Towards the End of Agile: Owing to Common Misconceptions in the Minds of Agile Creators

Necmettin Ozkan[1][a] and Mehmet Şahin Gök[2][b]

[1]*Information Technologies Research and Development Center, Kuveyt Turk Participation Bank, Kocaeli, Turkey*

[2]*Department of Business, Gebze Technical University, Kocaeli, Turkey*

Keywords: Agility, Agile Software Development, Scrum, Software Engineering Process, Project Management, Miscomprehension.

Abstract: The Agile Software Development movement emerged from practice just like most of the works in the Agile Software Development evolved through practice. Thus, the creators and consultants of the Agile world may evangelize it with commercial concerns, resulting in “selling agility” to organizations as an object in the form of packaged practices (of methods/models/frameworks). Owing to the “sold practices” of the market and misleading misconceptions in the minds of Agile creators, there are issues in Agile like regarding it as a “holy” product and everything, binary thinking, trade-offs, and determinism that do not support agility in an absolute sense and even inhibit it, which ultimately lead to the end of Agile™. This study handles and discusses such seven prominent misconceptions and makes a prediction about the possible course of Agile™ and rise of agility.

1 INTRODUCTION

While the transition of pure and full agility capabilities like in the nature (agile with “a”) to human-made-artifacts (Agile with "A") or from human-made Agile artifacts to another one, people naturally involve their perceptions and intentions to pure agility, inhibiting a common understanding and accurate transition of it.

The market’s intention to sell “agility” to organizations appears as a force pulling Agile in different directions, derailing it from its main axis. In connection with industrializing, putting the product in sacrosanct form to preserve its shape comes as a tendency. While selling industrialized Agile™ (shortly stated as Agile hereinafter) products (such as in the form of strict frameworks) enables the market to have easier transactions and more volumes of quantity. Meanwhile, it may easily overshadow the agile mind-set and also limit our understanding to the market’s intentions instead of pursuing the real agility. Therefore, some concepts in Agile are seldom scrutinized under the shadow of a strong marketing monopoly due to regarding Agile as “sacrosanct.” When regarding something sacrosanct, people consider it to be special and are unwilling to see it criticized or changed. However, the rarely scrutinized misconceptions about agility need to be discussed properly to rectify the concepts to reach a better understanding of and independent agility.

Misconceptions around the frameworks and manifesto that are a universal inscriptive source of Agile have rarely been investigated (Janes and Succi, 2012; Ozkan, 2019). Considering their significant importance, this study aims to uncover common and prominent misconceptions regarding the (unintentional) perceptions and intentions in the heads of Agile (manifesto, frameworks, methods) creators by keeping Agile at the centre and touching the dominant and most popular Agile method, Scrum, in particular. It is known that one of the most involved domain with the agility concept is software development, which is also the domain we deal with in our study.

2 BACKGROUND

2.1 What Is Agile and Agility?

Being linear suits who is absolutely perfect within relevant contexts. The claim of being linear implies a

[a] https://orcid.org/0000-0001-9876-8728

[b] https://orcid.org/0000-0003-4072-2641

Ozkan, N. and Gök, M.
Towards the End of Agile: Owing to Common Misconceptions in the Minds of Agile Creators.
DOI: 10.5220/0010575102240232
In *Proceedings of the 16th International Conference on Software Technologies (ICSOFT 2021)*, pages 224-232
ISBN: 978-989-758-523-4

way towards being flawless and free from errors, mistakes, faults or defects. Behaving linearly in software development have a delusion and illusion of harbouring a secret divinity, more or less. The delusion of Waterfall methodology, the most typical and well-known example of classical software development approaches, in following a systematic and linear approach (to the software development life *cycle*) is such. The more systematic (self-confident) approach makes it more linear (claiming being perfect). Agility, on the other hand, refuses this unrealistic divinity and proposes returning to the man's essence who is not naturally perfect, including those developing a software.

The form of vitality is circularity, such as in the electrons circling around the nucleus of the atoms, the circular rotation of the human body between birth and death (humankind is born without a tooth, and he usually dies without teeth, he is born in need of help, and he becomes dependent on the help of others as he ages). A water drop giving life to nature (as expected in Mars) takes a circular course between ground and sky. The form in which processes gain vitality evolves into a circular shape instead of linear. For this reason, it is ideal for the software itself (as learning algorithms), which adds life to hardware and for its development (software development life *cycle*) to be in the circular form. With this circularity, the human accepts his/her imperfection and seeks to discover, research, learn, make mistakes, learn from mistakes, return to a starting point, learn again, know the unknowns and adapt, which are especially required for agility in the complex domains. Based on this natural need for agility and unfortunate expedient interest, people have invented Agile. Today, the term Agile represents a widely appreciated mind-set, approaches, methods and applications, for especially the software development domain as well as for other domains.

