

---

## AGILE IN COMMERCIAL SETUP – MISSION IMPOSSIBLE?

**Jurij Matyskevic<sup>1</sup>**

<sup>1</sup>*Mykolas Romeris University  
Vilnius, Lithuania  
Tel. +370 640 63798,  
E-mail: [mediajurijlt@gmail.com](mailto:mediajurijlt@gmail.com)*

**Zaneta Simanaviciene<sup>2</sup>**

<sup>2</sup>*Mykolas Romeris University  
Vilnius, Lithuania  
Tel. +370 68751813,  
E-mail: [zasiman@mruni.eu](mailto:zasiman@mruni.eu)*

DOI: 10.13165/PSPO-20-25-24

---

**Abstract.** This article reviews applicability of the very trendy in IT world Agile framework or even mindset and culture (a set of practice that in contradiction to other product and project management practises treats predominantly IT development in very flexible and iterative manner). Research starts with the very foundations of Agile practise, revises its main building blocks (principles) mentioned in primary source – Agile manifesto and examines validity of such principles to be implemented in commercial and business setup, e.g. in commercial relationships between different commercial undertakings.

**Justification of the research.** Agile and agility is very popular nowadays in mainly IT world however due to its nature it is very close to LEAN principles being implemented in manufacturing operations. The main idea behind that mindset is that products have to be developed in iterative mode – closely communicating with the customers and delivering working pieces of functionality as soon as possible (MVP – Minimum Viable Product). Such approach without any doubts is perfectly suitable for internal R&D projects (developing absolutely new products, brainstorming together new ideas), however its full application is doubtful in commercial setup due to the number of reasons.

**Aim** is to examine the applicability of Agile framework in commercial setup, pointing out what and why Agile principles work, and what parts of this mindset can be ignored as non-applicable.

**Methods** employed are analysis of the scientific articles, analysis of best practice use cases, comparison, analytical descriptive and generalization methods.

**Findings** of this research support the assumption that through leveraged Agile practise commercial companies can essentially benefit in designing product's quality and speed (time to market). However some of Agile principles shall be adapted to the nature of commercial relationships by introducing more formal approach to the extent possible (without jeopardizing Agile principles).

**Conclusions** emphasize positive Agility features and recommend to imlement such practises, introducing additional key elements that would ensure formal contractual relationships between commercial undertakings.

**Keywords:** Blockchain, Sustainability, Technology application.

**Jel codes:** O31, O32

---

## INTRODUCTION

Agile is a very popular Product development framework, which has been emerged in IT society (to be more specific in Software development, as even main source of Agile principles was named as “Manifesto for Agile Software Development”) and predominantly still heavily being used in IT industry. It’s also is worth mentioning that this usage is not limited by only software, but there are certain trends that Agile goes mainstream in at least majority of IT offerings if not all. Set of Agile features (to be covered later in this research) allows to draw a conclusion that initially Agile as a framework has been developed to be used in internal R&D (Research and Development) projects.

As IT industry significantly contributes to global economy worldwide authors decided to examine suitability of Agile application in commercial setup, e.g. using this framework in commercial relationships between different commercial entities.

**Problem** is to search and select proper product/project management methodology that could accept current challenges in rapidly changing technology industry, taking into account global competition, increased demand to “time to market” KPI, demanding needs of end users and smooth delivery practice in commercial environment.

**Purpose** is to examine suitability of Agile framework’s application in commercial relationships taking into account problems mentioned above.

**Object** is Agile application in commercial Product/Project development setup.

Tasks:

- to review Agile principles that serve as Agile foundations with the aim of applicability in commercial environment;
- to compare Agile framework with it’s main competitor – Waterfall;
- to review and describe main features of Agile framework that allowed this practice to become so popular in at least Software development;
- to review requirements of commercial Product development;
- to find out and explain what Agile features and principals suits commercial setup completely and what additions and enhancements might be needed.

