Using a Statistical Method to Compare Agile and Waterfall Processes Performance

Alexssander A. Siqueira(✉), Sheila Reinehr, and Andreia Malucelli

Pontifical Catholic University of Parana, Curitiba, PR, Brazil
alexssander_as@hotmail.com, sheila.reinehr@pucpr.br,
malu@ppgia.pucpr.br

Abstract. The agile principles and methods have become an important contribution in the Software Engineering area. Several works reported successful cases of agile adoption in the software industry and academic environment. Organizations can adopt the agile methods into their software development process using incremental or big bang approaches. The incremental approach allows the team members and stakeholders to assimilate the agile principles and techniques progressively. However, the use of this approach can omit supporting agile practices. In this case, process specialists usually recommend the big bang approach that allows teams to have a full agile practices experience. The current work contributes reporting an alternative approach based on the parallel development of projects using plan-drive and agile based processes. In this case, two development teams using different processes developed each project simultaneously. A set of ten real small projects were successfully delivered using this proposed approach and their respective results analysis, in terms of spent hours and defects, was performed using the Wilcoxon signed-rank test that consists in a nonparametric statistical method.

Keywords: Agile software development · Process improvement · Wilcoxon signed-rank test

1 Introduction

The agile methods and good-practices have been employed by several organizations to support their software development processes. In the software industry, there is a significant pressure to continuously improve business processes in order to stay competitive. The companies' productivity is directly impacted by their software development process's effectiveness [1].

In practice, few organizations are technically able to adopt agile development approaches immediately and adopt them successfully over a short period of time. A complete transition often takes a few years [2]. As the continuous agile methods adoption, there are emerging patterns of success and failure. Organizations can adopt the agile methods using an Incremental [3] approach that allows the team members and stakeholders to assimilate the agile principles and techniques progressively. In this case, the software development organization can resume their agile transition. These

J. Stolfa et al. (Eds.): EuroSPI 2017, CCIS 748, pp. 523–532, 2017.
DOI: 10.1007/978-3-319-64218-5_43

organizations would have conventional processes with the use of some agile techniques, not achieving a complete agile development process with the main agile practices and principles adoption.

The Big Bang [4] approach is a wholesale agile process adoption that is recommended by process specialists due to the team's need to experience the synergy of agile practices and the danger of omitting supporting practices when using the incremental approach. An agile process adoption rollout is a progressive and evolutionary flow that requires certain organization investment, especially in cultural changes to allow the agile principles and practices acceptance [5]. However, this one-way approach can cause several impacts when not successfully implemented.

A previous experience report [6] described a parallel development approach that employed a spare team of developers to initiate the agile transition in a telecommunication services company. That experience is a compromise between the Incremental and the Big Bang approaches, once the organization and development teams received the necessary training and experienced the agile techniques in real projects. However, the results analysis was not performed using any consistent method.

The current paper describes a second experience report based on parallel development that was performed in the same organization. This time, a set of ten projects were developed and the results analysis was performed with the Wilcoxon signed-rank test.

The Wilcoxon signed-rank test is a nonparametric statistical method whose application is suitable to perform data analysis from two-paired groups with different conditions [7]. The "W" test allows researchers to calculate and analyze the difference between the paired samples [8]. This nonparametric test approach was previously employed by [9] to perform software development effort estimation when encountering missing historical data sets issues. In this research, as there were a set of data measures from the same set of projects, the research group employed the Wilcoxon test to evaluate the action-research results after the projects development and their respective processes performance.

2 Research Method

The reported experience took place within a telecommunication company in Brazil. A research group was composed by an external researcher and two employees (company's internal process specialists). In this case, the suitable research method was the Action-Research (see Fig. 1). This method is an interactive approach that requires intensive researcher involvement and is composed by three types of steps [10]:

- **Pre-step:** research's context and purpose understating and definition.
- **Six main steps:** based on the context data information to elaborate an action plan that will be implemented and evaluated.
- **Meta-step:** monitoring the execution of the six main steps and identify the need for additional cycle planning and execution.