2.2 Why, What and How

There is an antecedent-successor and hierarchical relationship between Why, What and How concepts. Why comes first. It determines the missions, purposes, motivations and such. After determining Why, What and How come after. For instance, someone who thinks that one of the purposes of coming to life is to be a good person (Why) may determine a good education (What) as a task and design how to get it (How). While the reasons for a "Why" consist of relatively few and static items, "How" and "What" for a particular "Why" can probably be many and varying. It is noted that while "How" is determined after each "What", (a) new "What(s)" appear(s) inside the determined "How's", recursively. The whole picture is similar to a fractal (a complicated pattern in mathematics built from simple repeated shapes that are reduced in size every time they are repeated [Cambridge Dictionary]) with endless recursive How and What.

3 MISCONCEPTIONS

This section aims to include common and prominent misconceptions, even though there may exist more and different ones.

3.1 Regarding Agile as a Holy Product

The Agile movement emerged from practice, just like most of the works in Agile evolved through practice (Rolland, et al., 2016), inevitably resulting in the emergence of many consultants in the sector. We know agile in an adjective form is not an object to sell in one go, like a pencil, rather as an adjective, it is a journey with no definitive end. However, these emerging Agile consultants have dominated the sector (Paasivaara and Lassenius 2014; Hobbs and Petit, 201), and thus, consultants of the Agile world have evangelized it with commercial concerns, resulting in "selling agility" to organizations as an object in the form of packaged practices, as pointed out by one of the manifesto authors, David Thomas (Hohl, et al., 2018). Another manifesto author, Andy Hunt, states that "the word "agile" has become sloganized; meaningless at best, jingoist at worst" (Hunt, 2015). Among others, Bob Martin points out that the misinterpretation of Agile is caused by politics, imagination, and economic interest (Hohl, et al., 2018). As many other Agile products, the original ideas of the manifesto have already become more and more commercialized (Hohl, et al., 2018).

In this selling of Agile, the evangelists have exhorted implementers to adopt their methods whole; as such, every project following a particular method must adopt every practice, as described in the manuals, books and courses (Hoda et al., 2015); indicating that every method is controlled by a warden – the guru that has invented it (Jacobson and Stimson, 2018).

The passion for sticking to maintaining the packaged practices of the consultants keeps the community away from real cases providing evidence about persistent difficulties, deterioration situations or even complete failure (Gregory, et al., 2015). In parallel, we see that persistent difficulties, deterioration situations or even complete failures are mostly and “wisely” addressed by the manifesto creators to “poor implementation of Agile”, as seen in Hohl, et al. (2018), rather than to those arising from the universal design of “sacrosanct” Agile. This kind of approach has also come with less risk for both consultants and adopters in organizations who are ensured with “well-known” and “proven” products. Thus, Agile has gained speed and penetrated into many and varying organizations (Madsen, 2020) with the relatively easy adoption of the “proven” products.

With less adaptation efforts of the industrialized Agile products, consultants and adopters tend to regard Agile as “sacrosanct,” and there is a tendency to add practices to resolve ‘unknown’ problems that arise in specific contexts as an appropriate solution (Rolland, et al., 2016) without changing the “sacrosanct” part of it (Ozkan, 2019). Some people believe in the manifesto as the “holy grail” for successful software development (Hohl, et al., 2018), and the illusion of staying at the “comfort zone” continues to lead to thinking that Agile has universal value and represents some ultimate recipe and the “holy grail” of software engineering (Kruchten, 2013). Unfortunately, often the proponents of Agile, instead of providing a rational explanation of their applications, cite the gurus, the “holy” texts and just feels like being a prophet (Janes and Succi, 2012). The gurus, thus, enjoy the advantage in making the followers dependent on them; the adopters need the guru to continue to use the method in cases not described by the guru up front (Janes and Succi, 2012), even though, the gurus have invented a few practices, but “stolen” most from other gurus (Jacobson and Stimson, 2018).

Therefore, the Agile movement is accepting few criticisms (Agrawal et al., 2016) and is often criticized for their virtues and not for their vices (Janes and Succi, 2012), even though there are already some widespread and underlying assumptions (Agrawal et al., 2016; Madsen, 2020; Meyer, 2014), accompanying limitations (Turk et al., 2002), context dependencies (Hoda et al., 2015), imperfections (Ozkan, 2019) and common misconceptions, even in the heads of the creators. When it comes to the creators, however, the results of the survey with the contributors of the manifesto, as a body representing a remarkable power on Agile, show they are happy with what they have created; eleven of the authors see no need to change even the wording of the manifesto, except one proposing a minor play on words for replacing the word “software” with “solutions” (Hohl, et al., 2018).