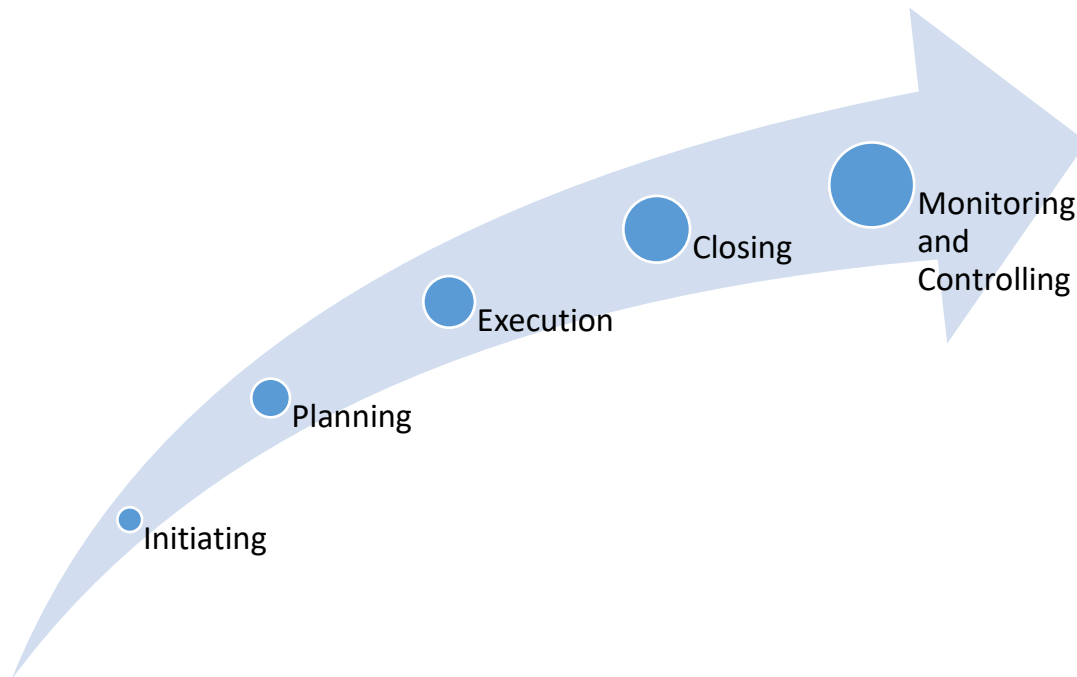
**Methodology of the Research** – analysis of the scientific articles, analysis of best practice use cases, comparison, analytical descriptive and generalization methods. It shall be noted that due to the topic’s novelty and practical application authors have put a lot of focus on practical sources.

---

## 1. AGILE FOUNDATION

Cambridge dictionary provides following meanings of Agile: able to move your body quickly and easily, able to think quickly and clearly, able to deal with new situations or changes quickly and successfully [5]. Of course in our context the latter meaning is most suitable, definitely such features as ability to deal with new situations or changes in efficient manner is very valuable asset both within internal and external development projects. Therefore Agile creation as a framework followed exactly that objective, as previous Project management methodologies (most common Waterfall) was unable to address efficiently increasing demand of functionalities. Markets are synonymous with rapid change and, as a result, commercial success or failure is largely determined by the organization's flexibility and responsiveness. Responsiveness is characterized by short time-to-market, the ability to scale up (or down) quickly and the rapid incorporation of consumer preferences into the design process [6]. Before explaining Agile framework it is worth explaining its main rival – Waterfall.

Waterfall is probably one of the earliest Project management methodologies which has started with the rise of first computer machines. The approach is logically sequenced, each phase follows previous one, deviation of this rule generally is not allowed. Therefore it comes with the name – natural flow as waterfall. Traditionally Waterfall is considered as best suited for large scale regulated projects due to its documented process.



**Figure 1.** Sequence of Waterfall phases  
*Source:* composed by authors

- **Initiating** is the phase, where essentially business case of the project, answering to the main questions such as who is going to finance the project (Sponsor), who will be affected by the project (Stakeholders) and what are actual benefits of the project (high level cost benefit analysis) shall be prepared and adopted. This information is contained in project manager's drafted Project charter (according to PMI) or Project concept definition statement (according to CompTia).
- **Planning** is the phase requiring most paperwork, as all activities shall be thoroughly planned, estimated and properly documented in project management plan (which consists of plenty of another sub plans such as Risk management plan, Change management plan, etc.)
- **Implementation** of technology is a real challenge which shall consist of development phase (installing Blockchain into actual IT infrastructure, merging it with current IT systems), testing phase (factory testing, functional testing and final E2E (End to End - acceptance testing) and deployment (with roll back scenario properly documented in above mentioned risk management plan).

