Agile Practices, Collaboration and Experience

An Empirical Study About the Effect of Experience in Agile Software Development

Martin Kropp[1(✉)], Andreas Meier[2(✉)], and Robert Biddle[3(✉)]

[1] University of Applied Sciences and Arts Northwestern Switzerland, Windisch, Switzerland
martin.kropp@fhnw.ch
[2] Zurich University of Applied Sciences, Winterthur, Switzerland
meea@zhaw.ch
[3] Carleton University, Ottawa, Canada
robert.biddle@carleton.ca

Abstract. Agile Software Development has been around for more than fifteen years and is now widespread. How does experience effect the application of agile methods in organizations and what are the implications on the individual and organizational culture? This paper presents in-depth analysis of the Swiss Agile Study 2014. Switzerland offers an illustrative microcosm of software development, with a range of industry domains and sizes, and well-educated and internationally aware professionals. The study included more than a hundred professionals and managers, contacted through professional and industry associations. The topics addressed included experience with Agile development, motivations for adopting it, barriers perceived, specific practices used, and specific benefits realized. Analysis of the data identified important trends and differences. Agile experience seems to be an important factor, which affects many aspects of practice and workplace culture. More troubling is that it appears stress and overwork may be common among Agile professionals. All these findings illustrate important differences between Agile processes as prescribed, and as actually practiced.

Keywords: Agile · Software process · Collaboration · Organizational culture · Software practices

1 Introduction

After 15 years since the publication of the Agile Manifesto [1], Agile software development has become mainstream. In most recent studies 70 % up to 94 % of the participating companies claim to follow an Agile approach in their software projects [11,15], with Scrum by far the most dominant process identified. These studies report about the distribution and application of Agile practices in companies, and the effects and changes they cause. But as far as we know, there are no studies about if and how the application of Agile methods and practices change

P. Abrahamsson et al. (Eds.): PROFES 2016, LNCS 10027, pp. 416–431, 2016.
DOI: 10.1007/978-3-319-49094-6_28

over time and with experience. Moreover, the Agile landscape has changed. In earlier times, for example, Extreme Programming (XP) advocates often advised novices to learn and apply all XP practices [5,8] whereas more recently, with a greater variety of Agile processes and practices, it has become common for educators and workplaces to make their own selections [6,7] and blend as they see fit. This has meant that there can be a large difference between Agile processes as-prescribed, and as-practiced.

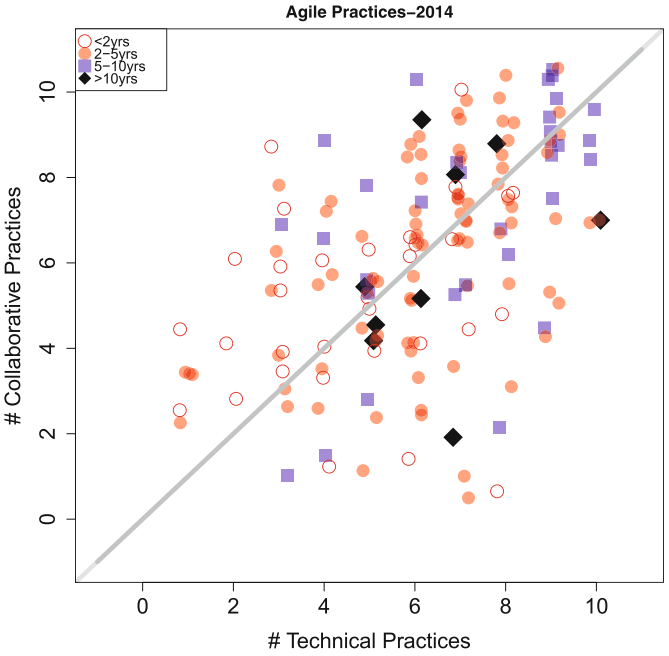

Fig. 1. Number of technical and collaborative practices applied by company.

In this paper we explore the data from the Swiss Agile Study 2014 [11] with respect to the agile experience of the study participants, and in particular we were interested in answering the following questions:

- *RQ1*: Is the usage of Agile practices dependent from the Agile experience of the IT professionals and organizations?
- *RQ2*: Does the application of Agile methods have any influence on the organizational culture of companies and, if so, how does this evolve with experience?
- *RQ3*: What is the influence of Agile on the individual IT professional? Does he work less overtime, has less stress? (as propagated by eXtreme Programming, for example).
- *RQ4*: What improvements are reported by Agile professionals and companies, and are these dependent from their experience?

Our analysis of the data reveals two major issues: The amount of applied Agile practices evolves only slowly with the Agile experience of the companies as indicated in Fig. 1. Second, following an Agile approach over a long time changes the organizational culture in companies.

