

Characterizing Technical and Organizational Practices in the Agile Community

Nicolás Paez, Diego Fontdevila, Alejandro Oliveros

Departamento de Ciencia y Tecnología

Universidad Nacional de Tres de Febrero

nicopaez@computer.org, dfontdevila@untref.edu.ar, aoliveros@untref.edu.ar

Abstract

Agile software development has grown tremendously in popularity since the publishing of the Manifesto catching the attention of industry and academia. Agile proposes a holistic vision of software development that includes organizational and technical practices. This study aims to characterize the usage of these practices in the Agile Community. The study is based on a survey ran among the participants of an agile conference. The results indicate that organizational practices are used far more than technical practices. At the same time use of organizational practices tends to be more compliant than that of technical practices.

1. Introduction

Agile software development has grown tremendously in popularity since the signing of the Agile Manifesto [1] and has become mainstream [2]. Practitioners and researchers alike have become interested in agile adopting agile methods and practices. The current popularity of agile can be evaluated by the great offer of conferences, books and training courses, and by the amount of industry consultants and companies deeming themselves agile [2][3]. Even formal education has begun to include agile in their curricula [4]. Researchers have also put significant energy trying to understand agile and its success [5][6][7].

Agile is a holistic vision of software development, not just a set of methods, and it has been remarked by both practitioners and researchers that it can be simple but at the same time hard to put in practice. It is not only a matter of understanding the profound underpinnings of agile, its values and principles [5][8], but also learning the practices and applying them effectively. As with many other successful ideas, agile methods tend to be attractive and therefore their adoption is attempted by many, but not necessarily completed successfully. Along the years there has been deep controversy about the value of partial or incorrect implementations of agile, characterized for example by Martin Fowler's "Flaccid Scrum" article [9] or Vas Bodde's "Nokia Test" [10]. At the same time, slight variations have emerged in the

process of implementing the more established methods. For example, a slightly simplified version of Scrum, with backlog item estimation but without estimation of tasks is very popular and has been assessed as more productive by [3]. Also, newer agile trends like Mob-programming, Continuous Delivery and "No Estimates" complement a rich and complex landscape of practice.

As we have seen, characterizing the state of agile practice has been and is currently a concern of several communities. We have selected a subset of the core agile practices, leaving out the more subtle principles and values, for example self-organization, and also some of the modern practices we have seen in agile teams like Continuous Delivery and Infrastructure as Code. In the next section we describe how we have organized the practices in our study.

To understand the state of agile software development practices we ran a survey at an agile conference. We asked participants about the practices of a representative project of their organization. It was a first approach or preliminary study to this field with the purpose of supporting the generation of hypotheses for future research.

In the rest of this paper, section 1.2 introduces the concepts of technical and organizational practices in agile software development. In section 1.3 we discuss the goals of this study. Section 2 summarizes the results of similar studies. Section 3 describes our research procedures in this study. Section 4 is the core of this paper. It contains the results in terms of demographic information and about technical and organizational practices. We conclude the section with relevant findings. In section 5 we describe our next steps in this field.

1.1. Agile is both Organizational and Technical

Agile addresses both technical and organizational aspects of software development [5][11]. The Agile Manifesto stresses both aspects in its main statements and principles. For example:

"Continuous attention to technical excellence and good design enhances agility.
[...]"

The best architectures, requirements, and designs emerge from self-organizing teams." [1]

The different agile methods tend to have a predominant focus on organizational or technical aspects, although some are more balanced in this respect. For example, Scrum, the most popular agile method today [12] is defined in terms of organizational concepts, leaving out any and all technical practices. Nonetheless, studies of Scrum teams also evaluate technical practices [13], and Scrum trainers stress the need for those practices even if they are not defined in Scrum. At the same time Extreme Programming has strong technical practices [14] but also addresses organizational issues (for example, planning). Finally, Crystal [15] is an example of a methodology family balanced between organizational and technical concerns.

This duality is relevant because the absence of any of these two aspects has a huge impact on projects' results. If the technical side of agile is ignored then software quality and maintainability will suffer (Chow et al [16] found that engineering practices are one of three key success factors of agile projects). At the same time, the absence of organizational practices impacts on project costs, deadlines and excessive pressure on development team members. It might also be argued that lack of organizational practices might also impact quality and, conversely, lack of technical practices might affect predictability.

