2015

Agile Drafting of IT-Outsourcing Contracts

Bjarne Rerup Schlichter
Aarhus University, Denmark, brs@mgmt.au.dk

Kristian Kromann Reumert Storgaard
Aarhus University, Denmark, kst@kromannreumert.com

AGILE DRAFTING OF IT-OUTSOURCING CONTRACTS

A conceptualizing case study

Completed Research

Schlichter, Bjarne Rerup, Aarhus University, Aarhus, Denmark, brs@mgmt.au.dk

Storgaard, Kristian Kromann Reumert, Aarhus, Denmark, kst@kromannreumert.com

Abstract

The concept of 'agility' has become quite popular in the development of IT-artefacts and has created interest in the more general project management literature. The process of drafting complex contracts for large IT projects (service as well as software) is often done under time pressure and in several parallel tracks using different competencies. By the use of an illustrative case-study, this paper explores how scrum can be applied to enhance the process of drafting outsourcing contracts. The analysis indicates that the use of an agile method, such as Scrum, can be beneficial in this context with a minimal adjustment and that the elements of roles, processes and artefacts may lead to better coordination and efficiency as well as higher quality. The paper concludes with suggestions for further research and discussion of the findings.

Keywords: Contract drafting, Agility, Scrum

1 Introduction

Agile methods are highly used in the development of software and based on a belief that the project in a broad understanding continuously needs to be adjusted based on the learning acquired during the process. The drafting of contracts to frame the agreement between supplier and customer has shown to become more and more difficult, following the increased size and complexity of the IT-artefacts to be supplied, being software, hardware, services or a combination of these. One of the areas where specialists (lawyers) in charge of drafting these contracts have experienced problems is in the outsourcing of IS/IT-services (Lacity et al., 2010). In other contexts, the concept of agility has been seen as one of the means to handle complexity and changing requirements (Smith and Fingar, 2003), and thus this paper identifies issues in the current work practice and explores how the drafting of these complex contracts (the contractual framework and set of written agreements regulating the delivery of the services) can be conceptualised by using the artefacts from Scrum.

The offset of the paper is the work done by lawyers, who carry out or manage the actual drafting of complex IT and sourcing contracts. Structural and internal as well as external factors in the business of lawyers' calls for cooperation and a more project based approach to providing legal advice and legal services (Storgaard, 2014b). It is seen that over the years, lawyers, especially in larger practices, have transformed from being generalists to become more specialized. The size and complexity of the cases naturally lead to the forming of groups of lawyers to meet the demand for resources in due time or the need for highly specialized competences.

The issues described above are prevalent during drafting of complex IT contracts where the actors involved face many of the same problems that the agile methods try to facilitate. Since agile methods have matured and are well documented in the development part of the procurement of software it seems fruitful to try to adapt this concept to other parts of the procurement process, including the contract drafting process.

By assuming that the use of agile methods can improve both the process of drafting as well as the contract itself, the following Research Question has been formed:

How can drafting of complex IT-outsourcing contracts be conceptualized using the agile method Scrum?

This paper presents the contract drafting process in a practice of lawyers. First, the paper introduces the theoretical background on agile development, including Scrum, and presents how contemporary research in project management can frame these ideas. The research approach section describes the case, the data collection, and data analysis. The findings section presents our analysis of the actor's perceptions and use of agile elements in this context. Hereafter we discuss how the analysis addresses the research question and contributes to previous research. Finally, we summarize the conclusions of the paper and suggest how to proceed the research.

2 Theoretical Background

This section provides a theoretical and contextualised introduction to agile development and management of contracts.

2.1 Rethinking Project Management

The agile approach has been used in the general understanding on how to modernize project management, especially in the stream of research related to Rethinking Project Management (Svejvig and Andersen). Traditionally a project has been seen as a tool applied to a single assignment, focusing on meeting time and quality under the available resources. This single-track approach applies a logical and chronological way through a set of (more or less) well-defined tasks.

