



LAPIN YLIOPISTO
UNIVERSITY OF LAPLAND



University of Lapland

This is a self-archived version of an original article. This version usually differs somewhat from the publisher's final version, if the self-archived version is the accepted author manuscript.

Agile Methods and Their Impact on Project Management and Contracting

Nuottila, Jouko; Nystén-Haarala, Soili

Published in:

Regulating Industrial Internet through IPR, Data Protection and Competition Law

Published: 16.08.2019

Document Version

Peer reviewed version

Citation for published version (APA):

Nuottila, J., & Nystén-Haarala, S. (2019). Agile Methods and Their Impact on Project Management and Contracting. In R. M. Ballardini, O. Pitkänen, & P. Kuoppamäki (Eds.), *Regulating Industrial Internet through IPR, Data Protection and Competition Law* Kluwer Law International.

Agile Methods and their Impact on Project Management and Contracting

§18.01 Introduction

The industrial Internet is a broad concept that includes several approaches to both technology solutions and business models. Thus, there are also alternative considerations on contracting and delivering value for this rapidly changing market. The development of third party IoT software applications, which the owners of IoT platforms have enabled through the publication of application programming interfaces (APIs), has created new business opportunities. In order to deliver commercial software that accesses the features of an IoT platform, an application developer must agree to and comply with the terms of a license agreement governing the use of that platform's APIs.

The prevalence of standardized, clickthrough license agreements is one trend in contracting, which digitization has increased. However, there is also another trend connected with new business opportunities for software and solutions developers: in software development, agile manifesto-based methods are currently the dominant approach. As such, the management and contractual challenges associated with agile methods must be discussed; these challenges usually require interaction and communication skills to solve. In this chapter, we focus on agile methods and their impact on project management and contracting.

§18.02 The Evolution of Agile Methods in Software Development

Software is everywhere nowadays, and almost all new software development occurs within projects. Managing software projects successfully is therefore crucial for a large portion of private and public organizations, and as such, the management philosophies of the software industry are constantly evolving. For decades, companies managed software development projects in a similar manner to traditional industrial projects. The first systematic model of software development, the waterfall

model, envisaged a sequential workflow of activities in software production projects.¹ The waterfall model was based on the workflow of industrial projects, not software development specifically, and was not adapted to reflect the different nature of software projects. Its separate stages (requirement analysis, design, programming, testing, and delivery) prevented work on multiple activities simultaneously, as they meant that one stage must be complete before work could begin on the next.² In addition to this rigid and inflexible software development model, the project management practices utilized in software projects also did nothing to facilitate flexibility, instead taking a formal, control-oriented approach.

As software development became a larger and better-recognized industry, interest in increasing the efficiency of operations grew. Apart from in the specific case of embedded software designed for hardware products, software development is entirely independent of physical products and production processes. As such, there is no specific order in which development activities in software projects need to be performed. The first attempt to create a software development model that did not depend on a direct sequential workflow was the spiral model proposed by Boehm in the 1990s.³ This model took an iterative approach, with several loops of similar task sets, including analysing requirements, planning implementation, testing, and verifying, following one another until the software system was complete. From a project management perspective, a project now consisted of several sub-projects, each delivering a small part of the project's overall goal. The spiral model increased efficiency and created a more responsive environment for customer input, but it changed neither the way in which software projects were negotiated, nor their contractual models. Project management principles hold that the implementation phase of a project should be isolated to ensure an undisturbed working environment for project personnel.⁴ The customer is considered to stay outside of the project implementation, only being able to interact with the project

¹ Winston W. Royce, *Managing the Development of Large Software Systems* 26 Proceedings of IEEE WESCON 328-338 N.8 (1970); Barry W. Boehm, *Software Engineering*, C-25 IEEE Transactions on Computers 1226-1241 N.12 (1976); Barry W. Boehm, *A Spiral Model of Software Development and Enhancement*, 21 Computer 61-72 N.5 (1988); Juhani Warsta, *Contracting in Software Business: Analysis of Evolving Contract Processes and Relationships* (2001) (Ph.D. dissertation, University of Oulu).