It is interesting that even though researchers acknowledge this dual nature of agile, they do not consistently categorize practices as technical/organizational. For example, Bertrand Meyer considers Continuous Integration an organizational practice [5], while we consider it a technical practice (see the list of sub practices in [17], most sounding technical).

1.2. Goals of the Study

The goal of this study is to characterize the agile practices of the local agile community, helping practitioners, agile consultants and researchers by making visible the state of agile practice. The research questions are: which of the core agile practices are actually used in industry agile projects? Are both technical and organizational practices used in industry agile projects?

Practitioners should profit from a deeper understanding of what their community of practice (their peers) are doing, using that as both inspiration and challenge, and maybe as validation of their own ways of working. Agile consultants might find new insights or confirmations of their intuitive perceptions of how practices are adopted and institutionalized. Researchers should find valuable information, interesting findings and opportunities for further research.

This work is also part of a larger research project on Usability of Process and Practice, with the objective of exploring the application of usability principles and heuristics to the understanding and study of process and practice adoption, design and evolution in software development organizations. The resonance between usability and software development improvement terminology (e.g. terms like feedback, tolerate mistakes and simplicity appear recurrently in both contexts with similar meanings) marks the way for valuable research on the problem. The overarching idea is that workers are users of their methodologies and ways of working. As one example of methodology analysis in terms of its relationship with its practitioners, Alistair Cockburn has reflected on the concept of high-discipline methodologies ([18] see the section "XP under the glass") which he defines as those that might probably be abandoned if a mechanism to keep it up is not put in place, an example of such mechanism is the Coach role in XP.

As a result of our study, based on a survey ran in the context of an Argentinian agile conference, we describe the characteristics of agile practices usage and conclude that the differences between the usage of technical and organizational practices is significant. Such marked differences as we have observed require further study, but preliminary results match previous studies and our expectations in terms of practice usability, with low usability practices like TDD ranking among the lowest and high usability practices like retrospectives ranking among the highest.

2. Related work

In coincidence with the goals of our study, we include work related to agile practice use, not going into any depth about the nature of the practices themselves.

A study analogous to our own was conducted in the agile community in Spain [19], with strikingly consistent results and both surveys having an Open Space conference as their context. Their work differs from ours in that it compares their results to the state of practice of several European organizations taken as reference. The study included around 100 participants.

In South America, a broad survey study on the historical evolution of the agile community was conducted in Brazil [20] including industrial and academic contexts, but it did not cover specific practices, dealing with agile methods in a more general way.

There are several industry surveys describing agile practices adoption [13][16][21] which show marked differences between practices (these studies were conducted as online surveys). For example, [21] ranks TDD as the most difficult to implement agile practice (the study is based in 123 online responses). As we have mentioned, Alistair Cockburn [18] analyses XP in terms

of practitioners' ability to sustain specific practices. Dybå et al [22] have made a systematic survey of empirical studies on agile, finding relatively few empirical studies, and they state both industry and researchers as their target audience. The already mentioned study by Chow et al [16] analyzed success factors in agile projects, searching for empirical evidence to support the popular belief in agile's ability to promote success. Their survey found three key success factors: delivery strategy, agile software engineering techniques and team capability. In our study we have evaluated Frequent Delivery and technical practices, covering two out of three of these success factors.

Diebol et al [23] studied what practitioners vary when using Scrum by using a survey and a set of interviews to validate answers. Given the nature of Scrum, the study is focused on organizational practices.

Kurapati et al [24] conducted a survey to analyze agile practices adoption and its level of compliance with Scrum and XP. They found that Scrum adoption tends to be more compliant than XP adoption.

3. Methodology and Study Description

In order to characterize use of agile practices we run a survey among the participants of the Agile Open Camp 2016 Conference [25] organized by the Argentinian agile community [26]. The conference was held in Lago Escondido, a conference center placed in a rural location, 80 kilometers from El Bolsón, the nearest city. The conference lasted 3 days and gathered about 100 participants. The conference format was Open Space mixed with a set of 5 predefined keynotes. We collected 44 questionnaire responses, which is comparable to samples in other studies [19][21], corresponding to 42 different projects belonging to 32 different organizations. We explicitly asked participants to answer the survey considering a project that fulfilled the following criteria:

- The project should be representative of their organization.
- They should have been actively involved on the project.
- The project should have been completed within the past year.