---

- **Monitoring and controlling** is unique phase as it is the only phase which shall go along with others, meaning that performance and quality control shall be in place from the very beginning when it is possible to define relevant KPI's (Key Performance Indicators). Essential part of this activity shall be documented in Quality Assurance plan (part of general Project management plan).

- **By Closing** phase projects are typically finished, when all deliverables are accepted, lessons learned are documented and properly archived. One of the extremely essential points here – all paper documentation and also technical source code shall be indeed stored very properly, as IT technologies are constantly evolving, and it is highly likely that soon project will need to be adapted to the new requirements. Moreover current implementation needs being maintained – in order to do that DevOps (Development and Operations) teams must have access to proper documented outcomes of previous projects.

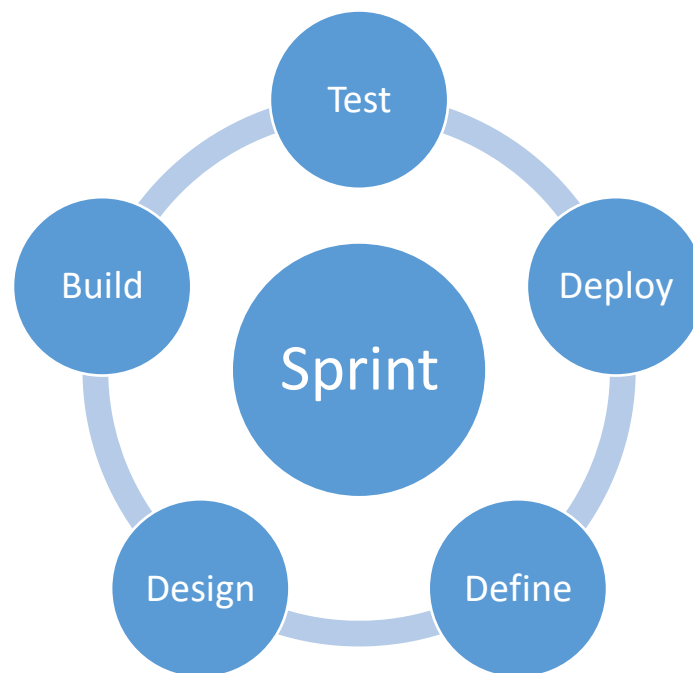
On the contrary Agile is not so strictly sequenced and it even is not a methodology, but rather a framework – a set of practice that in contradiction to Waterfall treats IT development in quite free manner. The Agile development values are particularly relevant to the implementation of a rule set using the Agile Business Rules Development approach [4]. This framework was born at 2001 by publishing Agile manifesto with following 4 values (Agile manifesto, 2001):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Another significant part of Agile Manifesto is twelve principles:

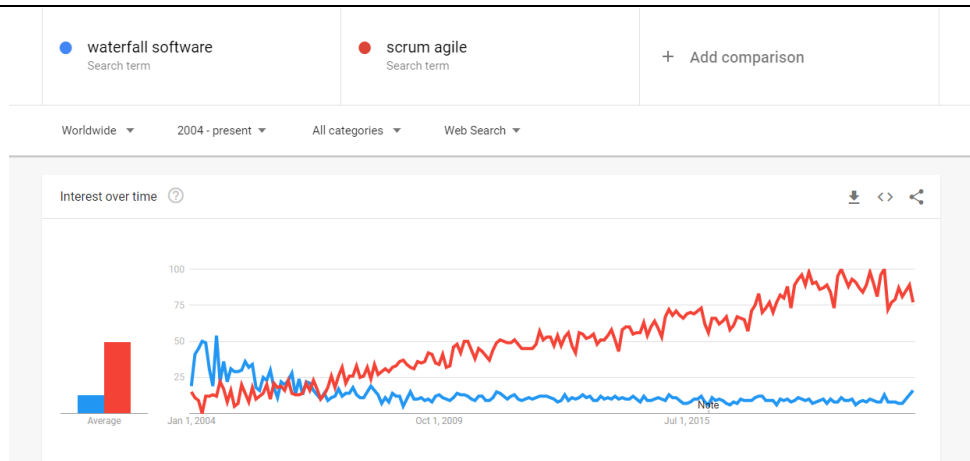
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