Consequently, the Agile movement has religious and cult-like aspects (Byker, 2017). Stakeholders of Agile including consultants, trainees and new holders of certifications devote an expedience interest in the preservation of the approach, turn them into firm “believers” (Madsen, 2020). However, as Kruchten, (2019) states, with the dogmatic aspects of Agile owned by its “believers”, its movement has not always been very agile in its application to itself and have slowed the expansion of its own principles to some of the more complex or much larger software development endeavours.

3.2 Binary Thinking

This Hegelian Dialectic-based mind-set first defines an opponent, then uses the opposite side as a leverage. Nourished by this dialectic, both the definition and marketing of Agile are flamed by a counter-power, making it easier to attract followers, especially for those who are already tired of classical methods. One of the places where it can be seen most clearly is the manifesto. The initial idea of the manifesto aims at “uncovering [not agile yet] better [than something] ways of developing software” with different lightweight methods. Rather than providing the full power of agility, there are views to the manifesto as the packaging and structure of certain earlier concepts, with new terminology (Clarke et al., 2018; Meyer, 2014) as a marketing gimmick to sell intuitive development behaviour within a new livery (Hohl, et al., 2018) and as a reaction to older development models (Ozkan, 2019). Consider the example of documentation; according to Holh (2018), the manifesto authors agree on that the manifesto does not prohibit documents or processes, the users should just value them less than individuals or working software. One of them points out that “the basic issue is [the use of] documents [which is] one of the old paradigms”. With this binary coding (as in “old-new”), not surprisingly, Agile has been regarded as the opposite of large and heavy software processes (Janes and Succi, 2012); Barry, 2004], and the manifesto is more about a replacement of traditional methods, especially with its four underpinning values (Rolland, et al., 2016), resulting in a dual polarization with two edges; heavyweight / disciplined / predictive / plan-driven / right and Agile/left.

When it is stated there is value in the items on both sides, such discrimination leads to a clear logical error in it. Consider, which one of us expresses his/her two valuable items like this: Love over respect, people over animals, water over food, transportation by plane over automobile; no one. Because, it includes a logic error. Such a comparison always gives the correct result under one condition, if it is a comparison between absolute right and wrong or such. In all other cases where this condition is absent, there is a high probability that an exception will occur. Most probably, a time and a place (a context) will emerge, then the right side will be more valuable than the left side (think of the road you cannot go by plane). Therefore, one item or a combination of items on the right and/or left can be more valuable than the others. In this case, who can guarantee that the left side will be more valuable than the right, always?

This desperate approach confronts us as a secret divinity inside, again, determinism in Agile (elaborated later on). Anyone or those 17 people cannot know all situations that depend on varying contexts. In essence, isn't it agile to act appropriately when circumstances call for it? Or is Scrum, which defines itself as a "*process* framework," in contrast to the manifesto's first value? Isn't sticking to the manifesto contrasted with agility? Yes, based on the "over" logic. So, it is necessary to avoid binary thinking to normalize Agile and avoid its biased position springing from its radical approach to the "others" for the sake of the fundamentals of logic and real agility (Ozkan, 2019).

3.3 Unspoken Trade-offs in Selections

Unlike binary selection, here, a single choice is made. It is about what the choice brings and takes away. Choosing a blue colour also means not choosing green, such as in:

- Project vs. Product-Based Development
- Dynamic vs. Static Iteration
- Digitalization vs. Physical Dependencies
- Centralization vs. Decentralization
- Up-Front vs. Emergent

Trade-offs in such selections in Agile are not discussed sufficiently with the "holy product" effect. Whereas, in product-based development teams, expertise and focus increase; meanwhile, due to their static structures, it requires breaking large developments into multiple teams, making its management difficult. While static iterations increase motivation and focus on output, they damage agility due to the determinism they bring. While physical boards reinforce deeper emotions and more visibility, they lack many digital capabilities especially when scaling. The nature of Agile, which depends on meetings and face-to-face communications on physical platforms, manifests itself as a factor inhibiting flexibility and accompanying agility (Ozkan, 2019). While a cross-functional team enables the development of competence within the team, supporting faster production, it comes with a considerable cost and can turn into isolated structures. Requirements that are manageable up-front, to a certain extent, turn into additional costs when they are postponed and become emergent. Self-organizing emerges as part of the problems when scaling is required (Rolland, et al., 2016).