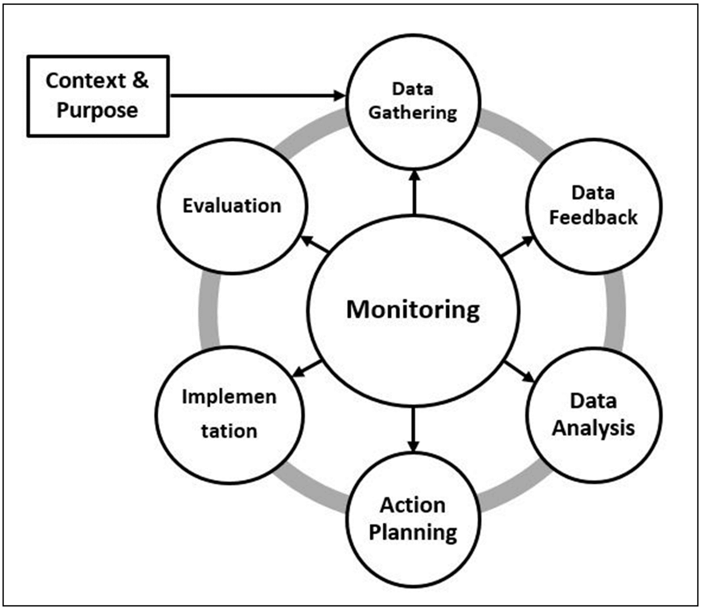

Fig. 1. Action-research method development cycle [10].

3 Research Development

A complete action-research development cycle was successful performed and described in this section.

3.1 Context and Purpose

The company had adopted in the past years many plan-driven development approaches such as the waterfall, spiral and a RUP based process. The main concerns were the difficulties to accept customer's change requests. A high number of software defects were found in the integration test and the projects costs were impacted by the unnecessary rework generated during the development phase. The main company's software development process (that it is the company's traditional process reference) was a waterfall-based process that has 5 simple phases: Business Analysis, Solution Design, Development, Integration Test and User Acceptance Test.

The development process was started with the internal customer writing the Initial Business Requirement document that contained the project's general scope that is described in functionalities and business constraints that must be implemented. Then, the development processes were performed as following:

- **Business Analysis:** A business analyst reviews the Initial Business Requirement document to describe in detail the new requirements in the Final Business Requirement document and identify any additional business constraint that must respected,

other functionalities, and projects that can be impacted by the change request. To complete the Final Business Requirement document, the business analyst informs the test cases that will be executed during the Test phase.
- **Solution Design:** A system architect reviews the Final Business Requirement document to describe the necessary changes in the involved components and databases. The architect output is the Solution Proposal document that will provide guidance to the developers during the Development phase.
- **Development:** The developers will execute the necessary changes in the identified components and databases using the Solution Proposal document as reference. The updated source code is uploaded into a specific source code repository to further deployment in the test environment.
- **Integration Test:** The project's source code is merged with other projects' source codes and deployed in the test environment. The test cases are executed as was previously informed in the Final Business Requirement document.
- **User Acceptance:** The internal customer representative checks if the delivered solution was developed as requested in the Initial Business Requirement document. After obtaining the users acceptance, the solution is deployed in the production environment for general availability.

The waterfall based process and its issues were discussed in the next action-research steps.

3.2 Data Gathering, Feedback and Analysis

The waterfall based process was presenting some issues during the project development that impacted the projects development performance (extra effort to projects completion, delivery delays) and quality (a high number of defects found during the integration test). The main issues were:

- **Documentation:** The communication channel to share information among the team members was the project's documentation. As the communication was very formal the requirements and the technical solution details were extensible documented. Even with the available documentation, architects had problems to design the solution using the project's requirements. Developers had problems to understand and implementing the solutions described by the architects. The quality of the software produced was impacted by defects that are generated during the solution coding and implementation.
- **Change Requests:** Any minimal change or update in the project‘s scope, was impacting the project development. The previous process phases were re-executed to support the necessary changes in the documents and components source code. In this case, changes in the project scope used to impact the cost and the project delivery date was usually postponed.
- **User Acceptance:** The business representatives (internal customer) were deeply involved in the beginning of the project when the Initial Business Requirement document was received and reviewed by the business analyst. However, during the project development the business representatives were not involved.