- 
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
  - Working software is the primary measure of progress.
  - Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
  - Continuous attention to technical excellence and good design enhances agility.
  - The best architectures, requirements, and designs emerge from self-organizing teams.
  - At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



**Figure 2.** Sprint within Agile framework  
Source: composed by authors

In practice poll below shows Agile domination over Waterfall:



**Figure 3.** Usage of Agile vs Waterfall  
*Source:* Google trends (2020)

As was already mentioned this practice is predominantly designed for software development but it even denies the concept of project – rather product development work conducted in time slots (sprints). Each sprint shall end with MVP (Minimum Viable Product) – deployable piece of functionality.

However having analyzed information above, authors still think that Agile is mainly internal product development practice focused on R&D activity due to the following reasons:

- Scope of future product is not clear, **collocated** and **dedicated** teams are brainstorming future product features together. When one thinks about agile software development and the challenging basic conditions of the business environment, one might think that nothing is fixed. Naturally this is particularly true of the scope. If we were able to describe it with sufficient accuracy in advance, then we probably would not consider anything other than a fixed price contract anyway [3];

- Roles within the team are not pre-defined, whole team is responsible for product entirely, all decisions shall be made by whole team;

- There is no Project manager role, Scrum master is a person only facilitating the process and removing impediments, no authority to accept or validate MVP;

- Teams shall consist of major product development stakeholders, that are building the product on daily basis together (including business units – in case of commercial setup – representatives from customer’s business units);

- Priorities are defined by Product owner (customer voice) however this role cannot assign more tasks/user stories on a sprint;

---

- Team is not committing for the MVP results (especially in the beginning of the actual work, as first sprints usually to realize and learn complexity of the product/user stories);

- Main sprint definition is not actual work done, it is simply time slot during which sprint team can commit just to be engaged in given product development.

Having said that let's review how main Agile basis – values fit to commercial relationships.

## 2. AGILE VALUES IN COMMERCIAL SETUP

As authors already mentioned above, Agile basis is 4 values and 12 principles. Aim of this chapter is to examine suitability of 4 values to commercial setup (principles essentially derived from values, so in this context they might be ignored). Contemporary software development entails considerable innovation, discovery, change sensing, and change responding (Conboy, 2009; Vidgen & Wang, 2009) while addressing various people, project, process, and institutional factors such as project scope and size, stakeholder participation, resources, technology, and outsourcing (McLeod & MacDonell, 2011) [2]. Despite further findings authors can make a definite statement – whatever product development practice is (internal or external), organization shall be ready to Agile practices, it has first of all review its business processes and align them to fit in Agile framework. Have you ever played the role of business owner and found yourself between "a rock and a hard place" of organizational politics when prioritizing backlog features? The Agile Stream approach negates those politics by dedicating development teams to organizational units and allowing those teams to continue working, iteration after iteration, as long as they continue delivering business value.[10]. Last but not least, both management and employees need to re-build themselves to Agile culture, without these assumptions Agile won't work in either organization.

### A. INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS

This value implies that Product development is not just ordinary task for IT team, but also requires intensive commitment and engagement from all major business units to actually build the product (share insights on current status quo, bringing business-related requirements, compliance, demand for product roadmap, etc.) **on daily basis**. The rule discovery, analysis, and validation activities require active and efficient communication between the rule developer, subject matter experts (SME), and business users. Such processes are defined as lightly as



---

possible [4]. It goes without any sayings that in such a case Product has much higher probability to be very functional and successful however even in internal projects that might bring an issue, as business unit people have their direct duties to perform, as this is their primary objective at given role, and respectively managers shall plan the load for business units accordingly.