Agile takes only one side of these choices and tries to cover up the problems occurred with something else from itself, such as proposing refactoring as a recovery for increments that are forced to be "ready" at the end of static iterations when it turns out to be high technical debt.

One of the prevalent examples of combination of binary thinking and trade-offs in selection is the concept of project in Agile. Binary selection logic has spread to Scrum, the most widely used framework of Agile, and has resulted in the following dilemma in project notion: by inferring that the classical project management, which prefers to manage projects with deterministic methods, is not applicable to software development, Scrum has taken a stance against the classical project management. This opposing position has come as an antithesis, not only against the project management style but against the whole project notion. Project notion, project manager role and project management, all of them, are left undefined in Scrum (Ozkan and Kucuk, 2016). However, although this vagueness of project preserves its existence in Scrum, the search in Google for "Agile Project Management" with quotes gives more than six million results, one of which is the book of one of the authors of the Scrum Guide - Agile Project Management with Scrum. Once again, the superiority of "what" (project) over "how" (project management) has been seen, and it has been proven that we cannot ignore the reality that stands in front of us by closing our eyes. In other words, whether to manage the project in a classic way or Agile, customers still expect for the project with its unifying and abstracting power, as it is a necessary phenomenon for them (like a colour among others on their canvas) to encapsulate and manage a change delta set developed by product-based teams or project-based teams.

3.4 Regarding Agile as Everything

In Agile, we can talk about three basic mistakes in its coverage approach. The first one is about its endeavour to define the out-of-Agile fields from scratch when it is advantageous for Agile. The second one is about its endeavour not to re-define "the out-of-Agile" subjects when it is "disadvantageous" for Agile. The third one is its avoidance to re-define the foundations on which it is to base on.

Regarding the first one, we see a tendency in Agile to view the universe solely based on itself. In another saying, it is like "if the only tool you have is a hammer, to treat everything as if it were a nail" (Maslow, 1966). For instance, Agile brings little to quality within its change-oriented approach or to simplicity that is not "the art of maximizing the amount of work not done" in definition (Meyer, 2014), to managers, or value management. Indeed, which of them can be defined solely with a base on change orientation? Annosi et al. (2020) reports how Agile harms quality especially due to the time stress inherited in agility. Conboy and Fitzgerald (2007) argue that dealing with change paves the way for diminished quality. It is apparent that agility and quality are two separate and somehow conflicting concepts. It may come with an advantage to be an alternative to management mistakes in the past by offering fancy leadership advices or suggestions on value management, resulting in irresistible attraction for organizations. However, only replacing current manager titles with "leaders" should not be a solution. So is value management. Focusing on the (how) part of keeping up with changing requirements, determining the right value (what) has remained insufficient in Agile (consider the place in the guide given to the place between product owners and customers).

Essentially, organizations do not locate in a fully complex world that Agile aims to address. It may be a mixture of complex, complicated, simple, chaotic environments, with different ratio for different time periods and completely different for a different organization. Focusing only on the complex domain and proposing Agile as the "only solution" for the organizations imply a tendency to ignore the "others" where an unsuitable environment due to continuous changes occurs all the time and establishing a stable behaviour becomes problematic. Basically, it is theoretically and practically impossible to shape an object with a single adjective. In situations where there is a minor need to respond to changes (low level of complexity), the life continues with its rules.

Agile is not the only or first capability in the universe nor the only capability that organizations need. It is an adjective among the others and like the others (being disciplined, solid, mature, sustainable, valuable ...). When Agile comes in, it can affect some areas, but this influence should not reach to resulting in a from-beginning-definition of "others." Thus, Agile should leave their proper spaces to the other adjectives with respect.

When it is "disadvantageous" for Agile, it keeps its distance from the other realities of the organizations, such as discipline, processes, with the aim of excluding the traditional by its revolutionary perspective having little respect to the old ("like a teenager"). Again, this situation is related to the effect of binary selection that brings an antithesis against not only the non-agile side of the classical approaches but the whole. Even so, already, within Agile, there has been a shift to discipline and process; "Individuals and interactions over processes and tools [Manifesto]…Scrum is a *process* framework [Scrum Guide]", "sustainable development," "a constant pace," "regular intervals."…

Instead of being isolated, staying far from the other realities of the outside world and being quarrelsome with its neighbours ("only interested in being with its peers"), Agile should seek proper integration and harmony with "others" and know how to get along well with them. Otherwise, time passes and turning back to "others" seems to be its own disadvantage. Unfortunately, with the effect of "accepting few criticisms," it seems that this period will be quite long.