² Bill Curtis, Herb Krasner, Vincent Yun Shen & Neil Iscoe, *On Building Software Process Models Under the Lamppost*, Proceedings of the 9th International Conference on Software Engineering 96-103 (1987); Lung-Chun Liu & Ellis Horowitz *A Formal Model for Software Project Management*, 15 IEEE Transactions on Software Engineering 1280-1293 N.10 (1989).

³ Barry W. Boehm, *A Spiral Model of Software Development and Enhancement*, 11 ACM SIGSOFT Software Engineering Notes 14-24 N.4 (1986); Boehm (1988), *supra* n. 1; Warsta, *supra* n. 1.

⁴ Rolf A. Lundin & Anders Söderholm, *A Theory of the Temporary Organization*, 11 Scandinavian Journal of Management 437-455 N.4 (1955).

through formal procedures, such as change requests or similar interventions as agreed in the project contract or plan.

The spiral model was not yet flexible enough to facilitate efficient working practices in software projects. Creating software is a highly dynamic process by nature. The source of software innovations is the creative minds of the individuals involved in development projects.⁵ As soon as a new innovation is conceptualised, it can be demonstrated and tested immediately, without the need for manufacturing prototypes or conducting laboratory examinations, as is necessary in other industries. In addition, changing software features that have already been implemented is relatively easy and inexpensive because there are no physical restrictions involved, no need to dismantle a product and build it again. Agile software development methods were created to harness these dynamic features of software design.

A more flexible approach was adopted by light software development models, which companies and institutions created mainly for small, independent teams working on in-house software development. The major step forward in the progression towards agile methods came with the publication of the Manifesto for Agile Software Development (Agile Manifesto) in 2001.⁶ The Agile Manifesto created the philosophical foundation for modern software development, a new way of thinking that started to spread widely throughout the industry. The following four statements of values represent the main principles of the Agile Manifesto:

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation,
- responding to change over following a plan.

These value statements were formulated based on observations made in the field. High-level experts with extensive experience of developing better practices in creating software signed the manifesto. The agile approach emphasizes the skills and expertise of the individuals working in agile teams. Designing new software and new features in software products is an intellectual exercise and co-

⁵ Alistair Cockburn, *Agile Software Development* (2002).

⁶ Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland & Dave Thomas, *Manifesto for Agile Software Development*, <http://www.agilemanifesto.org> (accessed 14 Feb. 2019).

creating value in software development projects entails interaction between individuals with creative minds. As such, the quality of this interaction is important. The agile approach also takes account of the fact that software projects undergo constant changes due to the continuous learning they involve. This is why planning documentation is not as important in software development as it is in traditional industries: documentation becomes outdated fast and quickly ceases to be useful for the project. The founders of the Agile Manifesto saw that the results delivered to the customer throughout a software development project are more important than trying to document all of the changes that take place during the project. They consider the working software, executable code running flawlessly, to be the sole reliable proof of what was actually built in a project.⁷ Extensive documentation and specifications can guarantee neither the successful implementation of, nor customer satisfaction with, a software development project.

The Agile Manifesto emphasizes the importance of involving customers in software development projects. Direct and fast communication contributes to an amicable relationship between the parties involved in a project, and it is vital to create an open and trustful environment in order to facilitate collaborative innovation and the co-creation of value. The Agile Manifesto also builds on the observation that there are always changes involved in software development, usually a large number of changes. Thus, it is important to adjust to fast-breaking project changes, and beneficial to refocus development efforts along new lines after gaining new knowledge and understanding in the course of development. As such, in the agile development model, the customer has the authority to add, change, and reprioritize feature-specific requirements during the project.

§18.03 The Agile Development Model

The principles presented in the Agile Manifesto gained wide popularity; thousands of software practitioners have signed the Manifesto so far. As the Manifesto does not specify precise details of the software development process, there are many methods and tools that adopt the principles and values of the agile approach. One of the most popular agile development methods used in the industry is Scrum (see Figure 1).⁸ The scrum framework is an example of the agile approach and of the influence of agile methods on project contracting and project management.

⁷ Cockburn, *supra* n. 5.

