.agile-training-courses.com/agile-methodologies.html> [27 March 2012]
- Scrum.org (2011), *The Scrum Guide*, Available at <http://www.scrum.org/scrumguides> [17 March 2012]
- Cohn, M. (2009), *User Stories Applied: For Agile Software Development*, Addison-Wesley, ISBN 0-321-20568-5