The adoption of agile methods is an alternative to mitigate the previously mentioned issues. A new agile development process was a suitable solution to the described organization context.

3.3 Action Planning

Due to listed productivity and quality issues that the company's traditional software development processes have presented in the last years, the company was looking for an alternative software development process to replace the traditional ones.

In this case, the necessary actions to achieve a software development proposal employing agile as an approach to solve the mentioned plan-drive issues are:

- **Action 1:** Provide the necessary agile awareness and training sessions.
- **Action 2:** Define an agile software development process proposal.
- **Action 3:** Develop 10 selected projects using the plan-drive and agile based processes.

These actions development are described in the next action-research step.

3.4 Implementation

Initially, the research group arranged a set of awareness and training sessions (Action 1) to support the use and understanding of the most important agile methods previously mentioned by [11], such as Scrum, XP (eXtreme Programming), TDD (Test-Driven Development) and PP (Pair Programing).

Then, to define an agile based development process (Action 2), the research group arranged a workshop with the development team. The result is an agile process proposal based on the core XP practices [12], such as:

- **Planning Game:** the iterations planning and control must use Scrum. The stories (activities) progress are visible for the project team and the internal customer representative (product owner). The iteration duration is 2 weeks. It is necessary to arrange a planning meeting in the iterations' beginning, daily meetings, and a final meeting in the iterations' end to verify its completion and discuss about the main goals and issues found.
- **Metaphor:** the business requirements have to be described using a simple and direct language (customer friendly language).
- **Small Releases:** all iterations should produce a software artifact that may be part of the architectural solution or a functional feature (working software) that will tested by the Test Team (Integration Test).
- **Simple Design:** the solution proposal document must have the fewest possible classes and methods. Just the minimal information to clarify the problem solution and to define a set of technical stories that will be addressed to the developers in the planning game (Scrum meetings).
- **Tests:** the source code development must be started with the elaboration of a test case. The TDD agile method supports the developer's activities.

- **Pair Programming:** this practice is applied just when a story involves a critical component change or development. This practice allows knowledge and experience sharing between two developers when working with a complex solution.

The research group organized the projects development by two different teams. All team members were newcomers to this development department. All of them, have similar skills, technological expertise and academic background. They were trained in the organization tools and technologies. However, they did not have any previous contact with the organization's waterfall based software development process.

Before starting the projects development, the following assumptions were stated as rules:

- **Parallel Development:** the same project must be developed using the waterfall based life-cycle and the agile development process. The start date was the same for both initiatives.
- **Team Member Assignment:** as the projects must be implemented in parallel, the team members would be assigned for only one of the projects implementation. It was important to balance the number of traditional and agile projects that each team member is assigned.

All selected projects were successfully executed in three months. The research assumptions were respected and followed. The parameters were registered as planned and the comparative study started analyzing the effort (hours) expended to develop the projects.

3.5 Evaluation

The initial analysis was focused on the necessary effort (hours) to deliver the project using both software development processes (see Table 1). Observing the hour's average spent in the projects, it could be possible to affirm that the agile process is more expensive than the waterfall one.

However, it is necessary to check if this difference in effort is statistically significant. The employment of a nonparametric statistical method is suitable alternative as previously described in Sect. 1. The test regarding the processes performance considered the following hypotheses:

- **Hypotheses 1:** The agile process is more expensive than the waterfall based one.
- **Hypotheses 2:** The agile process performance is not significant different from the waterfall based one.

To use the Wilcoxon test to evaluate the processes regarding their effort in hours, the following steps were performed as indicated in Table 1:

- **Step 1:** Firstly, it was calculated the difference between each value of hours by project.
- **Step 2:** Then, ignoring sign of the difference, each value was ranked starting from the smaller (Project 09 = 12) value to the highest one (Project 05 = 160). In this case, there was a match with 3 different projects with the value of 16, it was necessary to

Table 1. Wilcoxon test for projects effort analysis.