⁸ Hirotaka Takeuchi & Ikujiro Nonaka, *The New New Product Development Game*, 64 Harvard Business Review 137-146 N.1 (1986); Warsta, *supra* n. 1.

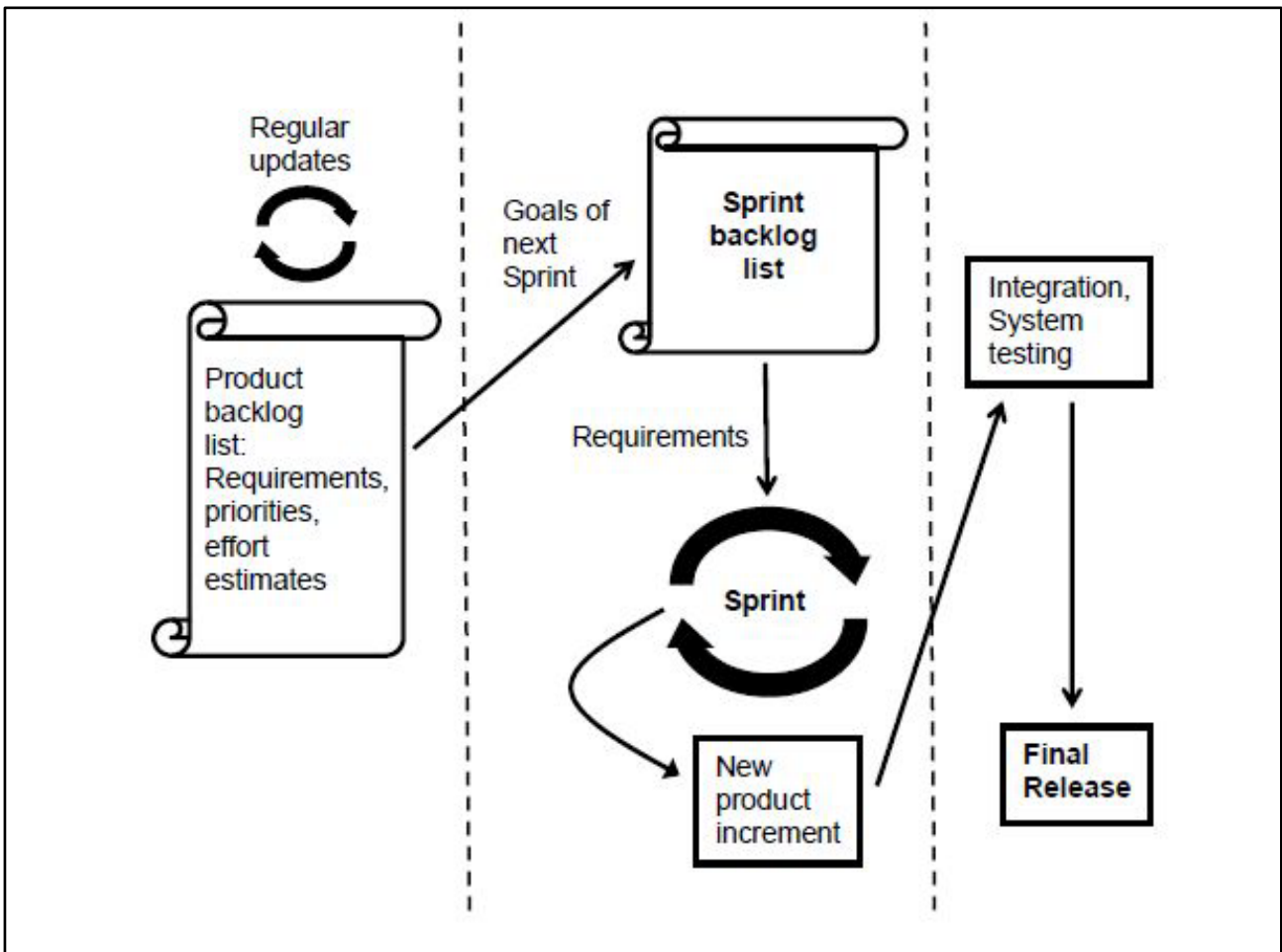


Figure 1. An Agile Model for Software Development Projects (Scrum).⁹

In the scrum framework, the product backlog contains a list of feature requirements for the software product, compiled by the customer. It is regularly updated in consultation with a customer representative in agile team meetings. Each feature requirement in the backlog has a priority level and an estimate of development effort. A sprint is a period of software development (i.e. actual programming activity), which usually lasts for a couple of weeks. The product features completed during each sprint are released to the customer for approval, after which the approved features are integrated into the product increment. A product increment is a collection of completed features that can already perform some meaningful tasks and operations. The agile approach favours building a minimum viable product, which is a software product that has enough features to start producing value and amassing feedback for the customer, as soon as is technically possible during the project. Finally, after several iterations of sprints, the final product is completed and ready for release.

⁹ Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta, *Agile Software Development Methods. Review and Analysis* (2002).

§18.04 The Impact of the Agile Development Model on Project Management and Contracting

While agile methods have worked successfully in smaller teams and in-house development projects, there have been some challenges in scaling them up for use in large corporations and multi-stakeholder projects.¹⁰ There are some complications associated with this, especially in the area of project management and project contracting.

The project contracting process is often illustrated as an activity-based workflow (Figure 2). In traditional industries, it is generally considered important that the customer clearly specifies what is going to be developed and delivered in the project at the outset. Contractors use this detailed information to put together their bids for the project. The project contract is then signed following additional negotiations with the contractor who is selected. After signing the contract, the contractor becomes responsible for the implementation of the project until the product is ready for delivery and ownership can be transferred to the customer. Both the customer and the project managers monitor and control the project's progress to ensure that everything is going according to plan.



