

## Fake News: Simply Agile

Jil Klünder<sup>1</sup>, Anna Schmitt<sup>2</sup>, Philipp Hohl<sup>3</sup> and Kurt Schneider<sup>1</sup>

**Abstract:** Since 2001, agile software development has increasingly been spreading into industry. The original contributors of the manifesto for agile software development had in mind to make the world a better place for software developers. Focusing on social and human factors, it is widely accepted that the adaptation of agile methods and practices leads to more satisfied and, hence, more productive developers. However, agile teams are not necessarily happier than teams working in a plan-driven way. This position paper presents reasons why an agile adaptation is hard to achieve and might lead to dissatisfied developers. It is based on preliminary results of our survey with the seventeen contributors of the manifesto. It further comprises the results of an interview study. Both data sources allow drawing conclusions concerning the consequences of agile transformation for developers.

**Keywords:** Agile software development, human factors, agile manifesto, Scrum

### 1 Introduction

With the manifesto for agile software development [AA01], the agile mindset has become available to the masses. Since then, the use of agile software development methods has been getting more and more important and continues spreading into industry [VO17].

There are mainly two reasons for companies to use agile: Some companies do so because they believe in the values and principles of the manifesto, and others do so because it is seen as the best practice [Lo17]. At first sight, adapting agile seems to lead to more satisfied developers because the manifesto clearly focusses on human and social factors. But this is only true if the team working on the process is involved during the transition and adaptation [Ga14]. The lack of this involvement leads to the refusal of the new way of working, and a cultural change is hard to achieve [DKZ15].

According to our experience, many companies claim to be “agile”. But they often neglect that agile has two components to consider: technical and cultural agility [DKZ15]. The neglect of cultural agility may lead to dissatisfaction of the involved persons due to the way they work. We want to examine the gap between the processes and possibilities presented in the manifesto and real-life experiences.

This paper is based on the preliminary results of a survey with eleven contributors of the manifesto and an interview study by Lockard et al. [Lo17]. The survey reveals reasons why developers were dissatisfied with agile transition.

---

<sup>1</sup> Software Engineering Group, Leibniz Universität Hannover, Welfengarten 1, 30167 Hannover, Germany  
e-mail: {jil.kluender, kurt.schneider}@inf.uni-hannover.de

<sup>2</sup> Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany  
e-mail: anna.schmitt@iese.fraunhofer.de

<sup>3</sup> Daimler AG, Research and Development, Wilhelm-Runge-Straße 11, 89081 Ulm, Germany  
e-mail: philipp.hohl@daimler.com

## 2 Status Quo and Related Work

The VersionOne survey [VO17] provides an overview of agile methods in use, the variety of existing practices and the benefits of adapting agile. The most widespread agile methods are Scrum, eXtreme Programming (XP) and custom hybrid approaches [VO17].

Diebold et al. [Di15] conducted an interview study with German companies investigating how Scrum is adapted in various contexts. The study discovered that most adaptations vary the sprint length and omit roles or artifacts. These adaptations are based on superior factors like the size of the company, efficiency or “agile maturity”. The fact that Scrum is the most used agile method is based on the presumption of increased team motivation, faster time to market, as well as the need to manage changing priorities and team morale [VO17, SD16].

But there are only a few studies considering the influence of the core values on the changing mindset towards agile and the satisfaction of the development team. Mohr and Schirmer [MS17] examine the relation between agile software development and developers’ happiness as one reason why companies want to be agile.

In contrast, agile software development approaches often focus on satisfying the customer. In smaller environments, agile software development processes often have positive impacts on customer satisfaction. Boehm and Turner [BT05] analyzed agile transformation in large projects. They identified three crucial areas for combining agile and plan-driven development. One crucial area is the different management attitude of people within the traditional and the agile development. They state that developers need to have higher-level skills, but Boehm and Turner do not focus on the developers’ satisfaction [BT05].

To analyze human factors and developers’ satisfaction, we present preliminary results from our survey. Initially, the survey aimed at getting to know about the origins and directions of the agile manifesto according to the thoughts of the 17 contributors. Besides, we gained profound insights into the areas of agile transformation and the influences on human factors. Eleven of the 17 contributors participated in the survey. Furthermore, we consider insights from an interview series by Lockard et al. [Lo17]. They also interviewed the original contributors of the manifesto for agile software development.

In extension to these studies, in this paper we focus on the human factors while adapting agile practices. It aims at analyzing the influences of agile transformation on developers’ satisfaction. In addition, it identifies the unconsidered difficulties complicating being agile.

