Abstract

Agile software development has been discussed in literature since the inception of the Manifesto for Agile Development. However, there is no unified perspective towards Agile development in the literature. Academics and practitioners see Agile methods as a philosophy, as collaboration and communication enhancer or as means to better business benefits. This study discusses the spectrum of perspectives that emerges when one analyses the literature on Agile methods and Agile Manifesto. The study presents example papers across the spectrum: literature that has taken an Avid perspective and treats Agile as a world-view, literature that emphasises Inclusive Agile methods and is not averse on method tailoring and literature which takes a Pragmatic perspective and seeks to maximise the benefits of Agile methods application.

Keywords

Agile software development, philosophy, Agile Manifesto

Introduction

Agile methods are rooted in the frustration of a group of software development practitioners who critiqued the ways software development projects were conducted in the corporate world and who had a desire to improve work and its outcomes (Beck et al. 2001). It is not accidental that the springboard for the Agile movement was named ‘Manifesto for Agile Software Development’ instead of something less politically charged. Today, in addition to the original 17 signees, the Manifesto has been signed by thousands of people who ascribe to the values and principles stated on the website where the Manifesto and the signatures can be viewed.

There are two parts of the Manifesto for Agile Software Development information systems development: the Agile values and Agile principles. The Manifesto declares the following: ‘We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation and Responding to change over following a plan.’ (Beck et al. 2001, emphasis in the original).

The second part of the Manifesto consists of 12 principles, which describe in more detail the ideas of the authors. The role of the developers as capable individuals, who should be trusted and not managed, is clearly and repeatedly stated. The Agile principles further stress the importance of working software and collaboration between all the project stakeholders. Harnessing change as an advantageous act, rather than seeing it as an annoyance for the work, is further emphasised in the principles. The authors of the principles proclaim that projects should be conducted at a sustainable pace that the team can maintain indefinitely. This means that overtime and weekend work, activities that projects often revert to when deadlines are pressing, are not acceptable.
The Agile values and principles targeted the software developer community with an emancipatory message. The authors of the Manifesto all have a background in software development and their experiences in the plan-driven environment of the software industry led them to test and design better ways to conduct software development. Highsmith (Beck et al. 2001) writes in regard to Agile practices that ‘practices define a developer community freed from the baggage of Dilbertesque corporations’. The Manifesto and collection of Agile software methods, modelled on the values of the Manifesto, propose a solution to the issues that have created such environments. In the Agile environment, the roles and distribution of responsibilities safeguards against micromanagement and other excessive manager control (Schwaber 2004).

Studies on organisations adopting Agile describe the methods as a game changer, a significant modification of the daily work that requires rethinking of the organisational culture (Strode et al. 2012). Agile methods and their use have ramifications, which extend beyond development teams (Martini et al. 2016). The need for support from management or other stakeholders is a continuing issue perceived by the Agile practitioners (ScrumAlliance 2015). However, Agile methods are becoming the norm and most organisations have adopted some form of Agile development (VersionOne 2017). But how do Agile practitioners and researchers who study Agile methods view the role of the Manifesto? Is the Manifesto seen as a philosophical foundation or something more practical? Now that the Agile methods are being adopted even outside software development (Rigby et al. 2016), an investigation on how the original ideas are being implemented nowadays is in place.

One would think that with such impact and widespread use of the methods, organisations applying Agile methods would have adopted similar views on the Agile Manifesto and Agile methods. However, our literature analysis, presented in this paper, reveals how differently the Agile Manifesto and the methods are viewed in the literature and how organisations and papers written on the subject, fall on a spectrum of Agile rather than a uniform interpretation of the Manifesto and the meaning of the methods.

**Agile Software Development in Literature**

The strong practitioner focus of the early writings on Agile and the ‘from developers to developers’ ideology behind the Manifesto can pose a challenge for a reviewer of the Agile literature. The literature discussing Agile development is fragmented and multidisciplinary (Dingsøyr et al. 2012) and there is an abundance of practitioner-oriented guides that detail practical application of the multitude of Agile methods (e.g. Agile series).