The suggestion here for commercial Product development would be that organizations-customers shall precisely assess the cost benefit analysis in the very initiating phase and decide how much it can spend working hours of their best employees on system development, which in case is much better developed will get back ROI at much greater level covering both financial and labor costs. In 2004 PMI (Project Management Institute) found out that project initiation is the main reason why projects fail [12]. Moreover the same study claims that only 56 percent of strategic initiatives meet their original goals and business intent. This poor performance results in organizations losing US\$109 million for every US\$1 billion invested in projects and programs. Other sources report other major losses: For example, British food retailer Sainsbury had to write off its \$526 million investment in an automated supply-chain management system.

Systems is that they are too rigid to cope with changing business demands, especially for long running BPs. A solution to overcome this problem is to combine BPs with business rules (BR) [9]. As for final statement to authors' opinion wording over doesn't eliminate completely processes and tools. Giving priority to live interactions both counterparties (customer and provider) could first of all begin these interactions according to the processes that have built predictability and environment for such interactions, and later conducting and fixing the results of such interactions in appropriate tools.

## **B. WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION**

At first look this value sounds impeccably, as indeed customer seeks for ultimate result. Each iteration produces a working, tested set of rules that can be executed, which has far more business value than a rule description manual. While all project stakeholders benefit from such a principle, business users in particular are then sure that what they see (the rules, the business process) is what gets executed in the deployed system [3]. However valid questions would be:

- As IT systems need constant maintenance and upgrade, what happens when initial product's developers left the company, and there is no relevant documentation of the Product?

- 
- Same question extremely valid in commercial product development, how company can change provider (due to number of reasons) if new vendor won't get any knowledge transfer on the Product?

As example the U.S. Federal Aviation Administration spent \$2.6 billion unsuccessfully trying to upgrade its air traffic control system in the 1990s [1]. Practical solution that authors suggest is once product is enough mature, e.g. MVP passed acceptance testing and this functionality is deployed in production environment to fix that code in technical specs. The approach would allow during design and development phase to focus main effort on working software, and after to fix that success in technical specs.

Our results show that iterative documentation practices led to more extensive and more detailed textual documentation. We found that writing documentation was perceived as a intrusive task leading to task specialization and allocation of documentation to less qualified team members. Consequently, this hampered collaboration within the team. Based in our findings, we suggest that if documentation is to be delivered with the project, producing documentation should be communicated and accepted by the team as a proper product [13].

Agile also shall be applied maintaining current company's licenses and certificates, that are based on proper documentation. Conventional thinking would conclude that agile and ISO must not be compatible. After all, ISO is often characterized as being heavy on process / heavy on documentation - the opposite of agile. Just as the assumption that agile is about no documentation is faulty, so are the assumptions that ISO needs to be a burdensome process. ISO 9001:2000 is not only compatible with agile, but can provide just enough structure to help ensure your agile processes are followed [11].

### **C. CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION**

Since Roman empire (Roman law) and till today commercial relationships are being performed by concluding legally binding documents – contracts. Contracts are society's programming language. Corporations are defined by contracts with investors, employees, customers, etc. Countries are defined by social contracts with citizens, representatives, corporations, etc. [14]. And this field also constantly evolves and develops, as we can face emergence of new generation, digital and smart contracts based on blockchain technology. First of all speaking about full blockchain contract we have to highlight that this is purely digital – so called „smart contracts“ that means coded with software. Because blockchain transactions

---

are programmable and self-enforcing, parties might use smart contracts to design contractual relationships that are automatically executed without the additional costs of monitoring or enforcement [15].

Contracts serve not only as “source of truth” and kind of guarantee for scope and nature of services to be performed by vendor and legitimate ground for money movements, but also as proof of legitimate business practice for state regulation authorities such as tax inspection. That means that counterparts could not totally avoid contractual phase of their relationships. Solution that could be offered but authors consists of several options. At first usage to maximum possible extent automated “smart” contracts based on blockchain, so that majority of standard clauses are cover by machines and parties can focus on specific provisions. Second usage of so called “framework agreements” whereas standard conditions could be negotiated once, and after that parties only negotiates on specific provisions (even double synergy can be achieved by automating smart contracts). Last but not least purely legal argument - contracts can be concluded orally, in writing or by conclusive actions [7].