One indication of the pattern and practices revealed is shown in Fig. 1. For each company surveyed, this shows the relationship of the number of collaborative practices reported to the number of technical practices reported. Points nearer the bottom left show companies reporting few practices, and points near the top right show those reporting many. Points above the diagonal show companies reporting more collaborative practices, whereas those below the diagonal report more technical practices. Each company's experience with Agile is shown by a colored symbol, with the darker colors indicating more experience. There is some variation, but the pattern is evident: in general, more experienced companies report using more practices, and especially more collaborative practices.

Our study data reflects the current state of Agile software development in Switzerland. Since Switzerland offers an illustrative microcosm of software development, with a range of industry domains and sizes, and well-educated and internationally aware professionals, we suggest that this situation might be very similar to other countries. The results of our study show that it takes many years of Agile experience until organizations have adopted the Agile work style. Change to the collaborative practices seems to be especially hard, due to resistance to organizational changes. However, the longer organizations are applying Agile development the more they tend to have a collaboration and cultivation organizational culture. The results also show that it takes many years of Agile experience until Agile development becomes effective; but once Agile has been really adopted it seems to bring improvements across the board. We speculate that if organizations would pro-actively address the organizational change from the beginning, Agile development might become effective much faster and organizations would benefit much earlier.

In the rest of this paper, we will show in detail how the application of the technical and collaborative practices changes with Agile experience, when the major improvements occur, how barriers for further Agile application change, and how the individual professional experiences the transformation to an Agile work style, with all its consequences like personal stress, work life balance, engagement in the project and identification with the work and team. We will discuss the results and their possible consequences for companies introducing Agile approaches.

2 Study Method

The Swiss Agile Study, conducted by the authors, is a biennial Swiss nationwide online survey about the usage of development methods and practices in the IT industry.

It addresses both Agile and non-Agile companies and IT professionals. It comprises a catalog of about 30 questions about applied software development methodology, techniques and practices on technical level, collaborative level and

Table 1. Demographics of participating companies: role, % of participants

Role	%
CEO	29 %
Development manager	14 %
Project manager	8 %
Team leader	7 %
CIO	6 %
CTO	5 %
Senior software developer	5 %
Designer/Architect	2 %
Product manager	1 %
Other	23 %

Table 2. Sizes of the participating companies

Size	%
Micro enterprise (≤ 9)	18 %
Small enterprise (10–49)	29 %
Medium enterprise (50–249)	25 %
Large enterprise ≥ 250)	28 %

value level as outlined elsewhere [10]. The complete study reports are freely available [11]. The latest survey, in 2014, included specific questions concerning the organizational culture in Agile and non-Agile companies according the organizational culture model of William Schneider [12].

101 companies and 128 IT professionals participated in the last survey in 2014. We emailed 1461 companies and about 5000[1] IT professionals in Switzerland. The addresses of the companies and the professionals were delivered from the participating IT associations SwissICT, SWEN and ICTnet, as well as from our own institutional databases. In the company survey we addressed representatives of the company or the development department of a company, i.e. the management level. Table 1 shows the demographics of the company participants. It shows that almost 30 % of the participants were Chief Executive officers. The relatively high number of "others" includes roles like Business Analysts, Business Unit Managers, and CFOs, for example.

Table 2 show the distribution of the sizes of the participating companies following the official categories of the Swiss Federal Statistical Office[2]. The main branches of the companies are IT Services/IT Consulting (40 %), Software Industry/Development (25 %). Medical and Health Care companies and Finance and Insurance companies make 5 % each. The rest are 3 % and below. The responding IT professionals were typically Senior Software Developers (19 %), Software Developers (18 %) and Project Managers (11 %).

[1] We do not know the exact number, since these mailings were partially done by the partner associations.

[2] http://www.bfs.admin.ch/bfs/portal/en/index/themen/06/02/blank/key/01/groesse.html.

3 Findings

3.1 Influence

In this section we look at the influences which Agile has on the software development process, i.e. which aspects in software projects and project management got better or worse and how they change over time. The professionals responding to the survey answered the question: "How has Agile software development influenced the following aspects?" To display which Agile influences are strongest, we use co-occurrence grids, as shown in Fig. 2. Each grid shows the occurrences of each influence, and co-occurrences of each pair of influences; this allows us to see which influences are commonly occurring together. The influences are listed on the left and the bottom of each grid. On the diagonal of each grid, the number of professionals answering the question with *improved* or *significantly improved* is shown on the diagonal, with higher numbers shown in shades of blue, and lower numbers in shades of red. At each grid intersection, the co-occurrence of two influences is shown.