Figure 2. The Project Contracting Process.

Agile methods, however, call for a different type of implementation, illustrated in Figure 3. Traditionally, the implementation phase of a project is isolated to ensure minimal disturbance for those involved. Agile practices, however, enable—and actually encourage—constant re-negotiation of the content of the software system under development.¹¹ This approach creates new and unique challenges for corporate lawyers, as in agile projects a signed contract does not signal the end of negotiations, which continue during the implementation phase. It is usually technical content and features that are renegotiated. According to agile principles, the customer has the authority to decide

¹⁰ Barry W. Boehm & Richard Turner, *Management Challenges to Implementing Agile Processes in Traditional Development Organizations*, 22 *IEEE Software* 30-39 n.5 (2005); Mikael Lindvall, Dirk Muthig, Aldo Dagnino, Christina Wallin, Michael Stupperich, David Kiefer, John May & Tuomo Kätkönen, *Agile Software Development in Large Organizations*, 37 *Computer* 26-34 n. 12 (2004); Saonee Sarker & Suprateek Sarker, *Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context*, 20 *Information Systems Research* 440-461 n.3 (2009); Jouko Nuottila, Kirsi Aaltonen & Jaakko Kujala, *Challenges of Adopting Agile Methods in a Public Organization*, 4 *International Journal of Information Systems and Project Management* 65-85 n.3 (2016).

¹¹ Jaakko Kujala, Soili Nystén-Haarala & Jouko Nuottila, *Flexible Contracting in Project Business*, 8 *International Journal of Managing Projects in Business* 92-106 n.1 (2015).

on the content of a software system, but this authority and the renegotiation that it entails have commercial implications. Agile contracts should support the principles of agile methods and enable cooperation and renegotiation. As the details of a commercial transaction cannot be left open in a contract, however, an agile contract should cover the commercial framework and the pricing scheme that are to be applied to a project to facilitate the use of agile methods. To avoid any disagreements over whether a product or feature has been completed to an acceptable level, agile contracts should also include explicit descriptions of and criteria for project completion for customer approval. In agile projects, this is often referred to as a ‘definition of done’.