The survey grouped questions in three sections: demographic information, organizational practices and technical practices. To select the target practices we considered the study by Bertrand Meyer published in his book Agile! [5] and our own combined 15 years of experience with agile. Bertrand Meyer categorizes agile practices into 2 groups: organizational and technical, with 5 practices in each group. We decided to limit the amount of survey questions to avoid dissuading participants from completing it, and that meant limiting the amount of practices we would include in the study.

Given that we wanted to include more than one question per practice and that we wanted to limit the amount of questions we chose 3 technical practices and 3 organizational practices. As we have seen, we considered Continuous Integration a technical practice while Meyer considers it an organizational practice.

The criteria we used to select the practices were:

- Should be among the most characteristic of agile.
- Should be used generally beyond any specific agile method.
- Should be reasonably simple to evaluate.

Table 1 lists the practices we finally focused on:

Table 1. Selected practices.

Organizational practices	Technical practices
Iterative process	Continuous integration
Frequent releases	Test automation
Retrospectives	Test-Driven Development

In order to characterize the adoption of each practice we included 2 kinds of questions for each practice: a direct question and one or more validation questions. For example, talking about continuous integration our direct question was:

Do you do Continuous Integration?

- Yes
- No
- I don't know / I don't have an answer

Figure 1. Direct question

One of our validation questions was:

How often do you run your integration build?

- Once a week
- Several times a week
- Once a day
- Several times a day
- I don't know / I don't have an answer

Figure 2. Validation question

In our analysis we evaluated the responses to the validation questions to determine the actual significance of the responses to the direct question. For example, we consider that only those that do integration at least once a day are doing Continuous Integration [17].

To host the survey we used Google Forms, an online tool that supports the edition and publication of forms with multiple types of questions whose answers can easily be exported to a spreadsheet.

The survey was announced at the opening of several sessions of the conference and was filled by participants using their own computer in most of the cases.

Once all the data was collected, we exported it to a spreadsheet and performed basic deputation. We analyzed demographic data in order to ensure the answers belonged to different projects. In this process we removed 2 records (out of 44 records collected) because we suspected they belonged to the same project.

4. Results and Findings

In this section we describe the results grouped by our three questionnaire sections.

4.1. Demographics

About 60% of participants identified themselves with technical roles (architects and/or developers) while the rest identified with non-technical roles like coaches, project managers and analysts. Talking about their personal experience using agile, 45% reported less than 3 years, while the remaining 55% reported more than 3 years of experience.

The distribution of agile experience among organizations shows that almost a third of projects belongs to organizations with more than 3 years of experience, while another third belongs to organizations with less than 1 year of experience. Almost 40% of organizations have between 1 and 3 years of experience.

Table 2. Organization's experience with agile (years)

$X < 1$	$1 < x < 3$	$3 < x < 10$	$X < 10$
10 %	8 %	10 %	10 %

Regarding organization's type, 48% of the organizations are software factories, 10% are IT consultancy service providers and the remaining 43% belongs to other kinds of businesses whose core activity is not software development or consultancy.

Table 3 shows the size of the organizations' IT departments.

Table 3. Organization IT department size (people)

$x < 10$	$10 < x < 25$	$25 < x < 100$	$100 < x < 500$	$500 < x$
10	8	10	10	4

Tables 4 and 5 summarize project characteristics:

Table 4. Project duration (months)

$X < 3$	$3 < x < 6$	$6 < x < 12$	$12 < x$	No answer
2	12	9	16	3

Table 5. Project size (people)