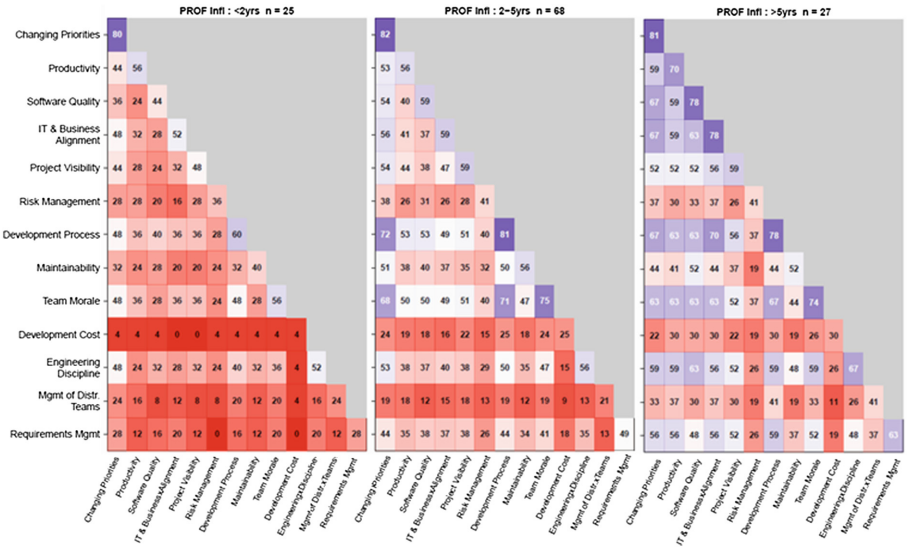

Fig. 2. Agile influence by experience. (Color figure online)

At the left of Fig. 2 are the results for the 25 professionals with less than 2 years Agile experience. The pattern is clear and a bit of a disappointment: Agile has clearly a positive influence on managing changing priorities and, to a lower degree, on team morale, productivity and the development process. The remaining items of the co-occurrence map are mostly red, i.e. those aspects have not improved. In particular, development costs have not improved at all.

Why is this? This is most likely because reducing development cost is not the main objective of an Agile transformation. Instead of reducing development cost, the money is spent on developing more features and better software products. Also note that nobody regarded distributed teams as an influence.

In the middle of Fig. 2 are the results for the 68 professionals with 3 to 5 years Agile experience. With more experience, a notable cluster of positive influences occur. This cluster includes, in decreasing order, responding to changing requirements, development process, team moral, software quality, alignment between IT and business, project visibility, maintainability and engineering discipline. Distributed teams are still not noted as an influence.

At the right of Fig. 2 are the results for the 27 professionals with more than 5 years Agile experience. The pattern is clear: mature teams profit much more of the benefits of Agile. This is reflected in the blue color of the co-occurrence grid. There are only three outliers: development cost, risk management and management of distributed teams. These outliers are a topic for concern and future study.

In summary, the presented data suggests that Agile has a profound positive influence on many aspects of software projects but it takes long time and suggests great effort.

3.2 Agile Practices

In the introduction to this paper we highlighted Fig. 1 showing the number of practices varied with experience, and also the balance between categories of practices: technical and collaborative practices [10]. The practices we enquired about are shown in Table 3. The practices comprise those recommended by eXtreme Programming [4] and by Scrum [13], plus new practices that have come since then like Continuous Delivery, Acceptance Test Driven Development, Behaviour Driven Development. To explore the relationship between experience and practices more closely, we now show which practices are used with which levels of experience.

At the left of Fig. 3 are the results for the 34 professionals with less than 2 years Agile experience. The pattern is clear: at the top in light blue are a number of commonly used technical practices: unit testing, automated builds, coding standards, and continuous integration. Lower down, the use of user stories, daily standups, and task boards are indicated. But otherwise, the grid is predominantly red, showing very little usage of many practices, especially the collaborative practices such as pair programming, collective ownership, retrospectives, and others.

To display which practices were used by each group, we again use co-occurrence grids, as shown in Fig. 3. In the middle of Fig. 3 are the results for the 93 professionals with 2–5 years Agile experience. Here the pattern is somewhat different. The practices used in the grid at the left are now more commonly used, as indicated by the darker shade of blue: both for the technical practices at the top, and the collaborative practices such as user stories and daily standups.