However, to address the question on how different views on Agile development might influence collaboration, we decided to analyse a selection of both practitioner-oriented and academic papers that could provide an overview of the theoretical views of Agile development methods and offer more insights on how Agile development methods are described in the literature. A more thorough analysis on how the papers see Agile and what is perceived as ‘proper’ application of Agile methods was relevant to our interests, because there was a distinct lack of theoretical understanding of Agile methods and unifying theory (Conboy 2009; Dingsøyr et al. 2012; Hummel 2014).

A starting point for our analysis were three extensive literature reviews by Dybå and Dingsøyr (2008), Dingsøyr et al. (2012) and Hummel (2014), which summarised the current body of knowledge on Agile development and guided us with our review and analysis. These literature reviews have distilled the most common research methods, most discussed topics and theories applied, and provide statistics on the state of the research. Two of these literature reviews (Dingsøyr et al. (2012) and Hummel (2014)) had provided their original search strings, which we adopted with a slight modification1 and performed on the Scopus search engine, limiting our search to journals in the field of Information Systems research and a selected set of Agile practitioner journals and conferences2.

---

1 Title OR Abstract:(agile OR agility OR extreme programming OR scrum OR pair programming NOT manufacturing) Keywords: ("Flexible Information Systems Development" OR "Agile Software Development" OR "Agile Information Systems Development")

2 Senior Scholars’ Basket of Journals (http://aisnet.org/?SeniorScholarBasket), selected IT journals such as Computer, Communications of the ACM, Proceedings of AGILE, IEEE Software and major AIS conferences
This search resulted in 788 documents. From this selection, we read the 100 most cited papers and skimmed through the next 200. In addition, we read the papers specifically cited in each of the literature reviews. Lastly, we organised the collection by date and skimmed through the 100 most recent publications, in order to include newer papers with novel perspectives towards Agile.

Our goal was not to conduct a literature review of every Agile paper ever written but to find whether there were differences in how the papers approached Agile development methods. Based on our initial reading of the papers, we created a diverse paper set which would become a starting point for our analysis. For this set, we applied a hermeneutical paper review approach, as described by Boell and Cecez-Kecmanovic (2014). First we summarised and categorised papers according to their topics; then in the second iteration, we delved deeper into the descriptions of Agile. Our goal was to analyse if there were different perspectives towards the Agile Manifesto and whether the authors were critical of the concept or the application of the methods, or if ‘agility’ of the projects was taken at a face value.

Based on our analysis, an Agile perspective continuum began to emerge. Papers written during different time periods had different purpose and different outlook on what Agile development was. The early guidebooks describing the popular methods, such as Scrum (Schwaber 2004), Extreme Programming (Beck 1999) or the Agile Ecosystem (Highsmith 2002), provide a window into the practical application and into the worldview of the prominent authors of the Agile Manifesto movement. These guidebooks and early papers, published in 'Computer' (e.g. Beck 1999; Cockburn & Highsmith 2001; Highsmith & Cockburn 2001; Boehm 2002), sparked the discussion on Agile in the development community and fostered the early Agile adoption in organisations.

These papers, describing what Agile was and how it should be applied, were aimed at the general public, developers and managers alike. Few papers included short examples of successful Agile implementation (Beck 1999); others relied on explaining the underlying philosophy of the values (Highsmith & Cockburn 2001). With a strong commitment to the newly minted Agile cause, the authors had an agenda and a mission to evangelise Agile into all organisations where software was being developed.

Once the methods became more widely applied, case studies of organisations utilising Agile began to emerge. A stream of research papers and case studies focused on the complexity of human interactions of the Agile teams in both collocated and globally distributed projects and virtual teams and their struggle with Agile methods (e.g. Holström et al. 2006; Sarkar & Sarkar 2009). Another set of papers investigated the application of certain types of Agile approaches, for example, Scrum or Extreme Programming, often backing their research with case data describing successful Agile projects (e.g. Sutherland et al. 2008).

Finally, in recent years, the Agile research has also focused on the business benefits of applying Agile methods, based on case studies in large organisations that have applied large-scale Agile projects (e.g. Martini et al. 2016). The following sections will analyse a selection of papers that present this spectrum of perspectives in the literature in more detail.