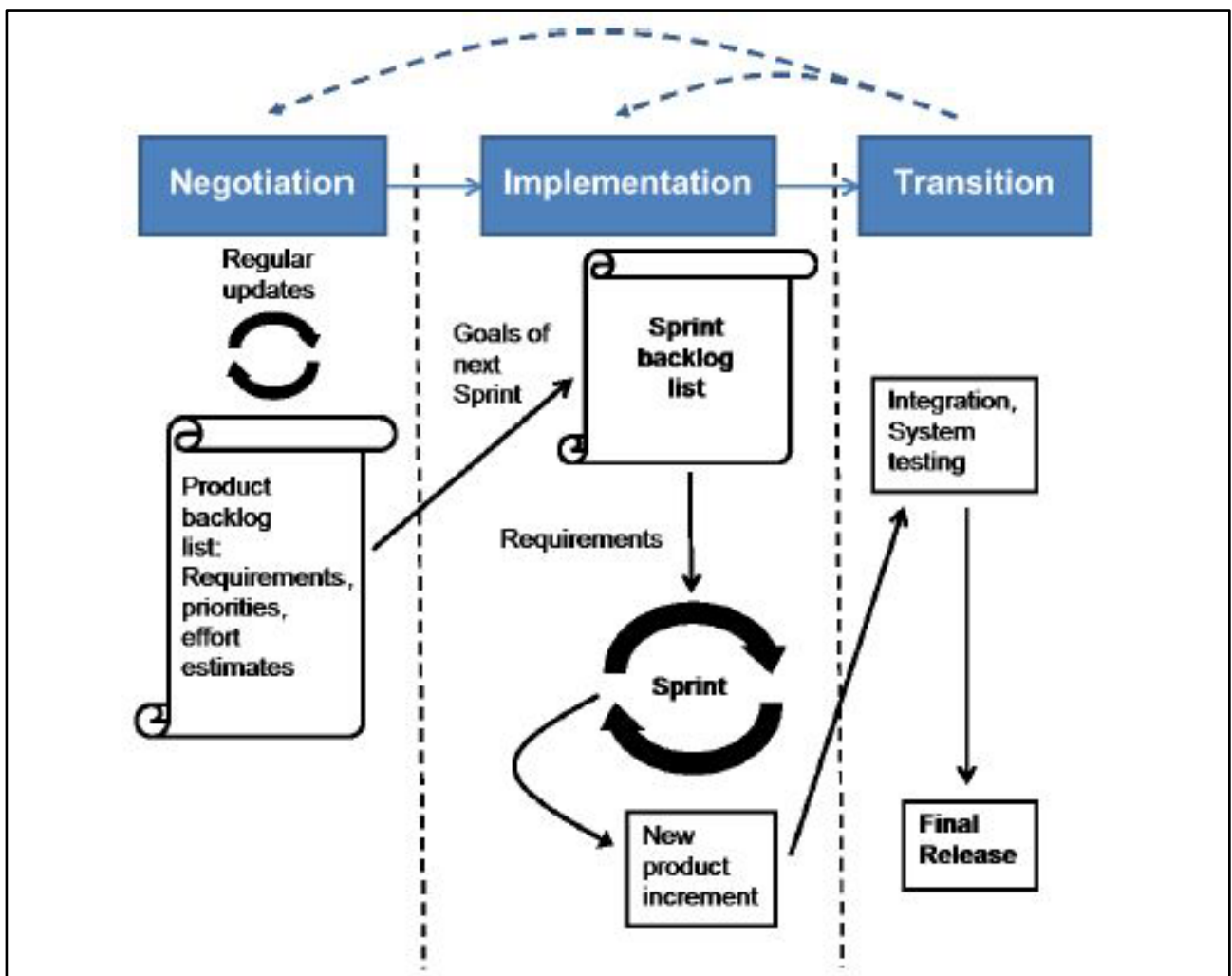


Figure 3. Agile Methods Favour a Cycle of Renegotiation.

Agile methods can increase flexibility in software projects. They facilitate collaborative learning during the development process and enable adjustments to be made to products based on the new knowledge obtained. This is especially important for software projects that create new innovations. Applications designed for the industrial internet are usually integrated into existing systems, to

which they contribute new features and functionality. Due to the complex nature of the industrial internet, new software development projects tend to encounter technological uncertainties at first. The flexibility offered by agile methods facilitates the management of these uncertainties. However, in order for software development projects to benefit from this flexibility, project contracting and project management frameworks must be adjusted accordingly. It is also important for business lawyers to understand that the agile approach lengthens the negotiation phase of software projects. Some other factors that need to be taken into consideration when working with agile contracts will be covered below in more detail.