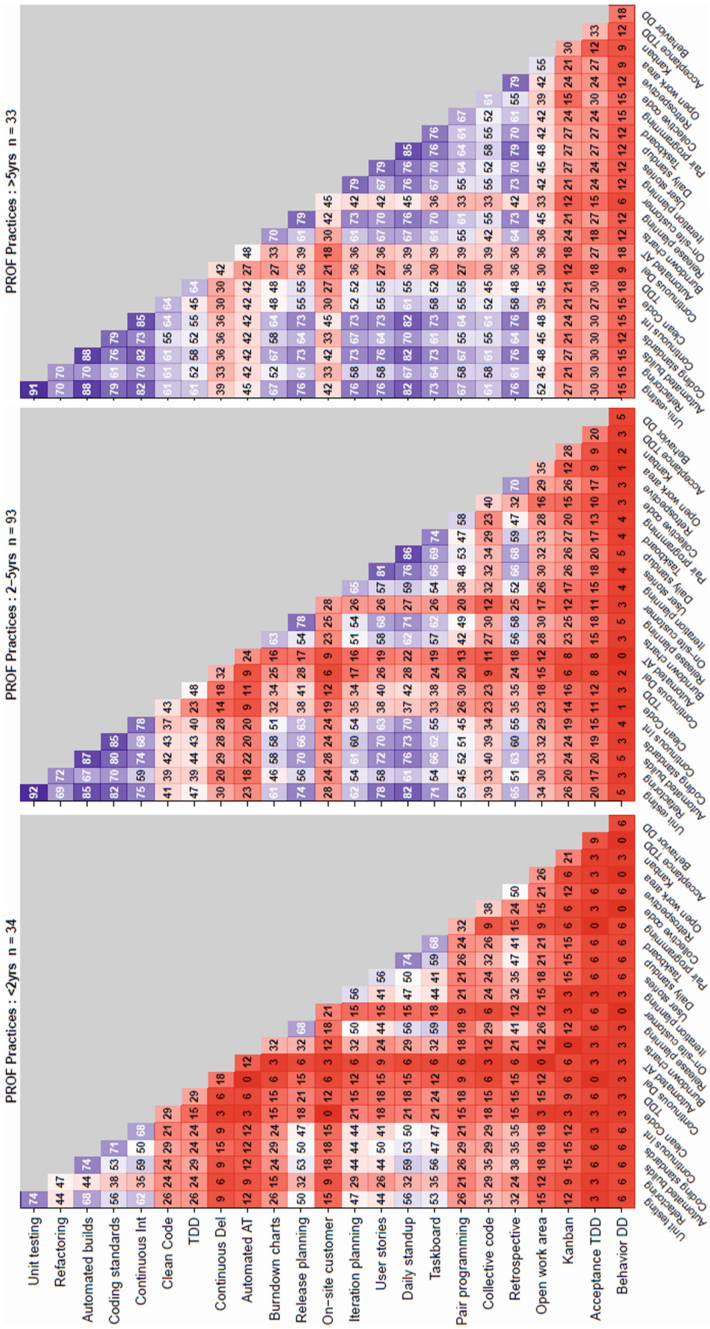

Fig. 3. Agile practices by experience. (Color figure online)

Table 3. Agile practices

Practice	Type
Unit testing	Technical
Refactoring	Technical
Automated builds	Technical
Coding standards	Technical
Continuous integration	Technical
Clean code	Technical
Test Driven Development (TDD)	Technical
Continuous delivery	Technical
Automated acceptance testing	Technical
Burndown charts	Technical
Release planning	Collaborative
On-site customer	Collaborative
Iteration planning	Collaborative
User stories	Collaborative
Daily standup	Collaborative
Taskboard	Collaborative
Pair programming	Collaborative
Collective code ownership	Collaborative
Retrospective	Collaborative
Open work area	Collaborative
Kanban pull system/Limited WIP	Advanced
Acceptance Test Driven Development (ATDD)	Advanced
Behavior Driven Development (BDD)	Advanced

But now there are other practices also indicated. Practices shown include burndown charts and release planning, retrospectives, and also use of pair programming.

At the right of Fig. 3 are the results for the 33 professionals with more than 5 years of Agile experience. Again, the pattern is different, but continues the same trends. Practices with light usages reported by the other groups are now much more common, and practices such as collective code, test-driven development (TDD), and clean code are strongly indicated.

Overall, the pattern is strikingly clear: professionals with more experience report applying considerably more practices, more consistent use of practices, and more use of related practices. Moreover, while those with less experience report principally technical practices, more experience brings use of more collaborative projects.

3.3 Barriers

Our basic study [11] revealed that the ability to change organizational behavior was listed as the greatest barrier to adoption of Agile software development in an organization. We were also interested whether the barriers reported vary with Agile experience within organizations, and if so, in which direction. Figures 4 and 5 show the relevant data for the IT professionals and companies. The figures show which three barriers were most important.

The figures reveal that the change within the organization is by far the most critical issue when companies switch to Agile, especially at the beginning. More than 50 % of the IT professionals and more than 40 % of the companies ranked this issue as the greatest barrier. We find it interesting that this issue remains among the greatest barriers even in companies experienced with Agile methods. The figures also show that other issues become more important. However the views of the professionals and the companies differ. Both see customer collaboration becoming an important issue. But while the experienced professionals rate the lack of skilled personal as critical, the managers of experienced companies suggest the handling of complex projects as the major barrier for further adoption of the Agile methods.

