Strategies for agile software development based on technical and environmental complexity factors

Fernando Pinciroli[1]

[1]Instituto de Investigaciones, Universidad Nacional de San Juan,
San Juan, Argentina
fernando.pinciroli@gmail.com

Abstract. Project management strategies require a perfect understanding of the problem to be faced. For this, the Cynefin framework is a good starting point, offering a way to classify reality according to its complexity. However, there are other aspects that impact and increase the complexity of projects. One of them is the number of triple constraint elements that are fixed; the other, corresponds to project characteristics such as size, criticality, time constraint, etc. In this work we offer a strategy to classify the best tools, techniques and approaches for project management depending on technical and environmental complexity factors.

Keywords: project management, agile methodologies, Cynefin framework.

1 Introduction

The management of software development projects is complex. It requires dealing with numerous unforeseen events that constantly arise along the way and that go against the expectations that had been established at the beginning. A good project leader is not so much the one who carries out what is planned, but rather the one who is able to deal with the inconveniences that arise and, in the end, achieve a decent outcome [1].

On the other hand, we are used to cling to the tools that gave us the best results, although many times we continue to do so even when the context has changed.

Also, software development is complex, both because software is inherently so and because people, who are an essential and intensive part of that development, and our relationships are even more so.

Finally, for some decades we have extrapolated many proven techniques, tools and approaches for project management from other fields to computer science [2]. Surely many of them were of value, but others did not give the expected results.

The appearance of the Manifesto for Agile Software Development [3], in 2001, radically changed the way of seeing project management and was like a breath of fresh air, but there are still several difficulties that sometimes arises:

- Agile tools and techniques applied in projects, but without an agile approach.

- Reality is not analyzed to determine the best tools to use.
- Agility implies adaptation, but there are not too many cases in which tools and techniques are adapted or combined depending on the problem to be faced.
- Many people are trained in a single agile method and use it systematically.

2 Mapping tools, techniques, and approaches

Snowden created the Cynefin framework [4], which describes a way to categorize reality based on its complexity. The model consists of five contexts that are defined by the nature of the cause-effect relationship present in them. The contexts are called “clear”, “complicated”, “complex”, “chaotic”, and “aporetic”. The first four can be identified by the characteristics that are observed, but the fifth corresponds to the situation in which it was not possible to characterize the context for whatever reason.

It is also widely known the triple restriction on projects: scope, time and cost [2] [5]. In the projects, some of these three restrictions are fixed, forcing the rest to be variables, in order to be able to manage the project. This characterizes the type of management that can be carried out, whether predictive or empirical.

Based on Cynefin and the characterization of projects regarding their fixed and variable restrictions, we proceeded to locate different project management techniques, tools, and approaches in the four contexts of the former. The objective of this mapping is to guide the selection of project management strategies.

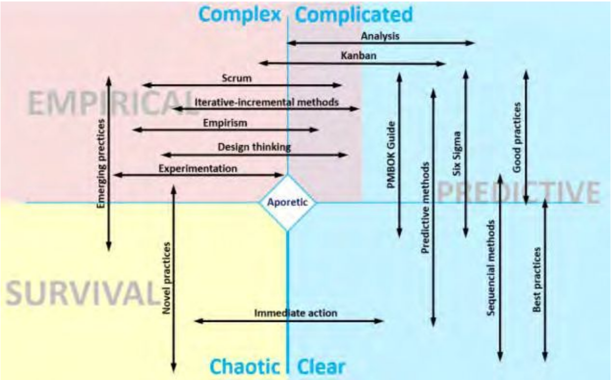

Fig. 1. Tools, techniques, and approaches for project management in Cynefin framework.

In this line of thought, Kenneth Rubin makes a first approach to locate the techniques in each Cynefin’s context [6]. It places the waterfall style, in the clear context; an analytical method, like Six Sigma, in the complicated context; and an iterative and incremental approach, like Scrum, in the complex one. Paulus et al. [7] also classify sequential development, based on a plan, as an appropriate approach for a clear context. Pelrine [8] understands that a project is not complex or complicated in itself, but rather its constituent parts that can be classified independently in the four Cynefin's contexts.