Newer ideas on management of projects introduce a plenteous of approaches, i.e. by understanding the project as a temporary organization, where learning, diversity, temporality, complexity, un-security and sociability are in play (Winter and Szczepanek, 2009). The increased complexity has led to the development of agile methods and to the acceptance of the need for a contemporary organization of projects that not only mechanically executes a process towards a narrow, product-oriented goal, but accepts projects as something business-like and value-creating (Davenport, 2013).

2.2 Agile development

Agile methods build on the assumption that the development of software can be complicated as well as un-predictable where upfront detailed planning is not always realistic. This class of methods shares the wish continuously to deliver value and learning (Cadle and Donald, 2008;p.78-80). Agile methods apply iterative and incremental development and hence divides the process into a number of smaller chunks – iterations – each ending with a delivery of functionality which is – in principle – ready for shipment. The detailed analysis of requirements and possibilities is done during the process at the beginning of each iteration (Schwaber and Beedle, 2002). Thus, no detailed requirement specification exists initially. The final delivery is decided during the iterations by exploiting the experience and knowledge achieved. This approach assures quality as well the possibility to adjust the project in a volatile environment thus reducing risk (Cervone, 2011, p.18-22). The principles constituting the agile concept are given in the 'agile manifesto' (Beck et al., 2001) and establish, among 11 others, that the highest priority is to provide the customer with early and continuous delivery of valuable software, illustrating that agility is not about abolishing traditional practices, but moving the focus from a 'locked' requirement specification to the business goals of the customer (Martin, 2003).

2.3 Scrum

Scrum is an iterative and incremental development approach, where planning is concurrent to the development activities and the work is divided into smaller chunks called sprints. Each sprint is planned to be self-contained, leading to a new running version on the road to the final software product (Jakobsen and Sutherland, 2009). Scrum is a framework to be used when developing complex products and based on the theory of empirical process control theory (empiricism), assuming that knowledge comes from experience and that decisions must be based on what is known (Sutherland and Schwaber, 2013) to optimize predictability and reduce risks.

Empirical process control is based on three pillars (Schwaber and Beedle, 2002):

Transparency (The participants should have insight and be able to understand the process.)

Inspection (This must be controlled on a recurrent basis)

Adaption (if needed, adjustments to meet the goal must be done)

Scrum is based on three main sets of elements: Roles, Event/Ceremonies and Artefacts. The elements have specific goals and are, as a whole, essential to the use and success of Scrum (Schwaber and Beedle, 2002). To establish a foundation for conceptualization the main set of Scrum Artefacts are presented below.

2.3.1 Roles (The Scrum Team)

The Scrum team is self-organizing and interdisciplinary, thus having the competences needed to act independently (Sutherland and Schwaber, 2013). The team consists of a Product Owner, a Development Team and a Scrum Master and is designed to optimize flexibility, creativity and productivity.

The **Development Team** is a group of software developers assigned to finalize the Sprints by delivering a potential shippable chunk of functionality (SW) respecting the budget.

The **Product Owner** is overall responsible for (maximizing the value of) the product and maintaining contact to the ‘customer’ and other stakeholders, thus being able to answer all questions.

A **Scrum Master** is a "gatekeeper" ensuring that the Scrum Team adheres to Scrum theory, practices and rules, as a of "serving leader" for the team. The Scrum Master offers coaching of the team and protects against disturbance from outside.

2.3.2 Ceremonies

Scrum is based on a set of fixed meetings and rituals to ensure regularity, and minimize obsolete meeting activity (Rubin, 2012,p. 7):

The **Sprint** is a delimited period of time (max one month) where the development of a chunk of software, in principle shippable, takes place. The idea is that no changes in neither the goals of the Sprint, the quality standards nor in the composition of the team are allowed.

The **Daily Scrum** is a <15 minute meeting used to synchronise activities and compose a plan for the next 24 hours. Achievements since last Scrum are inspected and each member has to account for:

“What did I do yesterday that helped the Development Team meet the Sprint Goal?”