Wilcoxon Signed-Ranking Test- Analyzing Projects Effort (Hours)							
Project	**Waterfall**	**Agile**	**Difference**	**Rank**	**Rank Sum**		
01	112	128	16	(2) 3	**Positive**	37	
02	600	680	80	8	**Negative**	18	
03	836	960	124	9		❸	
04	708	672	-36	7			
05	400	560	160	10	**Values**		
06	228	244	16	(3) 3	**W**	18	❹
07	488	456	-32	6	**N**	10	❺
08	272	288	16	(4) 3	**α < 0.05**	8	❻
09	240	252	12	1			
10	672	648	-24	5			
Average	**455,6**	**488,8**	❶	❷			

sum their respective ranking values (2 + 3 + 4) and divide by 3, obtaining the average value of 3 for their ranks.

- **Step 3:** Next, the sum of positive and negative sign ranks was calculated (Positive = 37 and Negative = 18).
- **Step 4:** As the negative score is the smaller one, it was selected as value of "W" (W = 18).
- **Step 5:** As there was not any difference value equal to zero, the "N" value is number of projects (N = 10).
- **Step 6:** Finally, consulting the Wilcoxon's table of critical values, considering the value of "N" the result value is 8 for the significance level of 0.05 ($\alpha \leq 0.05$).

By the Wilcoxon signed-rank test, a calculated "W" is significant if it is less than or equal to the critical value [8]. The obtained value of 18 is larger than 8, in this case the Hypotheses 2 must be assumed as true, because there was no significant difference between the effort spent in the waterfall and agile based processes.

Checking the processes' average number of defects, the waterfall process produced more defects during the integration test. The Wilcoxon test was employed to verify if this average difference is statistically significant. The test regarding the number of defects identified in the processes considered the following hypotheses:

- **Hypotheses 1:** The waterfall process produces more defects than the agile one.
- **Hypotheses 2:** The number of defects produced by the waterfall process is not significant different from the agile one.

Then, the following steps were performed as indicated in Table 2 to execute the Wilcoxon signed-rank test:

- **Step 1:** The difference was calculated between each project's hours value.
- **Step 2:** Then, ignoring sign of the difference, each value was ranked starting from the smaller (Project 05 = 1) value to the highest one (Project 08 = 16). In this case, there were two matches; the first match was between the projects with difference value of 8. The rank value for both was updated to 3.5. The second match occurred with the difference value of 14, obtaining the average value of 7.5 for both ranks.
- **Step 3:** Next, the sum of positive and negative sign ranks was calculated (Positive = 0 and Negative = 55).
- **Step 4:** As the positive score is the smaller one, it was selected as value of "W" (W = 0).
- **Step 5:** As there was not any difference value equal to zero, the "N" value is number of projects (N = 10).
- **Step 6:** Finally, consulting the Wilcoxon's table of critical values, considering the value of "N" the result value is 8 for the significance level of 0.05 ($\alpha \leq 0.05$).

Table 2. Wilcoxon test for projects number of defects analysis.

Wilcoxon Signed-Ranking Test - Analyzing Projects Defects							
Project	**Waterfall**	**Agile**	**Difference**	**Rank**	**Rank Sum**		
01	16	4	-12	6	**Positive**	0	
02	8	0	-8	(3) 3,5	**Negative**	55	
03	16	2	-14	(7) 7,5		❸	
04	19	5	-14	(8) 7,5			
05	8	3	-5	1	**Values**		
06	13	6	-7	2	**W**	0	❹
07	13	5	-8	(4) 3,5	**N**	10	❺
08	19	3	-16	10	**α < 0.05**	8	❻
09	12	3	-9	5			
10	19	4	-15	9			
Average	**14,3**	**3,5**	❶	❷			

The obtained value of 0 is smaller than 8, in this case the Hypotheses 1 can be assumed as true, because there was a significant difference between the number of software defects that were identified during the integration test.