The latter possibility seems very attractive as in this case let’s say start of development effort can be interpreted as already concluded contract however it shall be not misused, as legislation puts additional limitations on such way on conclusion in terms of value, specific kinds of the services, etc. In practice there are studies that align business contracts to Agile. We report on our experiences from two commercial projects conducted under a new form of contract that supports agile development and encourages efficient collaboration between customer and supplier. We show actual extracts from our contracts and describe how to adjust a contract to specific project conditions[16]. Specific Agile related advantage of this value is that Subject matter experts who define the business policies and the business rules are strongly involved in the development process. As the customers of the final system and owners of the policies, they are conveniently collocated with the development team during the project. There is no specification document thrown above a wall waiting for the IT to develop the system [9].

#### **D. RESPONDING TO CHANGE OVER FOLLOWING A PLAN**

This value probably contradicts to classical “waterfall” methodology most, as Agile mindset sets priorities only within Product backlog and separate sprints only without comprehensive planning. The failure of plan-driven waterfall-based methods in software projects experiencing significant uncertainty and frequent changes in requirements prompted

---

the adoption of the Agile Manifesto (Highsmith & Cockburn, 2001) and the proposal of several agile methods such as Scrum (Schwaber, 2004) and XP (Beck, 2000) [5].

Business rules evolve more often and faster than other standard pieces of software. This is actually one of the key values of the business rule approach. For this fundamental reason, the methodology to support the rule set development must be tailored to such rapid life cycle and include the appropriate activities, processes, best practices, and work products to support such changes efficiently[9]. Main idea behind is that circumstances such as user experience and demand, new technologies, new priorities shall be taken into account “on the run” as due to the speed and variety it is not possible to estimate it in thorough planning.

The only counterargument from the authors is that company-customer always performs planning of its business and of course has insights of timeline when product shall be launched. So inevitably high level planning of product release (and of course including respectful development and testing activities) must be in place. In the same time authors agree that given both customer and supplier are agile organizations and product development is agile-aligned (e.g. stakeholders actively participate in sprint activities) change management activities could be performed very efficiently.

Our goal has been to meet our customerpsilas needs for predictable results while maintaining our commitment to agile practices. Key success factors include developing a responsive contract modification process that allows for quick change management and

## CONCLUSIONS

Authors have performed research on Agile suitability for commercial Product development e.g. between separate commercial units that cooperate on contractual basis.

First the basis of Agile framework has been analyzed – demand behind emergence, initial suitability for certain kinds of projects, comparison to other worldwide methodology - waterfall. It has been defined that Agile product (Agile doesn’t use project terminology highlighting its focus not on processes but rather on value) management methodology is based on Agile manifesto containing 4 values and 12 principles.

Suitability for commercial setup research has been performed mainly analyzing 4 values, as 12 principles to authors opinion are auxiliary, as they derive from abovementioned values.

Individuals and interactions over processes and tools value can be used in commercial setup if two preconditions are met. First in the very beginning during product initiating activity

---

company-customer shall very precisely foresee man/hours labor budget of its employees participating on daily basis in sprint activities jointly with vendor's teams. Such commitment from the customer can be explained by much greater ROI achieved by such participation. Second – high level processes and tool still have to exist first of all creating environment for such close interactions and fixing results in appropriate tools.

Working software over comprehensive documentation value applies in commercial setup only under condition that once working software deployed in production environment respectful technical specification has been created. Failure to comply with this rule might lead to non-ability to maintain, update and develop systems in the future.

Customer collaboration over contractual negotiations value in authors' opinion might be mitigated by several assumptions. Given the fact that contracts are still required, first option would be to automate it on the basis of Blockchain technology, so that teams focus only on very specific several clauses instead of negotiating all contract text over and over again. Second option is framework contracts that provide similar benefit, e.g. standard provisions being accepted once without re-visiting it on every project. Final option which shall not be misconducted is pure legal "game" to conclude the contract by conclusive actions.

Final value Responding to change over following a plan implies controversial balance between flexibility and predictability. Authors recognizing flexibility of such approach in change management are still of the opinion that at least high level plan containing main milestone such as at least commercial launch shall be in place.