“What will I do today to help the Development Team meet the Sprint Goal?”

“Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?” (Sutherland and Schwaber, 2013, p.10)

Based on this, the Development Team can evaluate the progress towards the Sprint Goal and at the same time evaluate the progress in relation to the Sprint Backlog (e.g. by using the Burn Down Chart).

The **Sprint Review** is done by the Scrum team at the end of each Sprint with the purpose to inspect the software produced and update the Product Backlog. Often other stakeholders, such as management and sales, other developers and support staff take part to get insight in status and to give feedback. The Sprint Review is, as other meetings meant to be limited in time (e.g. four hours at a one-month-sprint).

The **Retrospective** is done by the end of each Sprint and is a review of achievement in the Sprint and updating the Product Backlog. The team and other relevant actors (management, sales staff, other developers, support functions etc.) take part in the short meeting to create a plan for improvements as well as coordination.

2.3.3 Documents

In addition to Roles and Ceremonies, Scrum applies Documents (sometimes called artefacts), these are tools designed to assure visibility and transparency for production in the Scrum Team.

The **Product Backlog** is a sorted and prioritized list of desired functionality in the final software product and hence the central, source of information about what should be produced and when. The product owner is responsible for the Backlog, including content, availability and prioritizing. The backlog is a living document based on initial insight and updated according to new requirements, experiences etc. (Rubin, 2012p. 12ff). All items contain a description, a priority and an estimate. The higher the priority, the more details. User stories (’as user of XXX, I would like YYY, so that I can ZZZ’) are often used to describe functionality in the Product Backlog.

The **Sprint Backlog** consists of the Product Backlog items assigned to the actual Sprint, and is as such a plan how to deliver the product incrementally and how the work should be done by the Development Team to transform Product Backlog Items into (parts of) a product.

The **Definition of Done** is a set of parameters to establish a shared understanding of when a delivery, a function or a product can be seen as ‘complete’ and hence to be released. In parallel the Burn Down Chart graphically indicates how well the team follows the plan. It is considered important that the actors have a shared understanding when something is done.

2.4 Scrum in other settings

Initially Scrums was developed to be used for software development, but recently several examples of use in other domains has emerged (Cadle and Donald, 2008), i.e. in universities, law defences, the car industry and development of mechanical products (REYNISDÓTTIR, 2013). At Roskilde University in Denmark, Scrum was used to rapidly produce a policy document. The evaluation indicated that Scrum could be used for development of reports and that the participants were provided with an overview of the project not seen before and should not wait for individual deliverables (Pries-Heje, 2012, p.130).

Apparently no examples on the direct use of Scrum on projects related to legal advice of drafting of contracts can be found (Goel, 2013), even though the idea of applying Scrum or agile methods within the legal industry is not totally novel, examples has been seen in litigating (Terrett, 2010). Drafting of contracts as such is well discussed (Pumphrey, 2009).

3 Current Practices

This section presents the case and its related context followed by an explanation of how we collected and analysed data. The case study approach is in terms with Cavaye’s single case with interpretive use of qualitative data for discovery (Cavaye, 1996). This interpretive research approach allowed us to investigate how the concepts of Scrum can be applied to contract drafting for complex IT-projects in their organizational and cross-cultural context and thus open to several interpretations by organizational actors, but also to us as researchers (Klein and Myers, 1999, Walsham, 1995, Walsham, 2006). One of the authors is an employee of the law firm used for the case study.

3.1 The case

The law firm, on which this case study is build, is one of the largest Danish law firms. A major part of the legal services are focused on transactions such as procurement of companies or services and hence drafting of complex contracts.

The lawyers' work is often centred on a case which can be a dispute (i.e. in form of a court case), a transaction (procurement, i.e. an IT project or something related to outsourcing) or more general advisory (i.e. on the structure of a company).