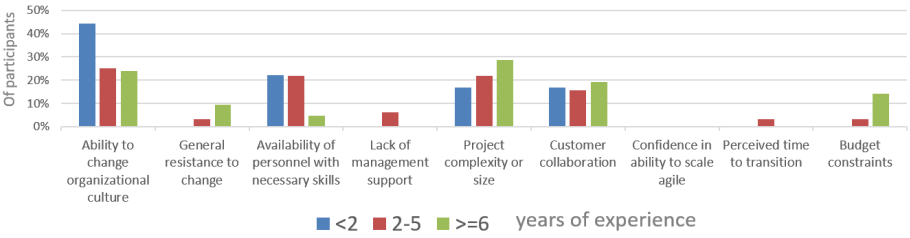

Fig. 4. Barriers by companies

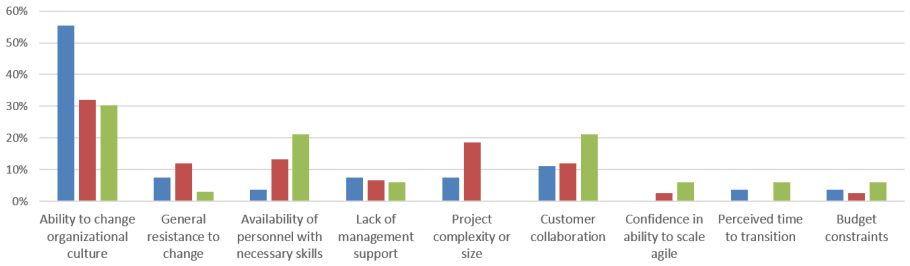

Fig. 5. Barriers by IT professionals (Legend as in Fig. 4)

3.4 Organizational Culture

We examined the organizational culture of the participating companies, applying the organization model of William Schneider. Schneider identifies four different organizational cultures, the control culture, competence culture, collaboration culture and the cultivation culture [12]. To identify culture, we used the questionnaire from Schneider's book [12]. We were using ten out of 20 of William Schneider's questions about which organizational culture the participating companies exhibit. In this approach, the answer to each question identifies which of the four culture categories is indicated, and so the overall response yields four numbers, one for each culture. In our results it was most common for one particular culture to dominate the others by three or four points, and we therefore chose that as the dominant culture. We then evaluated if the organizational culture depends on the Agile experience. Figure 6 shows that, as experienced by IT professional, organizations start with the preference of the traditional control and competence culture, which changes with more Agile experience towards a cultivation and collaboration culture. This seems to imply, that following an Agile approach over a long time changes an organization's culture to collaboration and cultivation culture.

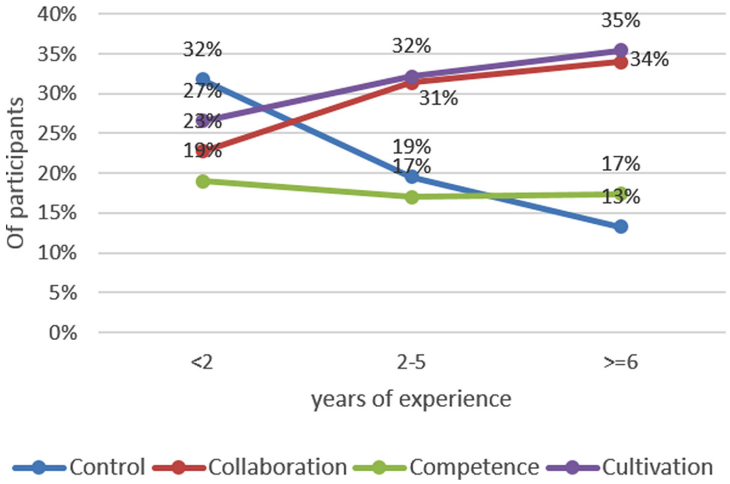

Fig. 6. From a control and competence to a collaboration and cultivation culture.

3.5 My Agile

One section of the survey asked professionals to reflect on their personal experiences with Agile development; we called the section "My Agile". Whereas earlier we had asked about the influences they saw, this set of questions asked about their actual personal experience. We suggested a range of possibilities, and asked whether it applied in their case.