## Summary

**Actuality** – validity of the topic determined by the fact that on one hand Agile becoming (or in some industries already became) mainstream framework for product development. As a lot of such development is being performed between separate commercial units on contractual basis, it's worth analyzing how main Agile principles apply to commercial relationships.

**Goals** - to examine suitability of Agile framework to commercial product development and consider how main Agile values could be aligned or enhanced to fit in commercial setup.

**Methods** - analysis of the scientific articles, analysis of best practice use cases, comparison, analytical descriptive and generalization methods. It shall be noted that due to the topic's novelty and practical application authors have put a lot of focus on practical sources.

**Results** – Author's opinion is that literal treatment of Agile values does not allow this methodology being implemented in commercial setup, mainly due to little guarantees to customer (ambiguous scope, little procedures, no firm commitment from the vendors, etc.) However this could be solved by interpreting Agile framework in more wider context.

---

## REFERENCES

1. Aotea. (2011). Aotea Business analysis insight. The impact of Poor requirements with a poster. Aotea.
2. Batra, D et. al. (2016), Agility Facilitators for Contemporary Software Development, Journal of Database Management (JDM) 27(1), p.28, doi:10.4018/JDM.2016010101
3. Book M., Gruhn V., Striemer R. (2016) Agile Contract Models. In: Tamed Agility. Springer, Cham. [https://doi.org/10.1007/978-3-319-41478-2\\_13](https://doi.org/10.1007/978-3-319-41478-2_13)
4. Boyer, J., Mili, H. (2011). Agile Business Rule Development. Springer-Verlag Berlin Heidelberg, doi:<https://doi-org.ezproxy.metropolia.fi/10.1007/978-3-642-19041-4>
5. Cambridge dictionary, [interactive]: <https://dictionary.cambridge.org/dictionary/english/agile> [accessed 2020-10-08].
6. Christopher, M., Lawson, R. and Peck, H. (2004), "Creating agile supply chains in the fashion industry", International Journal of Retail & Distribution Management, Vol. 32 No. 8, pp. 367-376. <https://doi.org/10.1108/09590550410546188>
7. Civil Code of the Republic of Lithuania, adopted by the Parliament of the Republic of Lithuania on 2000 July 18 d. No. VIII-1864 (1.71 article).
8. Franklin, T., "Adventures in Agile Contracting: Evolving from Time and Materials to Fixed Price, Fixed Scope Contracts," Agile 2008 Conference, Toronto, ON, 2008, pp. 269-273, doi: 10.1109/Agile.2008.88.
9. Graml, T., Bracht, R. and Spies, M., "Patterns of Business Rules to Enable Agile Business Processes," 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), Annapolis, MD, 2007, pp. 365-365, doi: 10.1109/EDOC.2007.35.
10. Hunter, Z. and Spann, D., "Creating Agile Streams for Business & Technical Value," Agile 2008 Conference, Toronto, ON, 2008, pp. 144-147, doi: 10.1109/Agile.2008.93.
11. McMichael, B. and Lombardi, M., "ISO 9001 and Agile Development," Agile 2007 (AGILE 2007), Washington, DC, 2007, pp. 262-265, doi: 10.1109/AGILE.2007.36
12. PMI (2014). PMI's pulse of the Profession: The High Costs of Low Performance. [interactive]. Access: <https://www.pmi.org/learning/thought-leadership/pulse/the-high-cost-of-low-performance-2014> [accessed 2020-02-07].
13. Stettina ,C. J., Heijstek, W. and Fægri, T. E., "Documentation Work in Agile Teams: The Role of Documentation Formalism in Achieving a Sustainable Practice," 2012 Agile Conference, Dallas, TX, 2012, pp. 31-40, doi: 10.1109/Agile.2012.7.
14. Omohundro, S. (2014). Cryptocurrencies, Smart Contracts, and Artificial Intelligence. Steve Omohundro.
15. Kiviat, T. (2015). Beyond Bitcoin: Issues in Regulating Blockchain Transactions. Duke Law Journal No. 65, p. 569-608.
16. Thorup, L. and Jensen, B., "Collaborative Agile Contracts," 2009 Agile Conference, Chicago, IL, 2009, pp. 195-200, doi: 10.1109/AGILE.2009.19