A new case will typically be staffed with a group of specialists (often from different departments. The case will always be anchored to a responsible partner (owner) and often a senior associate may be appointed as project manager to take care of progress, coordination etc. In some cases additional project support staff may be assigned to the case.

In case of complex IT and outsourcing contracts, there is a need for specialists in e.g. outsourcing, IT and terms of employment. They each develop their part of the contract in a process that could take weeks or even months. During this process, a large amount of information is collected and developed. A complex IT or outsourcing contract typically consists of a main contract and a large number of annexes. The main contract is the basic legal framework, where the annexes normally contain more detailed regulation or descriptions of different parts of the delivery, e.g. a catalogue of services, service level agreements or regulations of transfer of employees. Since all contributions will eventually be a part of same contract, there is a need to align them regarding e.g. terminology as well as assuring that an issue is not regulated (differently) more than one place.

Drafting of this type of contracts hence faces a set of challenges:

a) The work takes place in a project group with different employees not coming from the same department, not necessarily having worked as a team before (or even having limited experience from similar cases).

b) The initial level of knowledge will often be low and increasing during the project. Knowledge comes from many sources and not always in a structured form. Knowledge and experience is generated continuously, and there is an ongoing need of adaption, coordination and prioritizing of the different deliveries.

c) Requirements to the contract may change often and with short notice. Sometimes parts of the contract must suddenly be drafted differently or certain aspects regulated otherwise, e.g. due to commercial demands.

d) Time pressure can often be severe

If not taken care of, the above challenges may have negative impact on the writing speed as well as on the quality of the resulting contract. Ultimately this may impose a risk on the client and similarly liability for the lawyers.

3.2 Data Collection

The data collection included document studies and individual semi-structured interviews with team members and the management of the law firm on which the case study is based. The majority of these interviews, as well as insider observations, are documented in two reports compiled by one of the authors (Storgaard, 2014b, Storgaard, 2014a). In addition to forming the base for the analysis in the present paper, the interviews are seen as pilot interviews for the next phase of the research, since they furthermore provided an understanding of the environment and the challenges faced by the organization and helped identify additional candidates for interviewing.

3.3 Data Analysis

To understand how an agile project model can be applied for drafting of complex contracts, the analysis will be centred around a conceptualization of Scrum on the contracting process. Scrum is initially constructed with software development in mind and hence uses terminology and frames from the IT domain. To apply Scrum and evaluate the fit in a contract drafting context, an 'interpretation' is needed.

The analysis is initially done by applying the Scrum artefacts on the actors, tools and processes used during contract drafting. For each of the artefacts it is then evaluated whether it can be applied as it is or has to be adapted. In the latter situation it is then decided if the adaption should be in the Scrum framework for the contract drafting process.

4 The conceptualised framework

In this section, we present how our observations and insight from the case study can be conceptualized using the Scrum framework. The analysis has the contract drafting process in focus.

4.1 Problems related to the existing drafting practice

The interviews revealed a set of challenges of the present drafting practice of large and complex contracts (Storgaard, 2014b):

Exchange of information: A centralized flow and collection of the continuous influx of new information are difficult. As a result there is a major risk that lawyers end up working in siloes.

Task Management: The many sub-tasks are often changed, and tasks / delegation of responsibility are done structurally by using e-mails or oral agreements.

Lack of project management capabilities: The lawyers are not educated in the practice of projects and are thus lacking potential benefits and synergies.

Diversity of tools and processes: The work practice can be quite fragmented and may be based on many different tools and processes, even when applied on similar tasks.

Process Management and communication with clients: A single and straightforward system where the client can stay updated on progress is lacking.

Accounting and Estimation: Within the legal industry it is common to invoice based hours spent, which do not fully support the working in projects or even co-operation.

The analysis below indicates how some of these problems can be alleviated using a Scrum approach.