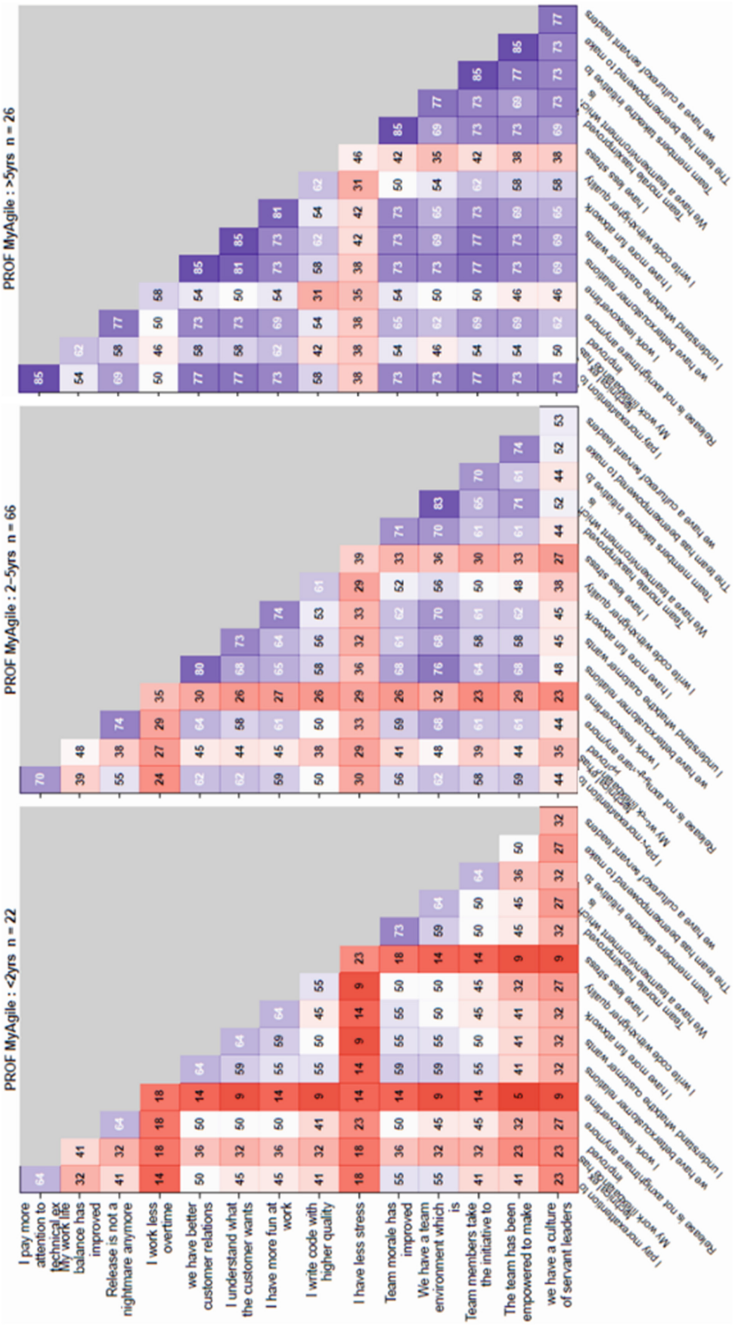

Fig. 7. My Agile by experience. (Color figure online)

The data is illustrated in Fig. 7. We again show co-occurrence grids, and distinguish professionals with less than 2 years, between 2 and 5 years, and more than 5 years. Again, the grids show shades of blue for higher reported factors, and red for those reported less. The patterns and differences are clear.

The grid on the left shows the data for the 22 less experienced professionals. The most reported individual element was team morale, and other common elements reported include technical excellence, release stability, customer relations, more fun, better team environment and initiative. There also are some patterns where several of these factors are reported. The most striking part of the grid, however, are the entries highlighted in red, indicating very few responses. Most prominent were "working less overtime" and "I have less stress". There are also absences shown for team empowerment and servant leadership.

In the grids in the middle (2–5 years) and the right (more than 5 years), we see remarkable changes. In particular far more factors are reported, and the most experienced professionals show many reports of almost all of the elements. However, in both cases, the weakest elements reported are the same as for less experienced professionals.

The progression from less to more experience clearly shows an increasing appreciation of a greater range of advantages, and that seems reasonable. However, the emphatic indications about overtime and stress lack are concerning. In Beck's books about Extreme Programming [3,4], there were specific recommendations about what in the second book was called "sustainable pace". It appears this is not occurring, especially for less experienced professionals, and perhaps even for those with more experience. It is interesting to note that the stress and overtime are happening at the same time that high team moral and fun are both reported, suggesting a mixed picture of positive and negative aspects.

The milder issue raised by the data involves team empowerment and "servant leadership". This concept is emphasized in several Agile processes, including Scrum and XP, neither of which suggests a manager in any traditional sense. This is a long history of approaches to socio-technical teams [14] and self-organizing teams [9], and a wide range of related thought before and since. However, our data suggests this is seen as one of the least common benefits of Agile processes. Further work is necessary to establish why this is so: Are professionals not concerned? Or has it simply not been possible? If, so, what were the circumstances and consequences?

3.6 Quality Control

Figure 2 showed that significant improvements in software quality come rather late. Accordingly, we were also interested if there is a relation between quality control and the late quality improvement, and if quality control is also dependent on Agile experience.

Figure 8 shows the corresponding data for the IT professionals. We focus on the data from the professionals because the quality control measures we explore are at a detail level commonplace to professionals – managers may be unaware of tools at this level.

