XP 2019 Panel: Agile Manifesto – Impacts on Culture, Education, and Software Practices

Dennis Mancl[1(✉)] and Steven D. Fraser[2]

[1] MSWX Software Experts, Bridgewater, NJ 08807, USA
dmancl@acm.org
[2] Innoxec, Santa Clara, CA, USA
sdfraser@acm.org

Abstract. Manifestos are often a vehicle to trigger change by catalyzing discussion around a core group of ideas and values – and there is no doubt that the publication of the "Agile Manifesto" in 2001 increased visibility for an emergent breed of lightweight software practices. The panel discussed how the Agile Manifesto has impacted academic and industry software professionals in the areas of culture, education, and software practices.

Keywords: Agile · Agile Manifesto · Software development practices

1 The Agile Manifesto at XP 2019

The Agile Manifesto [1] is the most widely read statement of agile values and principles. At the time it was written (2001), many of the thought leaders in software development practices were already beginning to use lightweight Agile development methods: Scrum, Extreme Programming, Adaptive Software Development, and others. The term Agile was coined to explain the common values behind all of these lightweight approaches. Many Agile instructors and coaches talk about the Agile Manifesto and its values when they deliver introductory agile training.

But today, many who come to university software engineering courses or corporate training courses in Agile development have never heard of the Agile Manifesto. As Agile approaches to business practices are spreading, the Agile Manifesto looks a bit out of date and developer centric. For example, the Manifesto's opening phrase refers to "better ways of developing software" and one of its four declarations is that "working software" is more important than comprehensive documentation.

Is it time to revise the Agile Manifesto? How is the Manifesto received by a new generation of Agile practitioners, and what kind of impact is the Manifesto likely to make on the future of Agile education and Agile culture?

The panel represented both academic and industry experiences. Rebecca Wirfs-Brock is a design methodologist, a consultant on patterns and Agility, and an author of two influential books on software design. Rebecca is the founder of Wirfs-Brock Associates and is based in Oregon. She also serves as the Director of the Agile Experience Reports initiative for the Agile Alliance. Maria Paasivaara is an Associate

R. Hoda (Ed.): XP 2019 Workshops, LNBIP 364, pp. 143–148, 2019.
https://doi.org/10.1007/978-3-030-30126-2_17

Professor at IT University of Copenhagen, with experience in running empirical studies in the area of software engineering and software development practices. Werner Wild is an Agile coach and consultant with Evolution Consulting in Austria, and he has significant teaching experience at technical universities in several European countries. Evelyn Tian is a full-stack Agile coach and trainer with Evelyn Konsult AB, based in Sweden. She has coached companies and collaborated with universities on different continents, and she is an Advisor to IEEE Software Board. Steven Fraser, the panel impresario, is based in California where he advises on tech transfer and open innovation strategies for Innoxec. Previously he was the Director of the Cisco Research Center and the Lead for HP's Global University Programs – and he has organized and delivered over 75 software engineering conferences, panels, and workshops.

2 Agile Manifesto Panel Discussion

In this panel session, the panelists explored a number of questions about the Agile Manifesto's impact on culture, process, and education in the software development community. The main discussion topics:

- Is it time to update the Agile Manifesto?
- How has the Manifesto affected the educational practices we use to teach software development in universities?
- How useful are Agile games in education?
- How do we assign grades to students when they are asked to work on a project as part of a team?

Some people believe in the Agile Manifesto with a religious fervor, and they would never want to change it. Other people believe the Agile Manifesto should be "agile," which means it should evolve as needed. And there are some complaints that the Agile Manifesto is too old and too limited, maybe as a result of the biases of a group of 17 specific people in a mindset of 2001 technology – it has been pointed out before that the group was all male persons (no women) and mostly consultants.

The panelists were asked about whether it would be a good idea to update the Agile Manifesto to take into account eighteen years of evolution in Agile development practices as well as the increased interest in applying Agility to non-software problems in business. Rebecca Wirfs-Brock asserted that the Manifesto not likely to change: it is "immutable… at this point in time." Because the Manifesto is owned by the original authors, it will never be changed. Maria Paasivaara agreed, citing a 2018 journal paper by Philipp Hohl and others that summarized some recent interviews with the original Manifesto authors [2].