$x < 4$	$4 < x < 8$	$8 < x < 15$	$15 < x$
17	11	9	5

4.2. Organizational practices

To evaluate the usage of an iterative process we asked participants about the length of their iteration: 71% of participants reported working in an iterative way. The most common iteration length was 2 weeks (~43 %) which matches the Scrum Alliance Report [13]. The validation question in this case inquired about the way they usually proceed when the iteration ends and the committed scope is not complete. Most participants reported closing the iteration as planned no matter what the completion status of the work being done. Some reported working extra-hours or extending the iteration in order to complete the committed scope, which is not aligned with the agile mindset (see the Energized Work practice in XP [Beck XP Explained]). Applying these criteria, we consider that only 62% of projects work iteratively. Those that do not work with iterations are split between those who limit the work in progress and those that don't. Limiting the work in progress is a common practice used by many companies working with Lean/Kanban approaches which are both under the agile paradigm. Even more, for some authors this approach of continuous flow with limited work in progress is considered a high maturity agile practice. In our case we have 12% of projects working in continuous flow and limiting the work in progress (as opposed to using iterations).

In the case of frequent releases, 83% of participants stated that their projects followed the practice, with the most common frequency being once every couple of iterations. The validation question in this case focused on the review activities because we consider that the value of releases without customer validation is very low and might mask other dysfunctional behaviors (for example, releasing buggy code into production tends to force a new release to provide a fix). In this case some participants stated their project did not perform review activities, so based on that we conclude that only 76% of projects actually perform Frequent Releases in the sense described by agile methods.

Retrospectives are said to be used by 64% of projects but in the validation question we asked about the frequency of this activity and some participants stated their project only included one at the end of the project which means that they are doing post-mortem analysis,

not agile retrospectives. So for our study we conclude that only 57% of participants are doing retrospectives. It is worth mentioning that the direct question in this case was more generic, about continuous improvement activities, where Retrospectives were included as one of the options. Because of this, we have projects that do not use retrospectives but use some other kind of continuous improvement activities like coaching sessions and communities of practice gatherings.

4.3. Technical practices

When responding about Test automation 75% of participants stated their projects have automated tests (direct question) but when analyzing the validation questions we found that only a third of them keep most of their automated tests updated. Having automated tests that are not updated when the behavior they test changes provides very limited value and is symptomatic of abandoned practice. In particular, we assume that out-of-date automated tests eventually break or lower their coverage up to a point where they lose most of their significance. So we conclude that only 38% of participants have meaningful automated tests. We also asked about test coverage but the resulting data did not provide any more insight into the issue.

Continuous Integration is reported as being used by 43% of participants but according to the validation question a third of them do not integrate daily which violates the definition of Continuous Integration, as put forth by Martin Fowler [17]. So we consider that only ~30% of projects do Continuous Integration.

Test-Driven Development (including its variations like BDD) is the least used practice in our study: 67% of participants report explicitly not to use it while only 11% use it in a significant way. This low adoption of TDD is aligned with the results presented by Rodriguez [19]. One cause of this could be found in [21]: Test-Driven Development ranks as the most difficult agile practice to learn. In terms of practice usability, this is also consistent with our preliminary research which characterizes TDD (and Test-First in general) as a "low" usability practice, mostly because it changes radically the conceptual model of when tests are defined (before writing the code) and what their purpose is (they become a design tool rather than just existing to find bugs).

4.4. Rank of practices

Table 6 shows the ranking of practices based on their usage:

Table 6. Practice Ranking

Practice	Usage %
Frequent Releases	76%

Iterative process	62%
Retrospectives	57%
Test automation	38%
Continuous Integration	31%
Test-Driven Development	11%

From the ranking above we clearly see that all organizational practices are used in more projects than any technical practice. Even more, all organizational practices have an adoption rate over 57% while technical practices have at best an adoption rate of 38%.

4.5. Relevant findings

As we have seen, we found that organizational practices are used in more projects than technical practices. We also found that the stated level of practice usage tends to be higher than the level we evaluate with our validation questions. Table 7 shows the comparison between stated practice and validated practice usage.