Thirdly, it is seen that the ground that should be the basis for Agile is neglected by it. For agility to shape an entity as an adjective, the entity should first exist. Being agile can be a feature of it, to bring agility to its existing behaviours. Still, code is written and tested, documents are produced, and analysis, design and planning are conducted. The pursuit for how agile tools, processes, documentation, coding can be is what we need. However, in Agile, how these standard activities can be performed are rarely investigated.

Shortly, as Philippe Kruchten wrote, "The Agile movement is in some ways a bit like a teenager; very self-conscious, checking constantly its appearance in a mirror, accepting few criticisms. Only interested in being with its peers, adapting fads and jargons, at times cocky and arrogant but I have no doubts that it'll mature further, become more open to the outside world, more reflective and also therefore also more effective." (Agrawal, 2016).

3.5 Determinisms in Agile

Again, there exists an unrealistic divinity with aiming to know all the conditions with ignorance of context; the desire to know the future. Who does not want it; the prophecy that the manifesto or Scrum will work for all time, location, in short, context variations.

The manifesto or an Agile framework is a tool proposed to achieve agility. The paradox is that the blind loyalty to them, is, at least, inconsistent with the first value of the manifesto. Any determinism or so called as formalism implies that their formalism can work for all context variations of us. However, what they propose cannot react to variations in contexts of millions and violates its own fourth value; Responding to change over following a plan (or any deterministic form). While this is the case, how can it be possible to think such a coding could work for all contexts? Maybe, it is because of compiler thinking.

When writing code for a compiler, a developer hardly consider and takes time, space, and variable situations into account. Because the compiler appeals to the same set up at all times and locations. A similar situation occurs when a group of people, mainly composed of software developers, write a manifesto or create a method for software developers; they do not usually take into account variations in context. However, it is not always right to regard water as more valuable than food; for a food-scarce location by the river, this equation is not right. But, the compiler cannot properly encode for this context variation if it is not coded accordingly because the compiler is also deterministic like the manifesto or the method. For all cases, it predicts (!) that the right side would be more valuable than the left side.

It is similar to the case of the Agile frameworks more restricting, determining (even in minutes), worrying about preserving its own shape, productizing and solidifying by being deterministic products. In summary, the logic of coding that creates and in blindly following the manifesto or a framework cannot react to changes in context variations and violates its own fourth value; instead of being open to changes, it tries to stick to the determinism that they like.

3.6 Agile Is Mainly a Matter of How

Let us go back to the time before the confusion has occurred and take a look at the still valid universal definition of agile: “able to move about quickly and easily” (The Cambridge and Macmillan Dictionary agree on this definition). According to this definition, agile or Agile is mainly a matter of “How”; to move quickly and easily to response to changes (What) with producing (another) “What.” However, in its pure form, it is regardless of to what it is to respond and what response is given. Just like independence from the layer between the product owner and customers, in Scrum, the events (planning, daily, review, and retrospectives) start after the determination of “What” (to develop). In this case, it is also quite possible to react to the wrong “What” in Agile. In many organizations that can be considered unsuccessful, there can be many changes happening in a usual day, and people response to these (wrong/worthless) changes, which is enough to call them agile. And, this is a relatively low level; no matter how teams do it when it is not the right change coming. In this sense, Agile can, at best, be a way to provide accurate response management to changes with, hopefully, a right “What”. Determining the right “What” coming is a matter beyond the bounds of agility and the dedicated effort that it deserves should be given to it.

3.7 Self-organizing, as a Deep and Hard Matter of “How”

The phenomenon of self-organizing is a matter of deep and hard "How," even at the team level. As the deep matter, self-organizing is a historical issue. The life of a human-being started as a self-organizing way at the beginning, then we wanted to have managers, and we are mainly still managed in that way. Today is what we have. Thus, self-organizing teams in Agile are to re-invent some code of life from the very beginning that can be very complex naturally. Shortly, regarding teams self-organizing is just the beginning.