**Agile as Philosophy**

Unsurprisingly, the original signees of the Manifesto have been the most vocal advocates of application of Agile values and principles in their original form. Agile methods have a clear emphasis on the social aspect of the software development but also philosophical and ethical underpinnings. For example, Beck (1999, pp. 71) has named one of his methods as ‘metaphor’ and describes it as ‘the shape of the system is defined by a metaphor or set of metaphors shared between the customer and programmers’. In the same paper, he writes (citing Lakoff & Johnson (1999) and Coyne (1996, pp. 73)): ‘XP’s use of metaphors comes from George Lakoff and Mark Johnson’s books, the latest of which is Philosophy in the Flesh. It also comes from Richard Coyne, who links metaphor with software development from the perspective of postmodern philosophy’.

Ethics and philosophy are also addressed by Beck (1999), who writes that one of the influential works for XP was a book named ‘Peopleware’, by DeMarco and Lister (2013, first edition 1987), which drew attention to the human aspects of project development and the sociological rationale of issues in projects.

Highsmith and Cockburn (2001) and Cockburn and Highsmith (2001) describe the two sides of the Agile development, what they call the people factor and the innovation factor. In their two papers, Highsmith and Cockburn unpack the values and disciplines of the Manifesto from both a people and a business
perspective. They reiterate, ‘Agile organizations and agile managers understand that demanding certainty in the face of uncertainty is dysfunctional. Agile companies practice leadership-collaboration rather than command-control management’ and that ‘they understand that agility depends on trusting the individuals to apply their competency in effective ways’. These statements reflect the origins of the Manifesto. When it comes to perspective towards Agile methods and the rigour they should be applied with, Highsmith and Cockburn (2001) do not compromise. To them, Agile methods are the baseline that should not be cherry picked but built upon.

A more recent paper that particularly discusses the perils of ‘cherry picking’ practices without a proper understanding of Agile comes from O’hEocha et al. (2010). The authors warn against ‘agile assessments’ that are conducted to measure compliance with a method, rather than the context of the organisation and the issues. The paper stresses that the feedback loop is a fundamental concept in Agile, a concept that is often forgotten. The authors also note that in their research cases, only the development team was converted into Agile; the holistic approach encompassing the whole organisation was not the focus of their paper. Only more recently has there been more discussion on implementing Agile across different organisational functions; albeit the proponents of such extensions are academics who have been advocating Agile methods since the beginning (e.g. Sutherland & Takeuchi in Rigby et al. 2016).

Nevertheless, the organisational aspect has been the subject of analysis since the very inception of Agile. The common viewpoint has been the comparison between the more traditional and the Agile methods. Boehm’s (2002) comparison between the two opposite ends—plan-driven methods and Agile methods—describes what he calls a ‘sweet spot’, a perfect balance between traditional and Agile. Boehm sees that these methods are not incompatible and that neither represents a fringe but rather a ‘responsible centre’ for the development. According to Boehm, Agile methods can be applied to entice people away from tempting fringes, the method-free and control-free ‘cowboy coding’ environment.

In contrast, other authors think that there are more fundamental differences between plan-driven and Agile methods. For example, Nerur et al. (2005) address the differences between the ‘traditionalists’ and the ‘agilists’, declaring that the methods are grounded in different concepts. They see that ‘a rationalised, engineering-based approach, grounded in principles of hard systems thinking’ has been the dominant way. This perspective assumes that problems can be specified and there are always solutions, whereas Agile is the antithesis of this way of thinking.

Furthermore, Nerur and Balijepally (2007) speculate on how organisations could better solve so-called wicked problems with Agile methods, problems which traditional methods fail to address sufficiently. They proclaim that ‘emerging agile philosophy heralds a new epistemology of software development’, compare Agile to Action Learning Theory, and declare that Agile facilitates double loop learning and offers ‘an expansive metaphor of design and the theory of holographic organization’ as theoretical bases for Agile development. Unfortunately, even though the 2007 paper is well cited, the suggestions of the authors did not catch the attention of other researchers to an extent that would have produced a theoretical explanation based on these ideas (Hummel 2014). The follow-up discussion by Nerur et al. (2010) focuses more on the historical underpinnings of the phenomenon, rather than new theory building.