The project contracting process depicted in Figure 2 represents how contracts are traditionally seen in contract law. Contracts are signed after a period of negotiation and then implemented. The upper part of Figure 3, which depicts the approach taken by agile methods, also illustrates how agreements are often made in long-lasting projects. In such projects, the need to renegotiate or find solutions that differ from those originally planned will sometimes occur. In contract law, such changes are the exception, but in contemporary business, they are the rule.¹² Agile methods are an extreme example of a process of constant renegotiation. The proactive approach to law has taken note of contracting practices in business and, in a similar manner to agile methods, it approaches contracting as a process in which the signing of the original contract is only one phase. The proactive approach suggests that contracts should be designed to match the business models of the parties involved in a contractual agreement.¹³

§18.05 Agile Contracts in Detail

As discussed earlier, adopting agile practices in software projects can increase efficiency. However, organizations starting to use agile methods for the first time might also face challenges when implementing them. Some of these challenges can be overcome by adopting an agile-specific contracting approach and using agile contracts.¹⁴

¹² Soili Nysten-Haarala, 'Why Does Contract Law not Recognize Life-Cycle Business? Mapping of Challenges for Future Empirical Research' in Soili Nysten-Haarala (ed.), *Corporate Contracting Capabilities* 18, 27 (2008).

¹³ Helena Haapio, *An Ounce of Prevention... – Proactive Legal Care for Corporate Contracting Success*, 2007 JFT 39-68 n.1 (2007).

¹⁴ Nuottila, Aaltonen & Kujala, *supra* n. 9.

Documentation. One of the principles of the Agile Manifesto is ‘working software over comprehensive documentation’.¹⁵ This statement does not mean that software projects should have no documentation at all, but that it should be kept to a minimum, as it does not add value to a project. Agile methods were first created for small teams working independently. Agreements can be reached in such teams during face-to-face meetings, since the whole team is likely to be working in the same location. In larger development projects, avoiding excessive documentation, which has no real value, is still a good target, but as there are likely to be several organizations or teams involved, some basic documentation needs to be in place. There are different requirements for documentation in a small team compared to a large team operating in a more complex environment. Lawyers often see documentation mainly as a proof of what has been agreed upon in a project for use in case of disputes, but documentation also functions as a map for the teams participating in the development process.¹⁶ Documentation can be used to monitor a project’s development. In agile projects, the focus on project monitoring more usually takes the form of visualization, testing, sprints, and iteration reviews, while the attitude to documentation is more flexible. In a complex environment, however, documentation can be a tool for enabling cooperation, not hindering it. Finding the right balance with documentation is important, and it is advisable that the parties involved in a project agree upon the amount and role of documentation in their agile contract.

Commitment. The waterfall method, which is a traditional software development method, relies on a well-defined process. The requirements and specifications of the software product to be developed are defined before programming starts. Agile methods, on the other hand, rely on the idea that skilful, innovative individuals and teams are able to identify problems during development and solve them together. In agile approaches, continuous communication is considered crucial to efficiently utilizing team members’ strengths. As such, it is important that an agile contract cover agile meetings and the need for continuous communication. Contracting parties need to commit to agile meetings and to delivering regular information and contributions as standard. This is an important point, especially if an organization has no prior experience of agile methods.

To organizations where agile methods are new, it may come as a surprise that the customer, too, must commit to constant communication and cooperation rather than simply waiting for product delivery. This commitment is an investment that requires the allocation of both human resources

¹⁵ Beck *et al.*, *supra* n. 6.