Even if they can't change the Manifesto, many people who want to be Agile feel compelled to extend it. Rebecca mentioned seeing this compulsion in her role working with Agile experience reports. She has seen a trend for Agile teams to talk about their attempts to "rewrite the Manifesto" to fit different contexts, including projects that don't involve software. She gave an example: "We had someone last year writing about how these principles and practices were restated for working in a research laboratory." In almost every case, their changes are minor edits of the Manifesto text, not major

reworking of the content or structure. It is clear that they find that the principles and values of the original Manifesto are "mostly" true, just software focused.

The Agile Manifesto became popular in the 2000s because of "good timing," according to Rebecca. It was written at a time when many researchers and industry managers saw the software industry swinging in the direction of heavyweight software development techniques such as the Rational Unified Process, with formal design diagrams and a complex design lifecycle. But there always was an active software development process counterculture, and the ideas of lightweight process had been explored for many years. It just needed a catalyst to turn into a visible movement.

Rebecca explained that the authors of the Agile Manifesto found a way to stir things up, to rally other people to follow a different path. "Manifestos are like shaking your fist in the air, declaring that this is something different. It was trying to pivot the pendulum from this stodgy, heavy process."

All of the panelists had examples of the impact of Agile on software engineering education. The biggest trend in many universities is to have students work on real-world problems, where they will get direct experience in the Agile mindset. Many educators feel that this is a much better than having their students learn a set of Agile practices merely by reading a book or attending a traditional lecture-based course.

The panel session audience included some faculty members from universities that were still following the model of standard lecture courses. These professors were curious about how they could improve the educational process for their students. Rebecca suggested something simple: "You just announce that you are setting up a practicum with project-based work, and then start doing it in an agile way." A professor could recruit some local small businesses to get involved (to propose some small software development projects) and some students to do the development work (with professors or students acting as Agile coaches). Students will learn a lot from the experience of talking with customers and demonstrating their software to real users.

Werner Wild mentioned his personal experiences with Agile training from his many years of teaching at universities in several European countries. Werner explained the evolution in teaching: "Teaching changed completely from lecturing to walking around. I did a measurement on my last course at TTU in Denmark, we had 70 students spread all over a floor like this here, and I walked between 6 and 10 km every day between the teams." Instead of lecture, Werner's university courses today offer more opportunities for students to work together on real software projects. His role has changed "from hierarchy to face-to-face teaching" and he acts "as more of a coach rather than just presenting the wisdom of previous generations."

Maria and Evelyn explained that companies are often pulled into using Agile by students, and students are finding that companies are looking for interns and new hires with knowledge of Agile.

Maria explained that some companies got interested in Scrum by working with student teams. She recently taught a pilot course where student teams worked on small Agile projects with outside companies. One of the companies was not using Agile, but one person from the company acted as a Product Owner for the student team. "He liked Scrum so much when he saw how the students were working that he decided that they will start using Scrum in their company. Now they are starting to learn Scrum just because of our student team."

Evelyn told the story about markets pushing students to learn Agile. She collaborated with universities at both undergraduate and master level to teach about Agile, Scrum and Product Ownership. “When students were looking for jobs at job fairs, most companies were looking for Agile and Scrum. The university started to feel the pressure, and university folks started to think that we should teach Agile to their students.” Evelyn worked closely with universities to help the situation. The goal was to have students equipped with foundation-level knowledge and practices, ready to contribute.

Maria talked about creative education and training approaches in the Agile world. One questioner was interested in hearing more about the value of “Agile games, workshops, and special learning experiences.” Maria uses Agile games in many of her university courses, and she has written several conference papers on the subject [3]. In the panel session, Maria recounted her experiences at two universities: “Students really like to use the games. They are really engaged. When I have asked their opinion afterwards, and they say that in the game sessions, the time just flies.” She uses games to teach Scrum (using a Scrum Lego game) and Kanban (Test Kanban game).

Werner agreed on the value of games in early Agile training: “Games are one of the things that drew me into field, because it was always fun. You run into things that crazy-looking at first glance, but when you apply it you realize how valuable they are.”