Even then, in theory, having purely and fully self-organizing teams is impossible. Having an integrity, the organs of an individual maintain their relationships and dependencies through other organs, vessels and nervous systems and receive commands from the brain. In other words, organs cannot be fully self-organizing. So are individuals. Individuals are dependent on government, laws, and individuals including himself/herself, organizations and culture. So are teams of individuals. Norms, rules, dependencies, restrictions and determining factors do not allow teams to organize themselves fully. Managers are still there. Regardless of whether there is an in-team manager or not, the team still receives direction from the top levels. Simply put, becoming more dependent upon other teams' practices as a result of inherent dependencies of projects/organizations makes teams hardly self-organizing (Rolland, et al., 2016). The emergence of dependencies between teams and other entities indicates that the ability of teams to act independently and in a self-organized manner is not valid (Ozkan, 2019) In summary, within this relationship spiral and boundaries involving the outward side of the teams, it is not possible to be absolutely self-organized (hard part).

The teams inside have different, even conflicting dynamics. Inside each individual, there are different, even conflicting dynamics (yin-yang and like fractals). To justify this, consider that self-organizing teams do not consist of absolutely self-organizing individuals; then, an organization cannot be expected to consist of absolutely self-organizing teams. The goal of self-organizing for each individual (team) appears to be in opposition to organizing with the consideration of others inside the teams (another hard part).

Another issue with self-organizing is that it is just a matter of "How," even at the team level (somewhere in the fractals of What-How-What-How…). The team's self-organizing around a determined "What" is at a relatively low level. Product owners determine what to do, and the teams are free in how to do it. This reminds us again that "What" comes before "How" (How part).

4 LIKELY FUTURE OF AGILE AND AGILITY

As many products do, Agile is inevitably following the Gartner Hype Cycle (Hohl, et al., 2018; Janes and Succi, 2012; Madsen, 2020) and it is unavoidable to witness Agile's disillusionment phase with the eyes of most of the readers of this article. Janes and Succi (2012) puts forward that Agile has already reached the trough of disillusionment phase, which implies it is time for opening gates to question "sacrosanct Agile" for the sake of agility. Otherwise, most organizations will keep "doing Agile," and the real agility will continue to stay behind the "sold" practices because the market wants to sell "agility" like an object for its profit, forever, which, as stated by Denning (2016), will make the Agile movement die.

Appelo (2011) and Leffingwell (2011) state that Agile is context-specific, having its roots in complexity theory. Appelo (2011) notes that agility is misunderstood by most people, because they have not understood complexity theories from which Agile originates. According to him, any simplistic, linear model [that poses predictability somehow, like an Agile method] is bound to fail. To make matters worse, the Agile methods are all monolithic, not modular, not designed to be reusable, incompatible to mix and match and controlled and dictated by a warden (the guru) (Jacobson and Stimson, 2018). Conboy and Fitzgerald (2007) notes that "the very name agile suggests that the method should be easily adjusted to suit its environment", which make agile system itself complex, not simple. It is claimed that agility is not guaranteed by applying an Agile method, even fully, (Hohl, et al., 2018) and even more, the real agility does not come to light with the prison of such methods (Jacobson and Stimson, 2018). Adolph (2006) stresses that agility depends on organizational culture and climate, and not on tools and processes. This cultural dependency makes agility complex enough beyond far from simplicity of any product, "process and tools".

Instead of monolithic, not modular, not reusable, incompatible, and controlled by a warden practices of the methods that prevent people from progressing from initial stages to more advanced ones, people should be able to customize and contextualize the practices by selecting, mixing, deleting and changing them from their customized library of practices.

This means that we can still use flexible, adaptable and responsive Agile practices in a good way, provided with the guidance of Shu-Ha-Ri philosophy that is a Japanese art concept describing the stages of learning to mastery. "Shu" stands for following traditional wisdom, fundamentals, and techniques, "Ha" stands for breaking them, finding exceptions to them, new ways or techniques reflecting on contextual truths, and "Ri" stands for leaving the rules to create new ones that are natural. After all, the journey that starts with Agile practices should continue with agility. In other words, Agile should represent not a final destination, but a temporary state ("Shu" stage at best) without excessive adherence to frameworks with knowing that one day they will be adjusted totally or partially. Otherwise, Agile as a noun, not an adjective, leads to an end; a binary state, finally, a dead state. However, agile as an adjective represents a journey guided by a mind-set and principles.

This reminds us again of what we had to do from the very beginning; we should first and foremost focus on the mind-set and principles rather than practices. We cannot reach to the agility mind-set through practices, on contrary, we need to create practices in the light of mind-set and principles. Practices without any mind-set leads to an identified end; a binary state, finally a dead state.