However, few other theoretical approaches have been applied. Vidgen and Wang (2009) have applied Complex Adaptive Systems (CAS) theory in the context of Agile development. They cite that a CAS viewpoint is appropriate as Highsmith and Cockburn (2001) refer to Agile as CAS: ‘One aspect of agile development is often missed or glossed over: a worldview that organisations are complex, adaptive systems’. The authors bring together Agile development and CAS concepts such as the ‘edge of chaos’ or ‘region of emergent complexity’; self-organisation, which is already addressed in one of the Agile principles; and the balanced innovation and knowledge creation and process and product improvements. Unsurprisingly, their two case study organisations have elements of Agile organisations but also elements that are unsupportive of the CAS view. Although Vidgen and Wang are not redefining Agile, they do provide a list of Agile team capabilities and organisational traits that act as Agile enablers and Agile inhibitors.

Theory creation has been a subject also for Conboy (2009), who provides one of the more often-cited definitions of Agile (pp. 338) ‘the continual readiness of an ISD method to rapidly or inherently, create change, proactively or reactively embrace change, and learn from change, through its internal
components and relationships with its environment’. Conboy bemoans the lack of clarity, theoretical glue, cumulative tradition and parsimony when it comes to Agile research.

Another example of a paper that clearly identifies its position on Agile and, in addition, elaborates on the Agile practices of its case study organisations comes from Strode et al. (2012), who discuss coordination of Agile teams in collocated premises. They begin by acknowledging the revolutionary nature of the Agile methods and proclaim that Agile development is a new paradigm in information systems development. They discuss the ramifications of Agile adoption; for example, changes in the organisation culture, roles, customer involvement and team interactions. They also discuss the Manifesto as a unifying philosophy that ties together Agile methods.

Introduction of new methods such as Kanban or Lean has driven researchers to re-examine the definitions of Agile. Wang et al. (2012) have analysed Lean (Poppendieck & Poppendieck 2003) against the values of Agile and conclude by encouraging further examination of these two concepts on a more detailed and operational level. The authors of the ‘Leagile’ paper see both Lean and Agile as compatible ways of thinking, similar and intertwined. The mindset aspect, understanding Agile as a philosophy with the willingness to allow compromises, makes this paper an excellent bridge between different schools of thought and perspectives towards Agile.

**Agile in a Constrained Environment**

Where some organisations and authors hold to the ideal Agile application and emphasise the philosophical nature of Agile (for example Strode et al. 2012), other organisations and authors see Agile development methods as a means to an end, as a way to enable efficient collaboration and communication rather than a strict rule that has to be observed. The literature that discusses case studies based on empirical findings often indicates that the Agile values and principles are not forgotten in the ‘real world’, but that when it comes to implementation of Agile methods, compromises are acceptable. The ideas of the Manifesto are still accounted for, but rather than transforming the organisations to fit the method, the methods are selected and applied however the organisations deem beneficial, to achieve better communication and collaboration. Agile methods are seen as facilitators of collaboration rather than a philosophy or mindset.

Such a perspective is especially prevalent in papers that discuss partially or fully virtually conducted and globally distributed software development (Sarker & Sarker 2009; Modi et al. 2013) or larger-scale projects (Sutherland et al. 2008). While these works detail the benefits of the Agile methodology, the focus is not on how best to implement the methods, but on how Agile can be used to make the development better in various circumstances. The practical constraints of the environment take precedence over the Agile principles.

By embracing this approach, the proponents of Agile methods have been able to overcome the original stigma of Agile: a view that Agile is only applicable for simple projects and smaller teams (Dybå & Dingsøyr 2008). For example, Manifesto signatory Jeff Sutherland describes how Scrum can be implemented in large organisations. In this description, Sutherland does not explore the values or the principles but matter-of-factly lists methods that work in the large-scale environment. He advocates applying the Scrum method in distributed teams as well as large teams, by scaling the method and combining it with the Capability Maturity Model Integrated model (Sutherland et al. 2008).

Sutherland’s claims are backed up by researchers who have shown that application of Scrum in large-scale, distributed projects, can be successfully applied with positive effects (Paasivaara et al. 2008). However, later work of Paasivaara et al. (2012) suggests that some organisations struggle with the application of the methods and fail to reap all the benefits that other organisations have achieved with their application of Agile. These two studies imply that Agile development methods have been applied outside the originally intended scope and must be tailored to fit the context of globally distributed development. The authors proceed to discuss Agile methods through the application of the suggested activities, rather than as a philosophy.