## 3 Agile is More Than Scrum

The initial idea of the manifesto aims at uncovering “better ways of developing software” [AA01]. It emerges from the core ideas of different light-weight programming methods. It comprises the influence of Scrum, XP and several more. This results in the four core values in the manifesto for agile software development [AA01]:

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

Many people think that Scrum is the only agile method [SS16]. Although Scrum is nowadays the most frequently used agile method [VO17], it is necessary to understand that “agile” is more than Scrum [Lo17]. With the possibility to certify basic knowledge about the method and a significant supporting community, Scrum transports agility into industry. The downside is that many practitioners are convinced of improving everything by introducing “agile” in the shape of Scrum. Introducing Scrum seems easy because it is mainly a process, and hence more of a management framework [SS16]. From the management point of view, the Scrum framework supports being agile. But according to one of the core statements of the manifesto, namely “individuals and interaction over processes and tools” [AA01], rigorously following Scrum to be agile is a contradiction in terms: On the one hand, interactions and individuals should be focused on in order to be agile, but, on the other hand, one strictly follows the planned Scrum process with all its meetings. Consequently, Scrum is neither a necessary nor a sufficient criterion for being agile. Only following Scrum does not guarantee a changing mindset and might not end up in more satisfied developers.

## 4 Study Design

We conducted a survey with the original contributors of the manifesto to gain insights into the initial idea of the manifesto for agile software development and their initial understanding of what “agile” was meant to be.

We asked the contributors to answer the following five questions by e-mail. We received answers from eleven of the seventeen contributors. Two authors of this paper also participated in a skype conference with one of the contributors.

- Why did you participate in the manifesto for agile software development 16 years ago?
- What was your original contribution to the manifesto for agile software development?
- Which parts of the manifesto would you skip or change today?
- Are there necessary new concepts and adjustments for the manifesto?
- What trends will shape the future of agile development from your perspective?

Two of the participants told us about the “Agile Uprising Agile Manifesto Review” interview study by Ryan Lockard et al. [Lo17]. We used these interviews as a secondary data

source to gain more insights on the contributors' thoughts about the human factors within the agile transformation process.

The questions in our survey and the questions from Lockard et al.'s interviews led to rather general answers which we needed to parse in order to investigate the human factors. We analyzed both data sources, i.e. the answers of our survey and the transcripts of Lockard et al.'s interviews following the procedure proposed by Mayring [Ma10]. This allowed us to examine the answers independently from the initially asked questions, which did not directly focus on human factors.

Of course, our results must not be over-generalized. We just considered the contributors' ideas. But it is not the aim of this paper to generalize our results. Instead, we want to present the views of the originators with respect to the human factors within the process.

## 5 Views of the Manifesto Originators and Interpretation

According to our survey with eleven contributors and the interview study performed by Lockard et al. [Lo17] with 15 contributors, we identified three main influences on human factors:

- Reasons why companies want to be agile,
- Misunderstanding and misinterpretation of what the term “agile” means,
- Acquisition of suitable methods without adaption.

Because it is “state of the practice” to be agile, many companies aim towards an agile way of working. It seems easy to implement Scrum by the book because it presents a concrete execution plan everyone wants to have and follow [Lo17]. This could lead to developers' dissatisfaction by introducing Scrum due to “the missing understanding of the applicability of agile methods within their context” [Ho16, p. 471]. Due to the focus on human factors, the developers have positive expectations before introducing Scrum. But when they realize that implementing Scrum is not as easy and as rapidly beneficial as they thought, they are dissatisfied.

As Mike Beedle emphasizes, being agile aims towards the understanding of what makes sense – and to what extent [Lo17]. Jim Highsmith points out that it is important to realize that the manifesto is not based on methods and processes [Lo17]. It bases on the four core statements and the twelve principles. Agile methods only help to fulfill the principles and, consequently, the values [Lo17]. Therefore, it is not sufficient to implement an agile method; it is furthermore important to understand the underlying agile basics. It could help to introduce agile in a slow and step-wise manner to gain the developers' acceptance [Ho16]. Otherwise, the agile mindset and the practices of self-organizing teams and empowering people get lost.

The initial ideas of the manifesto are developer-focused and rather technical. But often, managers or business owners decide to be agile. Due to the technical level, they often misinterpret the agile ideas and do not know how to integrate them into daily business

[Lo17, Ho16]. As a consequence, this misinterpretation on management level increases the pressure on developers and impedes a motivated and productive working environment.

One aim of the manifesto is to establish a better development environment for programmers [Lo17]. As Ron Jeffries mentions, this is not entirely true anymore [Lo17]. In the view of Brian Marrick, people consider agile and Scrum rather as a brand than a mindset [Lo17]. But this mindset is the basis for successful agile software development. Agile approaches might not always fit the context of a team and, hence, leave developers dissatisfied [Lo17].