4.2 Roles

In relation to an outsourcing contract, the senior associate (senior-lawyer) who is project manager will take the role as **Product Owner**, this is in line with the present situation where this role is to assure the value of the work and coordinate 'upwards' with the stakeholders. A new task will be the responsibility for administration of a centralised Product Backlog and hence defining the acceptance criteria for the individual backlog items. This requires more time to cope with the annexes than today, but will also assure a qualified overview – which today is often not fully present. Initially we refuse suggestions of (partly) appointing the Partner as product owner, since a partner is also seen as part of the management.

The Development Team is the group of lawyers, making the actual drafting of the outsourcing contract. Similar to SW development, this team should be interdisciplinary in nature (legal specialties and branches). Some adjustments are needed though, as the senior-associate in the present setup are both part of the production team and have managerial responsibilities for the team, and this is not in line with the Scrum principles. This can be partly solved by letting the Scrum team take over the development of the framework agreement (which is often responsibility of the project leader).

The present setup is without **Scrum Master.** Since it is not required that a Scrum Master has a deep professional insight (in other words: being able to take part in the production) (Rubin, 2012, l.4285), and it is recommended that the Scrum Master is not a part of the production team, it would be feasible to let one of the non-lawyers from the project support staff take this role. This is in consistence with the present situation where the project assistants support the project, and it would also let the assistants support the Product Owner by identifying procedures for efficient management of the Product Backlog.

4.3 Ceremonies

The ceremonies are important to secure transparency and provide possibility for inspection and hence adjustment of processes and deliveries. In an outsourcing contract drafting context this can be contextualized as explained below.

The Scrum Sprint is a novelty even though sub-deliveries are not unknown. Often the contract is delivered including annexes in two or three packages according to importance. To follow the idea of Sprint, the packages must be divided into even smaller chunks and timed after the actual project. Since the duration of the project is limited to a few months, a sprint lasting one to two weeks seems feasible.

The Sprint is initiated with a Sprint Planning Meeting, where elements from the product backlog are prioritized into the sprint backlog explicitly as to what the Sprint will deliver and how.

This type of planning is not seen in the present context, mainly coursed by the introduction of the shorter sprints. Presently this division of tasks between participants (often in the form of assigning the drafting of different annexes) is done in the start-up of the process. The analysis indicates the concept of the sprint, and sprint planning can facilitate realistic and aligned goals for each sub-delivery. As a (positive) side effect, the client will experience more deliveries and ongoing progress in the project.

Daily Scrums will be a relevant activity improving transparency among the members of the production (development) team. All members will know the status of the project and problems occurring in the other tracks. This will reduce the identified risk of working in silos and enhance the efficiency (as is also seen in the Roskilde University case (Pries-Heje, 2012,p. 130)) because the team members should not wait for each other. In the light of the shorter sprint, it could maybe be beneficial to introduce more than one Daily Scrum.

Sprint Review is one of the activities that must be adjusted to be feasible in an outsourcing context. A sprint lasting 1-2 weeks will easily result in +100 pages of contract and annexes. Due to the nature of the product, it is not easy to demonstrate this and receive feedback during a two hour meeting (contrary to software, where the new functionality could often rather easily be demonstrated). Therefore – the sprint review must be adapted to the context. Either: i) the participants must read the documents beforehand, letting the meeting be limited to discussion or feedback or ii) another type of presentation. Suggestion i) is not realistic due to the workload without a comparable anticipated benefit. The practical solution may be that the development team presents highlights and/or items, where feedback is particularly needed, followed by further in-depth reading. It was also suggested to invite the responsible partner and other relevant specialists not involved on a daily basis.

The Sprint Retrospective is used to evaluate the work done in the Sprint by reflecting on persons, relations, processes and tools summarized in a plan to improve the contextualized Scrum Process. This ceremony is also quite relevant for the outsourcing contract drafting process as well, but may be met with scepticism since process evaluation is not a part of the 'culture' in this environment. In addition, the time pressure will somehow prioritize production in favour of reflection, given the often narrow deadlines.

4.4 Scrum documents

The Scrum documents are artefacts documenting status, progress or work to be done.