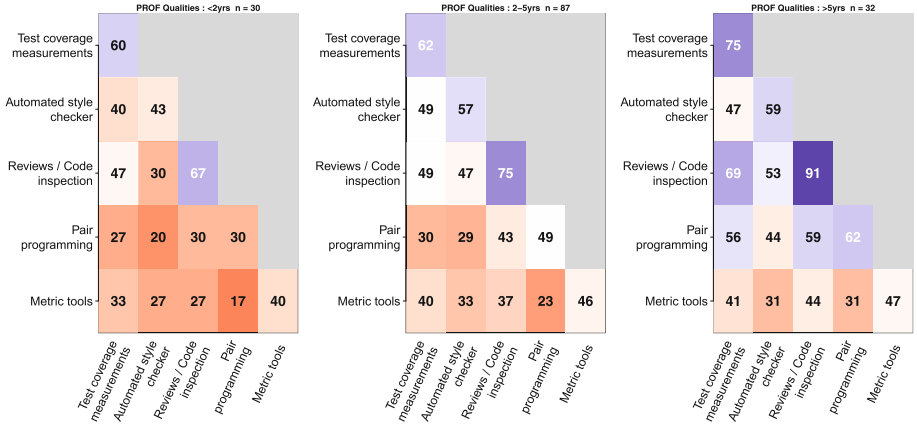

Fig. 8. Agile quality control by experience. (Color figure online)

At the left of Fig. 8 are the results for the 30 professionals with less than 2 years Agile experience. The grid shows that these professionals just start using code coverage and code reviews for quality control as indicated by the light blue color. The application of other measures is low (light red to dark red color). In the middle of Fig. 8 are the results for the 87 professionals with 2–5 years Agile experience. Here the pattern is a little different. The measures used before are now more common (dark blue color) and also the usage of style checker tools has increased. At the right of Fig. 8 are the results for the 32 professionals with more than 5 years of Agile experience. Here the pattern changed significantly. The numbers of the used tools have further increased, and also Pair Programming is now widely applied. Quite striking is the very high application of code reviews (91 %).

Again, the pattern is clear: the usage of quality control tools increases with experience. However, it takes more than five years of Agile experience until the majority of the listed quality control measures are applied by the majority of the professionals.

4 Discussion

Our goal in this paper was to study how Agile methods as-prescribed might differ from Agile methods as-practiced and how Agile evolves with experience in organizations. In the previous sections we have presented our main findings with respect to the research questions formulated in Sect. 1, and we will now attempt to articulate answers to these questions:

Agile and Workplace Culture: One of the issues addressed in the 2014 survey was workplace culture. In particular, the model of Schneider [12] was used, and questions from Schneider's book were included in the survey. The model suggests 4 kinds of workplace: Control, Collaborative, Competence, and Cultivation. We

did find that most sets did emphasize one of these, and the responses did align with responses to questions about Agile methods: see Sect. 3.4.

Agile Changes with Experience: Perhaps the most wide-ranging finding from our survey was how the responses to a range of questions related to experience with Agile. In Sect. 3.1 we showed differences in reports of influences of Agile processes, and in Sect. 3.2 we showed how the practices used varied from a few mostly technical practices for beginners, to a wide range of practices for those with more experience. As we outlined in the introduction, in the early days after the Agile Manifesto, it was common to advocate following all the practices, especially for beginners. More recently it has become common for workplaces to choose their particular process, mixing elements from Scrum, XP, and a range of sources. While our results show a clear pattern, we do not know why that pattern emerges, what the implications are, and what, if anything, might be preferred.

Agile Influence for Improvements: In Sect. 3.1 we showed that it takes very long until Agile development becomes effective in Agile organizations showing significant improvements. While there are immediate improvements with handling changing priorities, only in more experienced Agile organizations we see improvements in requirements management, software quality, productivity, and in the engineering discipline. This long-term benefit may be related to the late application of many practices, but it might also reflect the strong resistance to organizational changes, especially when starting with the transformation to Agile. It could be a matter of further research to find out why the improvements come so late, and if it is possible to shorten this path to success, and if so, how it could be done.

Agile is Commonly Stressful: In Sect. 3.5 above, we examined the results from the survey where professionals reported their own reflections. Surprising to us was that the topics least agreed with were all related to stress or overwork. This seemed odd, considering that time-boxed iterations, collaborative environments, and self-management are all Agile practices, and would appear beneficial and lead away from stress. It appears this might not be the case. A potentially important observation is that the professionals also reported that team moral was high. This combination has been suggested before, in the 2007 study of social factors in Agile teams by Whitworth and Biddle [16]. Quotes from that study reflect a kind of characteristic zeal with a dark side: "This isn't a place that you go and hide". Moreover, they point to research by Barker [2] on self-managing teams, entitled "Tightening the Iron Cage: Concertive Control in Self-Managing Teams". That research shows how the strong social control in teams leads to a peculiar kind of stress that is more intense because it is connected to social factors in team commitment. Barker's term "Iron Cage" is a sobering indication that this aspect of Agile processes needs more attention.

Our aim for the survey was to be descriptive, so we must consider threats to external validity, and especially factors that might make our results differ from reality. One such factor is self-selection, and this might lead to more participants with responses they strongly wish to share: these might be positive or negative, and might therefore exaggerate results both ways. Another factor might be

uneven response from different workplaces, especially in responses from professionals, where multiple participant might describe the same workplace. This was not possible for the company responses, but there still might be more responses from some industries or areas than others.