As Bob Martin recommends, teams should adapt carefully chosen methods according to their specific needs [Lo17]. Using Scrum by the book does not necessarily lead to success and happier developers. However, the concepts behind the manifesto are still valid [Lo17]. But there are many misunderstandings in the adaption of agile methods. Andy Hunt quotes the example of a team that participates in a 3-hour stand-up meeting, has a backlog, thinks it is agile and is still not satisfied with the way of agile working [Lo17]. From the developers' point of view, this management decision leads to long lasting meetings, monitoring the developers' performance and gaining the opportunity to call themselves agile. In such cases, the missing understanding of the right context-specific cherry-picking of agile practices and a compulsory way of working lead to a rejection of agile development. Agile transformation requires more communication effort to ensure developers' acceptance [Ho16]. It is necessary to set up a good working environment and to integrate the developers into this transformation process from the very beginning. By doing so, it is possible to establish a self-organizing team and initiate a slow and step-wise mindset change.

## 6 Conclusion and Future Work

Only using agile methods and practices does not necessarily lead to a more satisfied development team. Instead, it might impair software quality and productivity as well as social factors within a team, such as motivation and team performance.

Based on the two studies mentioned above, we identified three factors negatively influencing developers' satisfaction during the transformation process and afterwards: The various reasons why companies want to be agile, the misunderstanding and misinterpretation of what the term "agile" means and the acquisition of suitable methods without adaption. Misinterpretation of agile practices, wrong implementations, and market pressure will hamper developers' satisfaction. It is necessary to recall the original thoughts of agile development and focus more on the developers inside the process. In this way, a mindset change could be initiated, which leads to broader acceptance and smoothes the way for successful agile adaption with a content team.

Of course, our findings are just a small part of the challenges that might influence developers' satisfaction. In future, we will focus on difficulties in information flow and the creation of a collaboration model for satisfied agile software development teams.

## References

- [AA01] The Agile Alliance: Manifesto for Agile Software Development, <http://agilemanifesto.org/>, Status: 04.05.2017.
- [BT05] Boehm, B., Turner, R.: Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software* 22/05, pp. 30-39, 2005.
- [Di15] Diebold, P. et al.: What do Practitioners Vary in Using Scrum? In (Lassenius, C., Dingsøyr, T., Paasivaara, M. Eds.): 16th International Conference on Agile Processes in Software Engineering and Extreme Programming, Helsinki 2015. Springer, pp. 40-51, 2015.
- [DKZ15] Diebold, P., Küpper, S., Zehler, T.: Nachhaltige Agile Transition. Symbiose von technischer und kultureller Agilität. In (Engstler, M. et al. Hrsg.): *Projektmanagement und Vorgehensmodelle*. Bonn, S. 121-126, 2015.
- [Ga14] Gandomani, T.J. et al.: How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications* 8/14, pp. 129-148, 2014.
- [Ho16] Hohl, P. et al.: Forces that Prevent Agile Adoption in the Automotive Domain. In (Abrahamsson, P. et al. Eds.): 17th International Conference on Product-Focused Software Process Improvement, Trondheim 2016. Springer, pp. 468-476, 2016.
- [Lo17] Lockard, R. et al.: Agile Uprising Agile Manifesto Review [Online], <http://podcast.agileuprising.com/manifesto-author-review/>, Status: 03.04.2017.
- [Ma10] Mayring, P.: *Qualitative Inhaltsanalyse*. In (Mey, G.; Mruck, K. Hrsg.): *Handbuch Qualitative Forschung in der Psychologie*. VS Verlag für Sozialwissenschaften, Springer, Wiesbaden, 2010.
- [MS17] Mohr, J., Schirmer, U.: *Happiness Research und Agilität. Zwei Seiten einer Medaille*. Scrum Day 2017, Stuttgart, Filderstadt, 2017.
- [SD16] Schmitt, A., Diebold, P.: Why do we do Software Process Improvement? Study on Commonly Used Goals in Practice. In (Abrahamsson, P. et al. Eds.): 17th International Conference on Product-Focused Software Process Improvement, Trondheim 2016. Springer, pp. 360-368, 2016.
- [SS16] Sutherland, J., Schwaber, K.: *The Scrum Guide*, <http://www.scrumguides.org>, Status: 2016.
- [VO17] VersionOne: 11<sup>th</sup> Annual State of Agile Report, <https://www.versionone.com/>, Status: 2017.