The Product Backlog will be a central tool in the contract drafting process. In the present process a simple list of documents with their status (level of completeness) are normal. By introducing the Scrum approach, additional dimensions such as prioritizing and estimating are added. This will for sure improve the visibility and establish a better basis for planning. The use of User Stories could also be applied on contract drafting. Just as software ultimately is to be used by the users, then an outsourcing contract is to be used by the client in respect of the client's customer or service provider. User Stories can describe 'functional' requirements for regulations seen in a client perspective. Perhaps it would be even easier for the client to describe the needs, and this would force the lawyer to stick to the business goals of the client. Some lawyers are worried by the higher workload imposed when developing and maintaining the Product Backlog.

The **Spring Backlog** and its progress, as documented in the **Burn down Charts**, can also be beneficial during the contract drafting process. In case the Product Backlog contains wishes about a specific regulation (e.g. a guaranty or specific service), the breakdown of work into items in the Sprint Backlog tells explicitly where in the contract complex this is to be done and indicates the interdependencies to be elaborated on. Since deadlines are quite firm in this context, it is very important to get an 'early warning' in case of potential problems to meet these or to take other measures to align expectation with the client.

Definition of Done, also relevant for the release backlog, would be a novel thing in contract drafting. Even though a shared understanding (maybe implicit) about the quality or type a contractual document shall have, no checklists exist. A Definition of Done could be used for quality assurance and could be adjusted to the different contracting types. A structured approach to quality assurance *during* the contracting process, instead of after, is expected to reduce the number of errors in the finalized contracts.

#	Scrum	Contracting	Observation
	Roles		
1	Product Owner	The senior lawyer already acting as project manager.	
2	Scrum Master	Assistant from the Project Center	A non-lawyer
3	Development Team	A group of Lawyers from the production	Interdisciplinary
	Ceremonies		
4	Sprint	Smaller chunks that todays 'packages'	A novelty
5	Daily Scrum	Quite relevant, introduce transparency etc.	
6	Sprint Review	Changed to e.g. presentation and discussion of pre-read documents	Difficult to present a 'running' item

7	Retrospective	To be introduced, relevant	Cultural barrier.
	Documents		
8	Product Backlog	A simple list of documents and their status to which are added: prioritization and estimating	Addition of user stories?
9	Release Backlog	Not conceptualized.	
10	Burn Down Charts	To be introduced	novelty
11	Definition of Done	To be introduced. It would be beneficial to introduced a set of shared standards	novelty

Table 1: Conceptualization

5 Discussion

The case description accounts for problems related to the present contract drafting process. Some of these problems resemble experiences in the development of software, where agile methods seem to alleviate these problems. A few examples of the first tentative attempts to apply Scrum in other domains have been identified. The analysis shows how the contract drafting process (of complex outsourcing contracts) can be conceptualised using Scrum artefacts. For each artefact, pro et cons are identified and discussed. Scrum is a framework to be used in its full extent to give the best benefit (Sutherland and Schwaber, 2013, p.16) which questions if the use of Scrum on drafting of complex IT- or outsourcing contracts fully alleviates the challenges in the contract drafting process. The answer seems to be confirmative:

1) The **Scrum Roles** can be used and the distinct division of roles and responsibilities seems beneficial. It is expected that as Product Owner, the project responsible lawyer will work more with project management (in opposition to production) – but this extra workload will be balanced with a better overview and higher efficiency in the Development Team.

2) The **Scrum Rituals** can be used with some adaption to the contract drafting process. It seems that the use of Scrum (sprint, scrums etc.) will lead to better coordination, improved exchange of information as well as ongoing learning. This is consistent with experiences from other domains (Pries-Heje, 2012, pp129-135).

3) The **Scrum Documents** can also be applied to the contract drafting process. The use of Product Backlog and Definition of Done (and maybe even a Burn Down Chart) is expected to improve quality, coordination and conformity. The use of Scrum during contract drafting seems to alleviate some of the problems, especially regarding sharing information, task management, processes and tools and somehow also accounting.