Agile, the most well-known representative of agility today, with its high formation of "holy" industrialized products eventually leads us its end. Even more, as stated by Kruchten (2019), Hohl, et al. (2018), Prikladnicki et al. (2019); and Cagle (2019), Agile is already dead. Even so, reaching this "dead" state can be an opportunity for raising of agility, with leading to "Ha-Ri" stages; moving towards the Slope of Enlightenment and Plateau of Productivity stages on the hype. It can be a kind reminder of that we cannot ignore the reality (the whole picture) by wearing glasses of the methods; what we have to manage is still the same. We should continue our journey from Agile to agility by focusing on what is essential, with people, proper mind-set and fundamental principles, freed from "method prisons", determinism, dogmatism and binary thinking.

We anticipate that agility is an indispensable need for organizations. Then, during this journey, it is a need to search for the pure and real agility, rather than any Xgile formations of it. For the pure and real agility, we should stop repeating Agile [or any form of Xgile] like a mantra, go beyond the dogma of this or that method, this or that practice of it and exploit the fundamental principles of agile fully integrated in the way we work to get its real value (Kruchten, 2019). People should come to the fore by considering they are inherently more dominant than any frameworks, processes, tools or manifesto and because they are the most proficient ability we have to deal with complexity.

We believe that when Agile becomes more mature and more reflective, it will leave its "sweet comfort zone", normalize its biased position resulting from its radical approach to the "others" and have good relations with other needs and realities of organizations. However, such a radical initiation would not probably come from the original authors of Agile (manifesto), as they have stopped focusing on the future trends of the manifesto (Hohl, et al., 2018). Maybe, it is time to realize, again, that agility should not be under a monopoly of a certain coterie but a common property of the whole community and the responsibility of improving it should belong to the whole community (Ozkan, 2019). To help to define such adequate world outside of the "agile sweet spot" to profit from the real agility, cold-headed and impartial investigation is required even though such work is generally not very easy (Kruchten, 2013).

5 CONCLUSION

There are contexts in which we know what will happen, such as in math and classical physics. They do not delude us. We can play our game in this area. However, the complex world is not like that. It is not the realm of rigid frameworks, static rituals, blind followers, claims of holiness. This is where man should be a human, altogether, including the manifesto and framework creators.

However, self- actualization has been witnessed throughout history, and as for now, people want to use force, to gain respect, to be known, to know, to see, to help and to shape in order to actualize themselves. This is also the basic instinct underlying the desire of knowing the future, planning the future, prophesying, worshiping for plans, acting deterministically with formalization of the methods, feeling pain when the forms and plans deviate, covering up deviations (realizing the moment of not being sacrosanct). They design frameworks (borders), rituals and roles (POs, SMs), shortly their games (like life, which is another game) and become happy when people play with their games; remember, Scrum is a game ["The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game"). People who have this basic instinct seek the same thing, whether they design Waterfall or Agile; the desire of being deterministic (dignity).

It is natural that this new claim of divinity is subject to the regarding its products as sacrosanct and preserving them as they are. All of these are reflections of accepting no power but himself (binary thinking), the desire to be perfect and flaw-free (hidden trade-offs), covering up the trade-offs or closing them with another solution (!), isolation from and avoiding the need to define the outside world by proposing *self*-organizing and cross-functional team structures, giving autonomy to the team to choose between right and wrong (eating or not eating the forbidden *apple*) by offering them self-organizing, and then disappearing suddenly ("Scrum is lightweight, simple to understand" [my side is done], " difficult to master "[this is your side]).

However, the real agile suggests approaching the non-deterministic domain as a human. It accepts that a human is flawed and imperfect. It is the admission point that we cannot know perfectly the future or any formation that is expected to work in future. It is an acceptance to renounce the claim of being sacrosanct. However, the sacred acceptance of a group of leaders and their proposed/sold practices takes us back to the original mistake. The blind loyalty to frameworks, even the manifesto itself, which is a kind of process-derivation, is, at least, inconsistent with the first item of the manifesto. We all need liberated agility, not defective and "holly" Agile products. Shortly, Agile is already dead. Long live agility!

6 SUMMARY

Adequate comprehension of identifying right problems and defining them accurately are key points (Chen 1975). Issues of about a problem might be obscured from the surface and goes beyond its immediate range (Chen 1975). The main contribution of this study is a thorough identification of misconceptions of Agile beyond its surface. We believe this is a crucial contribution especially for practitioners before diving into an Agile implementation. In terms of the misconceptions in the context of our study, the current literature covers limited contents. For the academia, hopefully, this study serves to fill this gap and would shed some light on areas that deserve further investigations.

REFERENCES

Adolph, S. (2006, July). What lessons can the agile community learn from a maverick fighter pilot?. In AGILE 2006 (AGILE'06) (pp. 6-pp). IEEE.