Similarly, Holmström et al. (2006) only briefly refer back to the values of Manifesto, but swiftly proceed to the application of methods to bridge different elements of distance in globally distributed teams. They conclude that even a very rudimentary use of a few Agile methods helps to reduce distances between team members and they advocate further research into the application of the methods in such teams.
In addition to global software development and control issues, the more flexible perspective of framing Agile as a matter-of-factual way of working is prevalent in case studies involving the customers and their role in projects. Where the virtual and distributed nature of projects present geographical and temporal constraints, the customer involvement can present a different set of constraints. Customer involvement, or the lack of it, poses challenges for the Agile development teams (Hoda et al. 2011). These constraints are discussed from different perspectives in papers ranging from ideal customers (Martin et al. 2016) to the role of onsite customers (Koskela & Abrahamsson 2004). A common thread across these papers is that sufficient customer involvements can be achieved, but not necessarily without some compromises in Agile method application.

The authors of literature discussing distributed virtual collaboration and customer involvement acknowledge to the readers that they are familiar with the origins of the methods, but refrain from a critique of the modifications applied by the case study organisations and the perceived level of ‘agility’ of the organisations. The application of Agile methods in the case of organisations studied in these papers is not contested. By adopting this ‘liberal’ perspective on Agile methods, the researchers who have conducted these studies can aid other organisations in their journey of Agile adoption and adaptation. The empirical findings have highlighted issues and benefits of Agile application in their case organisations and the results and suggestions are actionable by other organisations who wish to follow the advice.

However, one might think that these papers could have benefited from a more philosophical perspective and a discussion on whether the members of the organisations themselves subscribed to the values and principles of the Manifesto. The analysis in the papers lacks discussion on whether the results were due to a proper understanding of Agile values and principles or despite it. Granted, there have been no clear guidelines on how to evaluate ‘agility’ (O’hEocha et al. 2010) and single case studies lack comparison aspects that would reveal differences in the perspectives. Nevertheless, even though an Agile purist might contest the ‘agility’ of the research premises, the flexible approach towards the application of Agile methods does extend the reach of Agile beyond the ‘ideal’, small and collocated organisations.

**Agile as a Business Benefit Driver**

While the philosophical view of Agile is at one end of the spectrum of Agile views, the opposite end of the spectrum is formed from a perspective that sees Agile methods as a means to provide organisations with the readiness to change according to market conditions. This perspective does not disregard the perspective of Agile, which emphasises the philosophy or craftsmanship and team empowerment, but these factors are not seen as the primary drivers for Agile adoption.

An example of this perspective is provided in a paper discussing the adaptation of Agile by Cao et al. (2009), who assert, ‘we define adaptation as the process of changing agile methods to align them with the needs of different projects and organizational environments. This process involves adding, dropping, or modifying specific practices prescribed by agile methods’. While the authors who view Agile methods as a philosophy would rather change the organisation to fit the philosophy, the more business-benefit-oriented authors would rather change the methods; however, Cao et al. do point out that if the adaptation of practices goes against the ‘spirit of Agile’, the adapted method might be less effective.

Chan and Thong (2009) only briefly mention philosophy, when referencing Nerur et al.’s (2005) paper. They have created a framework of Agile acceptance where, based on their literature review on Agile, they break agility into three factors: ability-related factors, motivation-related factors and opportunity-related factors. According to Chan and Thong’s framework, these three factors, along with the characteristics of Agile methods and knowledge management, have an effect on Agile acceptance in organisations. By knowledge management, they mean the creation, retention and transfer of both tacit and explicit knowledge. In their view, the more the organisation knows about Agile, the better the acceptance. Paradoxically, this perspective manages simultaneously to capture and lose the value-driven essence of Agile; for example, the factors of Agile highlight individual characteristics and motivation, but omit striving for self-management and craftsmanship. They discuss self-efficacy in ability-related factors but link it to use of ‘an agile methodology’, rather than accomplishments in the work. Training is seen as an encouraging factor for acceptance, but the authors also seem to believe that there should be pre-existing factors, such as good communication or shared understanding with the customers, for Agile methods to be appropriate. This seems to me to be a backwards approach: trying to fit Agile into organisations that are already fairly Agile, rather than educating organisations who struggle, to become better at Agile.
Lee and Xia (2010) approach Agile development from the perspective of ‘software development agility’: effective and efficient response towards change. They combine concepts of project team autonomy and diversity with response effectiveness and response extensiveness in an attempt to distil the building blocks of an Agile team. They focus heavily on the project outcomes, rather than the Agile journey, and see Agile development as a way to improve the change response. Software development teams are encouraged to prioritise between costs, schedule and scope; a prioritisation that will define the need for each agility dimension introduced in the paper, which should, in turn, define how much autonomy and diversity the teams require. To an author from the Agile-as-a-philosophy camp, this idea might be unsavoury, as it implies that the teams should conduct their work in the exact opposite way to what was described in the Manifesto. The Manifesto declares that the team should always be autonomous and diverse when it comes to technical competences.