When analyzing the data, we noticed that the results from the IT professionals partly differ significantly from those of the managers. This is very obvious in Figs. 4 and 5, in which both groups partially have quite different opinions on the barriers for further adoption of Agile methods. We feel these do not suggest inconsistency, but rather a different perception of their corresponding environments. While management may often have a high-level view on the business, the IT professionals are dealing with everyday detail issues. Another surprising result came up in the response to influence of Agile on various aspects in Fig. 2. While the majority of experienced participants report improvements in requirements management and in handling of changing priorities, and report applying the retrospective practice in Fig. 1, they report no improvements in risk management. We assume that this question has been misunderstood: While there is no explicit "Risk Management" activity foreseen in Agile methods, short iterations, early feedback and retrospective are risk management per-se. So we might have to change this question in future surveys.

5 Conclusions

In this paper we addressed the issue of how Agile software development processes are actually used and viewed, showing how the process as prescribed differs from the process as practiced. We presented data from the 2014 Swiss Agile Survey, with responses from 101 companies and 128 professionals in Switzerland. The survey covered a wide range of topics, from influences of Agile, to practices actually used, barriers perceived, and reflections on personal experience.

Our main findings show several themes. One is that experience with Agile methods is an important factor. More experience is related to a greater number of practices used, greater emphasis on collaboration, and more sustainable workloads. Another finding is that, despite Agile principles such as "sustainable pace" and strict time-boxing of iterations, there are warnings that stress and overload remain a problem, especially for those new to Agile, but sometimes even for experienced professionals. This is despite indications of high team morale.

There are limitations in our survey approach, because of self-selection and of inability to follow up interesting results immediately with questions to probe details and causality. In addition, the granularity in some questions might be improved, for example to better gauge experience on a finer scale.

In future research on this topic, we suggest that questions of causality should be the priority. In particular, it would be interesting to know why some practices appear only with more experience. In particular, practices around customer collaboration are seen as very important, but only arise with more experience. Is it that only then is the importance realized, or are there barriers that prevent it by those with less experience? Also, it would be important to explore causes

of stress and overwork. Is it simply enthusiasm stemming from high morale, or perhaps is it lack of familiarity with certain practices?

References

1. Agile Manifesto Signatories. Agile Manifesto (2001). http://agilemanifesto.org
2. Barker, J.R.: Tightening the iron cage: Concertive control in self-managing teams. Adm. Sci. Q. **38**(3), 408–437 (1993)
3. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley Longman Publishing Co., Inc., Boston (2000)
4. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley Professional, Reading (2004)
5. Bergin, J., Caristi, J., Dubinsky, Y., Hazzan, O., Williams, L.: Teaching software development methods: the case of extreme programming. SIGCSE Bull. **36**(1), 448–449 (2004)
6. Diebold, P., Dahlem, M.: Agile practices in practice: a mapping study. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, pp. 30:1–30:10. ACM, New York (2014)
7. Diebold, P., Ostberg, J.-P., Wagner, S., Zendler, U.: What do practitioners vary in using scrum? In: Lassenius, C., Dingsøyr, T., Paasivaara, M. (eds.) XP 2015. LNBIP, vol. 212, pp. 40–51. Springer, Heidelberg (2015). doi:10.1007/978-3-319-18612-2_4
8. Goldman, A., Kon, F., Silva, P.J.S., Yoder, J.W.: Being extreme in the classroom: experiences teaching XP. J. Braz. Comput. Soc. **10**(2), 4–20 (2004). http://www.swissagilestudy.ch/files/2015/05/SwissAgileStudy2014.pdf
9. Hackman, J.R., Oldham, G.R.: Work Redesign. Addison-Wesley, New York (1980)
10. Kropp, M., Meier, A.: Teaching agile software development at university level: values, management, and craftsmanship. In: Software Engineering Education and Training (CSEE&T), pp. 179–188. IEEE, May 2013
11. Kropp, M., Meier, A.: Swiss agile study 2014. Technical report, Swiss Agile Study (2014). http://www.swissagilestudy.ch/files/2015/05/SwissAgileStudy2014.pdf. ISSN: 2296–2476
12. Schneider, W.: The Reengineering Alternative. McGraw-Hill Education, Columbus (2000)
13. Schwaber, K., Beedle, M.: Agile Software Development with Scrum 1st edn. Pearson (2001)
14. Trist, E.L., Bamforth, K.W.: Some social and psychological consequences of the longwall method of coal-getting. Technol. Organ. Innov.: The Early Debates **1**, 79 (2000)
15. VersionOne. 9th state of agile survey. Technical report, VersionOne Inc. (2015)
16. Whitworth, E., Biddle, R.: The social nature of agile teams. In: Proceedings of the AGILE 2007, AGILE 2007, pp. 26–36. IEEE Computer Society, Washington, DC (2007)