Agrawal, A., Atiq, M. A., & Maurya, L. S. (2016). A current study on the limitations of agile methods in industry using secure google forms. Procedia Computer Science, 78(291-297), 35.

Annosi, M. C., Foss, N., & Martini, A. (2020). When Agile Harms Learning and Innovation:(and What Can Be Done About It). California Management Review, 63(1), 61-80.

Appelo, J. (2011). Management 3.0. Leading Agile Developers, Developing Agile Leaders. Addison-Wesley.

Byker, M. (2017). Is Agile a Religion? Available online: https://www.linkedin.com/pulse/agilereligion-martin-byker.

Cagle, K. 2019. The End of Agile. Forbes.

Chen, G. K. (1975). What is the systems approach?. Interfaces, 6(1), 32-37.

Clarke, P., O'Connor, R. V., & Yilmaz, M. (2018, May). In search of the origins and enduring impact of agile software development. In Proceedings of the 2018 International Conference on Software and System Process (pp. 142-146).

Conboy, K., & Fitzgerald, B. (2004, November). Toward a conceptual framework of agile methods: a study of agility in different disciplines. In Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research (pp. 37-44).

Denning, S. (2016). What's Missing In The Agile Manifesto: Mindset. Forbes, Available Online: https://www. forbes. com/sites/stevedenning/2016/06/07/the-key-missing-ingredient-in-the-agile-manifestomindset.

Gregory, P., Barroca, L., Taylor, K., Salah, D., & Sharp, H. (2015, May). Agile challenges in practice: a thematic analysis. In International Conference on Agile Software Development (pp. 64-80). Springer, Cham.

Hobbs, B., & Petit, Y. (2017). Agile methods on large projects in large organizations. Project Management Journal, 48(3), 3-19.

Hoda, R., Kruchten, P., Noble, J., & Marshall, S. (2010, October). Agility in context. In Proceedings of the ACM international conference on Object oriented programming systems languages and applications (pp. 74-88).

Hohl, P., Klünder, J., van Bennekum, A., Lockard, R., Gifford, J., Münch, J., ... & Schneider, K. (2018). Back to the future: origins and directions of the "Agile Manifesto"–views of the originators. Journal of Software Engineering Research and Development, 6(1), 15.

Hunt, A. (2015). The Failure of Agile. URL: https://toolshed. com/2015/05/the-failure-of-agile. html, accessed, 01, 2021.

Jacobson, I., & Stimson, R. (2018). Tear Down the Method Prisons! Set Free the Practices!. Queue, 16(5), 101-127.

Janes, A. A., & Succi, G. (2012, October). The dark side of agile software development. In Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software (pp. 215-228).

Kruchten, P. (2013). Contextualizing agile software development. Journal of software: Evolution and Process, 25(4), 351-361.

Kruchten, P. (2019, May). The end of agile as we know it. In Proceedings of the International Conference on Software and System Processes (pp. 104-104).

Leffingwell, D. (2011). Agile Software Requirements. In: Lean Requirements Practices for Teams, Programs, and the Enterprise. Addison-Wesley ISBN-10: 0-321-63584-1

Madsen, D. Ø. (2020). The Evolutionary Trajectory of the Agile Concept Viewed from a Management Fashion Perspective. Social Sciences, 9(5), 69.

Maslow, A. H. (1966). The psychology of science. p. 15. ISBN 9780976040231.

Meyer, B. (2014). Agile!: The Good, the Hype and the Ugly. Springer Science & Business Media.

Ozkan, N., & Kucuk, C. (2016). A systematic approach to project related concepts of scrum. Revista de Management Comparat International, 17(4), 320.

Ozkan, N. (2019, November). Imperfections Underlying the Manifesto for Agile Software Development. In 2019 1st International Informatics and Software Engineering Conference (UBMYK) (pp. 1-6). IEEE.

Paasivaara, M., & Lassenius, C. (2014). Communities of practice in a large distributed agile software development organization–Case Ericsson. Information and Software Technology, 56(12), 1556-1577.

Prikladnicki, R., Lassenius, C., & Carver, J. C. (2019). Trends in Agile: From Operational to Strategic Agility [Practitioners. IEEE Software, (1), 95-97.

Rolland, K., Dingsoyr, T., Fitzgerald, B., & Stol, K. J. (2016). Problematizing agile in the large: alternative assumptions for large-scale agile development. In 39th International Conference on Information Systems (pp. 1-21). Association for Information Systems (AIS).

Turk, D., France, R., & Rumpe, B. (2002). Agile Software Processes: Principles, Assumptions and Limitations. In Technical Report. Colorado State University.