Similarly, research by Goh et al. (2013) discusses how trust and control can be balanced in a large-scale Agile implementation, but seems to misunderstand the main Agile principles. Goh et al. are not satisfied by current empirical studies, which adopt the practitioner-driven Agile practices as their foundations, because these practices are interpreted differently. Instead, they rely on the Conboy (2009) definition of Agile: the taxonomy discussed in a previous section. Goh et al. emphasise control over collaboration and mention rigorous change management processes that make the Agility of their case project questionable. The paper cites the Agile taxonomy coined by Conboy (2009) and discusses internal control mechanisms that are relevant to self-organising teams (Manz & Sims 1987), yet it manages to bypass the values of the Agile Manifesto. By applying only Conboy’s taxonomy and definition and ignoring the values and the disciplines of the Manifesto, one can avoid applying practitioner-driven Agile, but I find this approach misguided. We agree with Wang et al. (2012, pp. 436), who point out: ‘Agile method use is often superficially judged as used or not used whereas the actual implementation can be subtle, partial and inconsistent and so categorising a method as used or not used may be overly simplistic’.

The last example of a business-oriented perspective comes from Martini et al. (2016), who have researched boundaries between the Agile teams and other business stakeholders. Their study focuses on software development teams who are applying Agile methods, but who reside embedded inside larger organisations whose core capability is not software but manufacturing. Martini et al.’s paper is oriented around the speed of the development and challenges in interactions between the Agile teams, the larger organisation and the non-software stakeholders. They see that Agile methods focus on responsiveness and continuous interactions between the stakeholders. Rather than a philosophy, Agile development is seen as an effective way to bridge gaps between teams and mitigate interaction challenges to reach business goals.

The Agile Spectrum

By conducting this literature analysis, we have established that there is a spectrum of perspectives towards Agile development methods, observable by analysing how both the researchers and their case organisations view Agile methods. All papers which discuss Agile methods fall to a point on this spectrum but in order to distinguish the perspectives presented in the literature, we need to create distinctions, points on the spectrum, which we can use to highlight the differences in these perspectives. We have labelled these different points on the spectrums as: 1) Avid, 2) Inclusive and 3) Pragmatic Agile. Each label attempts to capture the perspective held by the organisations as well as reflecting the views apparent in the Agile literature.

Avid Agile

Avid perspective towards Agile development presents an approach where the Agile methods are avidly followed and rigorously implemented in the organisation. Avid Agilists are devotedly following the values and principles of the Agile Manifesto and the methods guidelines as originally defined. This does not mean that Avid Agile perspective is not open to modifications if the need arises. However, the modifications have to fit the Avid Agile approach and not compromise the integrity of the Agile methods applied. Agile ‘purists’, strident advocates of Agile methods, would see every Agile organisation adopt the Avid perspective with their chosen method.

The Avid perspective chosen and carefully maintained by some Agile organisations is an example of the most devout followers of Agile Manifesto and organisation whose members have a philosophically
oriented view, falling on the extreme end on the Agile perspective spectrum. One could safely speculate that most organisations would fall into less absolutist ends of the spectrum. However, some organisations, most likely the ones where Extreme Programming is practised (discussed for example in Strode et al. 2012), are willing to implement and maintain the most rigorous methods and remain faithful to the ethos of the Manifesto. Most organisations have adopted a less strident perspective. Selecting Extreme Programming as the main method and especially the rigorous practice of pair programming is anomalous, even among organisations with a lengthy history of Agile application and recent studies point towards a declining interest in the method (VersionOne 2017; Dingsøyr and Lassenius, 2016).