¹⁶ Soili Nysten-Haarala, ‘CCC – Contracting Capabilities in Industrial Life-Cycle and Service Business’, in Hanna Lehtimäki (ed.), *Yearbook of the Liito Programme 2008* 93-99 (2008).

and time to a project. A commitment to communication is also an important starting point for developing trust in the team and maintaining it throughout the development process. Open communication is not possible without mutual trust. Often, both lawyers and managers regard contracts primarily as weapons to be wielded in potential disputes, making sure that they contain obligations and take an adversarial approach to the interests of the contracting parties.¹⁷ However, when a contract is regarded instead as a road map or a tool for cooperation, as it is in the proactive approach, it can be used to document the crucial elements of agile development from the beginning of the process. A mechanism for effectively coordinating and managing changes to a project is one such crucial element. Furthermore, it should be stressed that contracts, which aim to foster cooperation, also need to express trust and a commitment to joint objectives, which is not self-evident.¹⁸

Stakeholder communication.¹⁹ Applications developed for the industrial internet and the IoT are typically designed to be integrated with several existing systems, or with systems developed in parallel. This integration creates interdependencies between several software systems, meaning that there are multiple stakeholders involved in most IoT software development projects. Agile planning is usually done iteratively, sprint by sprint. As such, it is possible that design choices that were agreed at the outset of a project will need to be changed as the project progresses. Stakeholders may not be informed of changes that will affect them and their systems until quite late in the process, which can pose challenges. It is good practice, therefore, to identify interdependencies between software systems in agile contracts and to anticipate challenging situations early enough to allow sufficient time for all stakeholders to make the changes needed. This is important, as based on the principles of agile planning, the interfaces between systems may not be defined far enough in advance to be able to communicate this information to stakeholders as early as necessary.

Role definitions. Agile methods rely on skilful individuals working as a team. It is important that all the roles in an agile team are occupied by individuals who are familiar with and committed to agile practices.²⁰ To ensure smooth and efficient cooperation between team members, agile

¹⁷ Jouko Nuottila, Osmo Kauppila, & Soili Nysten-Haarala, *Proactive Contracting: Emerging Changes in Attitudes Toward Project Contracts and Attitudes on Lawyers' Contribution*, 2 *Journal of Strategic Contracting and Negotiation* 150-165 nos 1-2 (2016).

¹⁸ Soili Nysten-Haarala, Nari Lee & Jukka Lehto, *Flexibility in Contract Terms and Contracting Processes*, 3 *International Journal of Managing Projects in Business* 462-478 n.3 (2010).

¹⁹ Nuottila, Aaltonen & Kujala, *supra* n. 9.

²⁰ Jeffrey A. Livermore, *Factors that Significantly Impact the Implementation of an Agile Software Development Methodology*, 3 *Journal of Software* 31-36 n.4 (2008).

contracts should clearly define the roles involved in a project. Agile methods require teamwork. An agile team is autonomous; it possesses a lot of decision-making power and holds joint responsibility for the project on which it is working. Agile team members are constantly developing their capabilities, supporting the self-organization of the team. The project manager in an agile team works more as a facilitator or a coach, whose main role is to support the other team members. Another key role in an agile team is that of the product owner, who represents the voice of the customer in the project. The product owner defines the product's feature requirements and identifies the requirements to be implemented in each development sprint. The team should constantly involve the customer (product owner) in decision making. The team is also responsible for creating and strengthening the customer's capabilities. This development of customer capabilities is crucial to the success of agile projects. The team is also responsible for monitoring the project. Sprints and iteration reviews that include feedback from the customer are utilized in monitoring. Visualization and product testing are also emphasized when monitoring the project. It is important for the lawyers who design an agile contract to understand the individual roles within an agile team so that they can define them correctly.²¹

Incentives. Incentives are needed to encourage teams and businesses to commit to agile methods. The literature on agile methods emphasizes that team members who have internalised the agile philosophy will have the most incentive to commit to a project's aims. Respect for other team members and the potential to interact with customers and take on decision-making responsibilities further encourage commitment from team members. Lawyers and managers recognize monetary inducements and the sharing of risk as incentives for both cooperation and the transaction of business in general. There is almost no research on the effectiveness of such incentives in agile development projects, however, though lawyers seem to offer ordinary risk-sharing solutions, adopted from contracts governing the sale of goods, in agile contracts, too.²²

§18.06 The Future of Agile Contracts

Agile methods are in wide use in software development. They are used in projects that have been subcontracted to facilitate the delivery of certain software components to a larger system. They are also used in situations where a whole third-party development team is hired to complete a particular

²¹ Soili Nysten-Haarala, Jaakko Kujala, Kirsi Aaltonen, Ketterät Menetelmät Julkisissa Hankinnoissa JIT 2015 Vakioehtojen Valossa (Feb. 15, 2019) (unpublished manuscript) (on file with author).