6 Conclusion and further research

This paper answers the research question: *How can drafting of complex IT-outsourcing contracts be conceptualized using the agile method Scrum?* The contract drafting process has been successfully conceptualised using Scrum concepts as shown in table 1. Suggestions on how to change the drafting process in a Scrum conceptualisation are given.

During the following research project we will, with offset in the contributions from the present paper, broaden our literature study to identify contemporary research that can support the development of our findings. It could also be beneficial to re-visit our case and collect more data to challenge and discus our findings, especially focusing on different actors (including the client side).

References

BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A. & JEFFRIES, R. 2001. Manifesto for agile software development.

CADLE, J. & DONALD, Y. 2008. *Project Management for Information Systems,* Essex, England, Pearson Education Limited.

CAVAYE, A. L. M. 1996. Case study research: a multi-faceted research approach for IS. *Information systems journal,* 6**,** 227-242.

CERVONE, H. F. 2011. Understanding agile project management methods using Scrum. *OCLC Systems & Services: International digital library perspectives,* 27**,** 18-22.

DAVENPORT, T. H. 2013. *Process innovation: reengineering work through information technology*, Harvard Business Press.

GOEL, S. 2013. Agility - Looking Beyond Software Development. *Scrum Alliance Community.*

JAKOBSEN, C. R. & SUTHERLAND, J. Scrum and CMMI Going from Good to Great. Agile Conference, 2009. AGILE '09., 24-28 Aug. 2009 2009. 333-337.

KLEIN, H. K. & MYERS, M. D. 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly,* 23**,** 67-93.

LACITY, M. C., KHAN, S., YAN, A. & WILLCOCKS, L. P. 2010. A review of the IT outsourcing empirical literature and future research directions. *Journal of Information Technology,* 25**,** 395-433.

MARTIN, R. C. 2003. *Agile software development: principles, patterns, and practices*, Prentice Hall PTR.

PRIES-HEJE, J. 2012. *Project Management Multiplicity*, Samfundslitteratur.

PUMPHREY, J. H. 2009. Thinking like a Writer: A Lawyer's Guide to Effective Writing and Editing by Stephen V. Armstrong and Timothy P. Terrell. HeinOnline.

REYNISDÓTTIR, Þ. 2013. *Scrum in Mechanical Product Development. Case Study of a Mechanical Product Development Team using.* Master, CHALMERS UNIVERSITY OF TECHNOLOGY.

RUBIN, K. S. 2012. *Essential Scrum: A practical guide to the most popular Agile process*, Addison-Wesley.

SCHWABER, K. & BEEDLE, M. 2002. *Agile software development with Scrum,* Prentice Hall Upper Saddle River.

SMITH, H. & FINGAR, P. 2003. *Business Process Management: The Third Wave,* Tampa, USA, Meghan-Kiffer Press.

STORGAARD, K. 2014a. Agil Metode Anvendt på IT-Kontrakter. Aarhus University.

STORGAARD, K. 2014b. IT-Mæssig undersøttelse af samarbejde og projktarbejde i en advokatvirksomhed. Eksamensopgave i faget "IT strategi i Praksis": Aarhus Universitet.

SUTHERLAND, J. & SCHWABER, K. 2013. The Scrum Guide.

SVEJVIG, P. & ANDERSEN, P. Rethinking project management: A structured literature review with a critical look at the brave new world. *International Journal of Project Management*.

TERRETT, A. 2010. Do Litigators "scrum" ? *Slaw*.

WALSHAM, G. 1995. The Emergence of Interpretivism in IS Research. *Information Systems Research,* 6**,** 376-395.

WALSHAM, G. 2006. Doing interpretive research. *European Journal of Information Systems,* 15**,** 320-330.

WINTER, M. & SZCZEPANEK, T. 2009. *Images of projects*, Gower Publishing, Ltd.