Table 7. Practice Usage

Practice	Stated use (Direct questions)	Validated (Validation questions)	Relative % of validated/stated
Frequent Releases	83%	76%	92%
Iterative process	71%	62%	87%
Retrospectives	64%	57%	89%
Test automation	75%	38%	51%
Continuous Integration	42%	31%	74%
Test-Driven Development	33%	11%	33%

These results show a consistent overestimation of agile practice in use, which is consistent with the aforementioned controversies about variations of agile practice and the tension between the apparent simplicity of agile and the actual challenges of its adoption in practice. It is also noteworthy that technical practices tend to be overestimated more than organizational ones, they have a lower relative level of validated usage than organizational practices (92% for Frequent Releases vs. 51% for Test Automation). Again, this maintains the

¹ As we have seen, there is an additional 12% that works in a continuous flow mode while limiting the work in progress, which is considered a lean/agile best practice, although not actually iterative

trend of technical practices being harder to adopt, in this case, harder to follow respecting their underlying principles. This might also mean that they are less well understood by the general practitioner community, or in general harder to learn. We find this is consistent with our ongoing research on process and practice usability.

Based on our preliminary observations, we pursued the analysis of characteristics that seemed to imply increased practice usage. Along those lines we found two interesting patterns, which relate Retrospectives to organizational practices and Years of organizational experience with agile to technical practices.

In those projects belonging to organizations with more than 3 years of experience using agile we found significant increases in technical practice usage. In particular:

- 75% has automated tests (vs. 38% for the sample)
- 69% uses continuous integration (vs. 31% for the sample)

This might be explained by the fact that although technical practices might be harder to adopt, an organization persisting in its adoption of agility might eventually find its way to implementing them.

In those projects which use Retrospectives we found significant increases in organizational practice usage. In particular:

- 84% use an iterative process (vs. 62% for the sample)
- 84% performs frequent releases (vs. 76% for the sample)

In both cases, the levels of practice use are way higher than the sample average, and we cross-checked that the converse was not true to validate the significance of the findings.

Regarding the count of practices concurrently used by projects, only 1 project in the whole sample is using all the 6 practices included in the study. The largest groups are using just 2 or 3 practices.

Table 8. Count of practices

Count of practices	Count of projects	% of projects
0	3	7%
1	4	10%
2	12	29%
3	11	26%
4	6	14%
5	5	12%
6	1	2%

By doing a drill down considering the organizational agile experience we see that the count of practices tends to increase with organizational experience.

Table 9. Count of practices

Count of practices	Count of Projects	$x < 1$	$1 < x < 3$	$3 < x < 10$	$10 < x$
0	3	100%	0%	0%	0%
1	3	67%	33%	0%	0%
2	11	18%	55%	27%	0%
3	13	38%	46%	15%	0%
4	7	14%	14%	57%	14%
5	4	0%	50%	25%	25%
6	1	0%	0%	100%	0%

Finally we analyzed practices by demographic information segment (project duration, team size and industry type) but found no relevant patterns.

5. Conclusions and Future work

Technical practices have significantly lesser usage levels than organizational ones. This might point to interesting aspects of practice complexity and usability, affecting the learning and adoption processes. Future work might pursue the following questions: What are the root causes of the differences between technical and organizational practices? Do these observations hold for other practices like pair programming? Are there synergistic practice groups that thrive together, for example iterations and Frequent Releases? Would it help to use a Usability framework to analyze these patterns?

The validation questions helped us to detect cases of incorrect use of some practices, for example automated test that are out-of-date or continuous integration not so continuous. This situation of “incorrect use” is more frequent in technical practices than in organizational practices. This situation might deserve a deeper qualitative study of the practices and how practitioners are actually using them.

Finally, the fact that only 1 project (2%) is reported as using all the selected practices, while only 4 projects use all practices except TDD leads to the question: why these practices that are so popular in the agile bibliography and among agile gurus have this low adoption rate among practitioners even inside the agile community? This is an opportunity for further research on the subject, and work on process and practice usability might provide meaningful insights.