3.6 Results Analysis

The agile development process consumed more hours to deliver the same set of projects, but it is not possible to affirm that the proposed agile process is more expensive than the waterfall process. In this case, it is necessary to consider testing the number of identified defects in each project, once this is a company issue reported during the action-research previous steps of Data Gathering, Feedback and Analysis (Sect. 3.2). As the defects average score for the waterfall process is larger than the proposed agile process, it is possible to affirm that the use of agile methods such as TDD and PP can improve the software quality as employed in the company new agile development process. However, the PP was employed when the teams had complex solutions to be developed. In fact, the PP supported the share knowledge among the team members due to its collaborative and interactive approach.

4 Conclusion

The experience reported in this work, compared the performance of two different software development processes, using a nonparametric statistical method that allowed the test and analysis of data provided by the different measures obtained by the same set of projects. Instead of performing a simple analysis on the obtained averages values, the test identified if there was a statistically significant difference between the measured values. In this case, the Wilcoxon signed-rank test become an important contribution to compare process's performance indicators.

This research was possible due to the use of an alternative parallel approach that consists of the development of selected set of projects in parallel using two different processes and separated teams of developers.

The use of real projects was possible once the new employees were not engaged in projects and they were available to receive the technical training. As a similar context of having spare development teams is very unusual to happen in a corporative environment, a similar experience could be developed in an academic environment using active methodologies such as Project-Based Learning or Problem-Based Learning approaches.

An important research limitation was the teams' size and projects' scope extension. Some small projects were performed in just one single sprint. In this cases, the team members were not fully assigned to these projects. They completed their activities working less than 40 h/week. This is not a recommended practice, however this was not an issue to the research development. Further work, should consider all projects with a minimum number of three sprints. Then, the research group will be able to register the teams' feedbacks and sprints retrospective properly.

References

1. Kukko, M., Helander N., Virtanen, P.: Knowledge management in renewing software development processes. In: Proceedings of the 41st Hawaii International Conference on System Sciences, Waikoloa, Big Island, pp. 1–9 (2008). doi:10.1109/HICSS.2008.229
2. Qumer, A., Henderson-Sellers, B.: A framework to support the evaluation, adoption and improvement of agile methods in practice. J. Syst. Softw. **81**, 1899–1999 (2008). doi:10.1016/j.jss.2007.12.806
3. Hodgetts, P.: The impact of refactoring the development process - experiences with the incremental adoption of agile practices. In: Proceedings of the Agile Development Conference, pp. 106–113. IEEE Computer Society, Salt Lake City (2004)
4. Mencke, R.: A product manager's guide to surviving the big bang approach to agile transitions. In: Proceedings of the 2008 Agile Conference, pp. 407–412. IEEE Computer Society, Toronto (2008). doi:10.1109/Agile.2008.65
5. Roche, G., Vaquez-McCall, B.: The amazing team race - a team based agile adoption. In: Proceedings of the 2009 Agile Conference, pp. 141–146. IEEE Press, Chicago (2009). doi:10.1109/AGILE.2009.67
6. Siqueira, A., Reinehr, S., Malucelli, A.: Agile adoption - developing projects in parallel with agile and traditional life-cycles. In: Proceedings of the 24th European System, Software and Service Process Improvement and Innovation Conference, pp. 54–66. Publizon, Luxembourg City (2014)
7. Corder, G.W., Foreman, D.I.: Nonparametric Statistics: A Step-by-Step Approach, 2nd edn. Wiley, Somerset (2014)
8. Graham Hole Research Skills: Wilcoxon hand out. http://users.sussex.ac.uk/~grahamh/RM1web/WilcoxonHandoout2011.pdf. Accessed 25 May 2010
9. Idri, A., Abname, I., Abran, A.: Missing data techniques in analogy-based software development effort estimation. J. Syst. Softw. **117**, 595–611 (2016). doi:10.1016/j.jss.2016.04.058
10. Coughlan, P., Coghlan, D.: Action research for operations management. Int. J. Oper. Prod. Manag. **22**, 220–240 (2002). doi:10.1108/01443570210417515
11. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. J. Syst. Softw. **85**, 1213–1221 (2012). doi:10.1016/j.jss.2012.02.033
12. Beck, K.: Embracing change with extreme programming. Computer **32**, 70–77 (1999)