²² *Ibid.*

project. As discussed earlier, in the agile approach, the features of the product or component under development are not specified in detail at the beginning of a project. This approach raises questions about the optimal pricing scheme for work commissioned from third-party organisations.

Currently, most agile contracts are based on a framework agreement between the contracting parties, with additional amendments used to define details. Subcontracted work is usually paid by the hour at rates that may vary between the individuals involved in a project based on their roles and experience. Sometimes, a buyer will also set a maximum cost that the project must not exceed. This kind of pricing scheme has the disadvantage that the exact features of a finished product cannot be confirmed beforehand, since they will be partially cost-dependent.²³ Setting exact budgets for development activities can also be challenging. It has further been claimed that this contractual model decreases efficiency. Furthermore, there are currently no well-established models for incentivising individuals or organizations in agile projects.

One promising model that could provide solutions to these challenges in agile contracting is the alliance model, which is a relational project delivery model. Project alliances have led to several success stories in the infrastructure and construction industries.²⁴ A project alliance is formed by all of the parties involved in a project, who sign a contract agreeing on the project's target and goals. In this contract, they agree to share information related to the project and commit to joint decision making. The commercial contract also ensures that all parties have incentives to commit to the project: if the project's budget or schedule is exceeded, the monetary loss is shared between all the members of the alliance. If the project manages to achieve its target or to keep costs under budget, however, the members also share in the resulting benefits.²⁵

The project alliance model was recently used in a public sector ICT project.²⁶ In this project, a transparent alliance agreement was signed by all participants. This agreement was used in 'forming a collaborative multi-party contract that includes equal incentives and risks, and enables transparent open-book financials and communication'.²⁷ The alliance model could represent the future of agile

²³ Bruce Eckfeldt, Rex Madden & John Horowitz, *Selling Agile: Target-Cost Contracts* (2005) (Conference paper, Agile Development Conference).

²⁴ Anna-Maija Hietajärvi, *Capabilities for Managing Project Alliances* (2017) (Ph.D. dissertation, University of Oulu).

²⁵ See, for instance, Derek H.T. Walker & Beverley M. Lloyd-Walker, *Collaborative Project Procurement Arrangements* (2015).

²⁶ Teemu Mikael Lappi, Kirsi Aaltonen & Jaakko Kujala, *The Birth of an ICT Project Alliance*. *International Journal of Managing Projects in Business* (2018), available at: <https://doi.org/10.1108/IJMPB-02-2018-0036> (Feb. 18, 2019).

²⁷ *Ibid.*

contracts, as it allows them to further facilitate collaboration and to introduce efficient incentive schemes.

§18.07 Conclusion

The industrial internet and the IoT will increase the importance of software and the applications of embedded software in many industries. Learning to manage software development, software projects, and contracting will be key to building the success of software projects in the future. Agile project management is an emerging field of research. Many traditional project management frameworks are currently undergoing adjustment in order to suit them to agile software development models, thereby enhancing flexibility and efficient cooperation between project parties.

Agile principles challenge the conventions not only of project management, but also of contracting and contracts. The traditional approach to contracting, which emphasizes safeguarding and control, does not facilitate collaborative trust and collaborative innovation. It is important to develop a contracting approach that supports efficient models for managing software projects. In this chapter, a brief history of agile methods was presented to explain how software projects first started to differ from traditional industrial projects. In addition, several examples were presented to illustrate the aspects of agile contracts that need further development. A proactive approach to contracting can help in designing contracts that reflect agile methods. By taking a proactive approach, contracts can serve as road maps and tools for cooperation for agile teams.