**Inclusive Agile**

Inclusive Agile perspective refers to a view that embraces Agile methods as described by the Agile advocates but does not shy from modifications that might enable application of the methods in environments that stray from the ideal collocated development spaces. Inclusive Agile perspective is inclusive when it comes to different Agile methods applied by different organisations. Rather than focusing on a single method, the perspective allows collaboration parties to mix the Agile methods and amend them to the needs of the parties.

Organisations that operate fully or partially virtually would not be able to apply the most strident Agile methods that the catalogue of methods offered, such as Extreme Programming. Instead, they applied the Scrum methods (Schwaber 2004), which are often suggested and applied methods for virtual and globally distributed projects (Sutherland et al. 2008; Modi et al. 2013; Paasivaara et al. 2012; VersionOne 2017). Better customer inclusion is often the key driver for organisations with an Inclusive approach (Hoda et al. 2011). Literature discussing other virtual, globally distributed projects shares the Inclusive perspective (e.g. Holmström et al. 2006).

This does not mean that the Agile perspective is necessarily lacking in rigour. The Inclusive approach fosters an Agile environment that is as close to the spirit of the Manifesto as is possible due to the restraints posed by the distributed and/or virtual nature of the project. Many Agile organisations and studies written about them can identify themselves somewhere in the middle of the perspective spectrum, in the Inclusive categories rather than the extremities. If Avid Agile is one of the extreme ends of the Agile spectrum, Inclusive Agile perspective falls in the middle and encompasses probably the majority of Agile organisations. Lastly, the Pragmatic perspective, the other end of the Agile spectrum, is discussed.

**Pragmatic Agile**

Pragmatic perspective towards Agile development is the most lenient when it comes to the methodical application of the methods. In an environment where a pragmatic perspective is adopted, one could see the first team practising very rigorous Scrum whereas a second team could be in their early stages of adopting Agile methods and a third team could be practising traditional waterfall management. In the Avid approach, such combinations would not be possible and organisations with an Inclusive Agile approach would feel uncomfortable as well.

The practitioners of Pragmatic Agile know that their environment is complex and chaotic and they are applying Agile methods to be ready for changes and chaos, or as described by Goh et al. (2013, pp. 749) ‘focus is to develop capabilities that can flexibly respond to unpredictable project uncertainties stated above to meet the urgency to complete the project and to develop capabilities that allow learning from that experience’. This model is trying to strike a good balance between control and flexibility of the Agile methods’ application and focused on the right aspects of the development, interactions between the different parties (see for example Martini et al. 2016).

Unlike in the Inclusive approach, where the Agile method choices are made between very rigorous adherence to the selected Agile method and flexibility and involving everyone, the Pragmatic view has to choose between rigorous application of Agile methods and controls of stakeholders (Lee and Xia 2010; Goh et al. 2013). The Pragmatic perspective towards Agile acknowledges that Agile methods are generally well suited for software development projects but amendments and compromises to the most extreme methods are needed when project size and the number of stakeholders grows.
Conclusion

Our review of the literature across the selected body of knowledge of Agile development presents a spectrum of views and perspectives on the values and principles of the Manifesto, the methods themselves and their application. This spectrum of views ranges from very strict adherence to the Manifesto to advocacy of Agile methods as a flexible business tool and every combination in between. Authors and the organisations they have studied fall into a continuum of different views: the Manifesto and the guideline of Agile methods as either the non-negotiable baseline, as a loose set of ideas that can help organisations to succeed or as a set of best practices that can help the organisations to become more efficient and effective.

Adopting a certain philosophical view on the world does not necessarily make software development more or less successful but acknowledging differences in the views provides a lens for understanding why organisations collaborate and apply boundary objects and Agile methods in the ways they do.

Our study provides a novel perspective on Agile literature and distils the different views of the Agile perspective spectrum. This spectrum can be seen as a useful tool for future Agile literature reviews as it expands understanding of the body of knowledge. There are practical applications for this study as well. Organisations can apply the spectrum and analyse how their organisations approach Agile methods or whether there are differences between views inside the organisation. By using the spectrum as a lens, members of organisations can identify the extant state and possibly a future state, how they wish to transform the views towards Agile to understand better and potentially more rigorously apply the methods.

REFERENCES