Games can play an important role in educating corporate executives about the Agile mindset. Evelyn Tian explained that after running an Agile game, she would get the executives to do an activity to reflect on the Agile Manifesto – to create their own Leadership Manifesto by writing four new statements in the form “we value A over B” for the context of executive management. When she coaches companies who are considering to adopt an Agile scaling framework, she would ask the managers to reflect and create a Scaling Manifesto. The managers would therefore consider the factors behind the scaling practices, rather than just going with a framework that makes them feel safe and then move into pure implementation mode.

Assigning grades to students for project work is a big challenge. One questioner asked: “So much of Agile is about teamwork and collaboration, yet the university insists on individual assessment. How can we motivate students to really get involved in Agile and really play nice in a team, when we are only really assessing them as individuals?”

Everyone shared some experiences about how they assigned grades for team projects. The grading process shouldn’t be limited to assigning grades to each code module written. Rebecca noted that the learning process isn’t limited to students developing their coding skills: “People pick up skills and contribute in different ways on Agile projects. Not everyone is the fastest coder or algorithm designer, but they can contribute to the team.”

One approach to grading is to assign group grades. When Maria was a professor in Finland, she was able to assign grades by team: “In Finland, we worked it so that every member of a team receives the same grade, unless the team members decide they want to give a lower grade to one person who has not contributed that much and then give another person a higher grade.” When Maria moved to Denmark, she had to change her approach to grading, because educational regulations in Denmark require instructors to assign individual grades. When Werner was a professor in Copenhagen, he had each

team give final project presentations – and there was a fixed set of project questions which they posed to team members in a round-robin sequence. It became very clear when there was a team member who hadn't contributed to the team's final product.

The panelists referred to "Agile mindset" throughout the discussion.

Maria noticed the Agile Manifesto is not a well-known document among the current generation of computer science students. Although most of her students already knew a little bit about Agile when they came to her class, "what they always seem to know is the some of the practices. When I ask them if they know what thoughts underlie the practices – what is says in the Manifesto – they have no idea." She isn't sure how to teach the mindset, but she thinks that the mindset is more important than the practices.

Rebecca and Evelyn both believed that it takes time to teach the Agile mindset, and beginners to Agile may need to get some experience with the practices before their brains can absorb the concepts of the Agile mindset. Rebecca explained "when you are a beginner, you do things without asking why or knowing why." But she also pointed out that some beginners will be curious, and they will ask why. While supporting organizations with Agile transformation, Evelyn said she focuses on encouraged behaviors that are aligned with Agile values and principles, which reduces uncertainties and also builds an environment to experience the values and principles. She explained some of the logic behind experiential learning: "Instead of lecture-based training that tries to convince people that something is important by telling them so, it is better to have them experience the value and get convinced by themselves."

Werner related his experiences teaching Agile concepts to people outside of the software developer community. He has already been successful teaching Agile to management consultants in the Chamber of Commerce in Innsbruck, Austria, and he is looking forward delivering an introductory Agile course to 10-year-olds in Innsbruck in summer 2019.

3 Summary

While the Agile Manifesto will remain unchanged for the foreseeable future, it has already succeeded in creating a revolution in software development practices. Certainly there will be new ways of doing things in the future, so many people will continue to feel the need to write their own version of the Manifesto to apply to their own context. The Manifesto will continue to have an impact on culture, education, and software practices. The panelists focused on Agile's impact on both the content and style of teaching software development – especially in European universities. In Europe, many universities have included Agile development in their curriculum, to meet the demand from industrial companies in Europe. One of the biggest challenges in teaching the practices and values embodied in the Agile Manifesto is how to teach students the Agile mindset.

References

1. Beck, K., Beedle, M., van Bennekum, A., et al.: Manifesto for Agile Software Development (2001). http://agilemanifesto.org. Accessed 19 June 2019
2. Hohl, P., et al.: Back to the future: origins and directions of the "Agile Manifesto" – views of the originators. J. Softw. Eng. Res. Dev. **6**(1) (2018). https://doi.org/10.1186/s40411-018-0059-z
3. Paasivaara, M., Heikkilä, V., Lassenius, C., Toivola, T.: Teaching students scrum using LEGO blocks. In: Companion Proceedings of the 36th International Conference on Software Engineering - ICSE Companion 2014 (2014). https://doi.org/10.1145/2591062.2591169