6. References

- [1] Beck, K. et al, "Manifesto for Agile Software Development." [Online]. Available: <http://agilemanifesto.org/>. [Accessed: 26-Aug-2016].
- [2] West, D., and Grant, T., "Agile Development: Mainstream Adoption Has Changed Agility," Forrester, 2010.
- [3] CA Technologies, "The Impact of Agile. Quantified," CA Technologies, 2015.
- [4] ACM/IEEE Joint Task Force on Computing Curricula, "Computer Engineering Curricula 2016, Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering," Association for Computing Machinery/IEEE Computer Society, Interim Curriculum Report, 2016.
- [5] Meyer, B., *Agile!: The Good, the Hype and the Ugly*, 2014 edition. New York: Springer, 2014.
- [6] Paulk, M. C. "On Empirical Research into Scrum Adoption," Institute for Software Research, Carnegie Mellon University, 2011.
- [7] Dingsøyr, T., Nerur, S., Balijepally, V. G., and Moe, N.B., "A Decade of Agile Methodologies: Towards Explaining Agile Software Development." *Journal of Systems and Software* 85, no. 6 (June 2012): 1213–1221.
- [8] Beck, K., and Andres, C., *Extreme Programming Explained: Embrace Change*, 2nd edition. Boston, MA: Addison-Wesley, 2004.
- [9] Fowler, M. (2009, January 29). FlaccidScrum. Available: <http://martinfowler.com/bliki/FlaccidScrum.html> [Accessed: 26-Aug-2016].
- [10] Vodde, B. (n.d.). History of Nokia Test - Bas Vodde. Retrieved August 15, 2016, from <http://blog.odde.com/basvodde/2011/02/history-of-nokia-test.html>
- [11] Robinson, H. and Sharp, H., "The Social Side of Technical Practices." In *Extreme Programming and Agile Processes in Software Engineering*. Sheffield, UK: Springer International Publishing
- [12] Version One, "State of Agile Development Survey," Version One, 2016.
- [13] Scrum Alliance, "The 2015 State of Scrum Report," Scrum Alliance, 2015.
- [14] Paulk, M. C., "Extreme Programming from a CMM Perspective," *IEEE Software*, vol. Nov/Dic 2001.
- [15] Cockburn, A., *Crystal Clear, A Human-Powered Methodology for Small Teams*. Addison-Wesley, 2005.
- [16] Chow, T., and Cao, D.B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961–971. <http://doi.org/10.1016/j.jss.2007.08.020>
- [17] Fowler, M., "Continuous Integration", Available: <http://www.martinfowler.com/articles/continuousIntegration.html>. [Accessed: 8-Aug-2016].
- [18] Cockburn, A., *Agile Software Development: The Cooperative Game*, 2nd edition, Addison-Wesley Professional, 2006.
- [19] Rodríguez, P., Adopción de metodologías ágiles: un estudio comparativo entre España y Europa. *Revista Española de Innovación, Calidad E Ingeniería Del Software*, 6(4), 2010.
- [20] Melo, O., Santos, C.V., and Katayama, E., "The Evolution of Agile Software Development in Brazil." *Journal of the Brazilian Computer Society* 19, no. November 2013 (n.d.): 523–552.
- [21] Ambler, S., "Agile Practices Survey Results: July 2009", Available: <http://www.ambysoft.com/surveys/practices2009.html>. [Accessed: 26-Aug-2016].
- [22] Dybå, T. and Dingsøyr, T., Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9–10), 833–859, 2008 <http://doi.org/10.1016/j.infsof.2008.01.006>
- [23] Diebold, P., Ostberg, J. P., Wagner, S. and Zendler, U. "What Do Practitioners Vary in Using Scrum?," in *Agile Processes in Software Engineering and Extreme Programming: 16th International Conference, XP 2015, Helsinki, Finland, May 25-29, 2015, Proceedings*, C. Lassenius, T. Dingsøyr, and M. Paasivaara, Eds. Cham: Springer International Publishing, 2015, pp. 40–51.
- [24] Kurapati, N., Manyam, V. S. C., and Petersen, K. (2012). Agile Software Development Practice Adoption Survey. En C. Wohlin (Ed.), *Agile Processes in Software Engineering and Extreme Programming* (pp. 16-30). Springer Berlin Heidelberg. Recuperado a partir de http://link.springer.com/chapter/10.1007/978-3-642-30350-0_2
- [25] "Agile Open Camp 2016." [Online]. Available: <http://www.agileopencamp.com.ar/>. [Accessed: 27-Aug-2016].
- [26] "Ágiles de Argentina, Sociedad Argentina de Informática." [Online]. Available: <http://www.sadio.org.ar/divisiones/agiles-de-argentina/>. [Accessed: 27-Aug-2016].