3 Technical and environmental complexity factors

There are other factors that influence the complexity of the projects and that would move them from context to one of higher complexity: time constraints, knowledge of the problem domain, aspects of interest to stakeholders, size of the project (in people, requirements, duration, etc.), criticality, global software development, culture and maturity of the organization, etc. [9].

Alistair Cockburn establishes a principle that the size of the project management methodology to be used will be determined by the presence of these factors [10] [11]. Along with Jim Highsmith, Alistair states that agile development is more difficult with larger teams [12], referring to projects involving hundreds of people, indicating that extra precautions must be taken due to the increasing complexity of the project.

Thus, an agile development complemented with the necessary methodological elements to manage the complexity added by the size of the problem, could be an adequate solution, since a waterfall model would not be adequate in these cases [13]. Serrador and Pinto also found evidence that larger projects require a combination of agile management with a higher degree of advance planning [14].

Alistair [10] mentions that the criticality of the problem demands a greater methodology and McConnell confirms it through another way [15], by presenting the productivity that can be obtained according to the criticality of the problem. Other authors recognize the contribution made by approaches based on a plan to projects that require important levels of security, reliability and protection [7] [14].

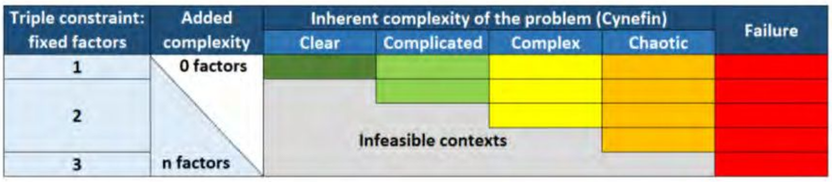

Fig. 2. Possible scenarios depending on the project's complexity.

We have designed a problem classification scheme in which possible scenarios are presented in the presence these complexity factors (Fig. 2). We must add to the Cynefin's contexts the complexity added by the number of fixed elements of the triple restriction and the number of factors that impact a project complexity. Thus, we present the contexts that are possible and those that are not, as the technical and environmental complexities of a project grow.

5 Conclusions and future work

The nature of a project should be characterized in order to select the best strategies to manage it. This characterization could be done based on the triple restriction and the added complexity factors present in the challenge to be faced. We have proposed a strategy to locate every project in the Cynefin framework depending on its complexity and then choosing the most appropriate techniques, tools, and management approaches. This selection must continue throughout the life of the project because reality changes and the project can move from one Cynefin context to another, with the consequent need to employ new techniques, tools, and management approaches to keep the project under control.

As future work, it remains to obtain more evidence from the literature and describe with more detail the impact of each factor of technical and environmental complexity.

References

1. McConnell, S. Software project survival guide. Redmond, Microsoft Press, 1998.
2. Project Management Institute. A guide to the project management body of knowledge. 6th ed. Project Management Institute, 2017.
3. "Manifesto for Agile Software Development". https://agilemanifesto.org/
4. Snowden, D.J. and Boone, M.E. "A Leader's Framework for Decision Making". Harvard Business Review, p. 10, 2007.
5. Atkinson, R. "Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria". International Journal of Project Management, vol. 17, no. 6, pp. 337–342, 1999.
6. Rubin, K.S. Essential Scrum: a practical guide to the most popular agile process. Upper Saddle River, Addison-Wesley, 2012.
7. Paulus, S., Mohammadi, N.G. and Weyer, T. "Trustworthy Software Development". In: Communications and Multimedia Security, vol. 8099, De Decker, B. et al. eds. Berlin, Springer, pp. 233–247, 2013.
8. Pelrine, J. "On Understanding Software Agility – A Social Complexity Point Of View", vol. 13, p. 13, 2011.
9. Špundak, M. "Mixed Agile/Traditional Project Management Methodology – Reality or Illusion?". Procedia - Social and Behavioral Sciences, vol. 119, pp. 939–948, 2014.
10. Cockburn, A. "Selecting a project's methodology" IEEE Software, vol. 17, no. 4, pp. 64–71, 2000.
11. Cockburn, A. "People and Methodologies in Software Development," PhD Thesis, University of Oslo, 2003.
12. Cockburn A. and Highsmith, J. "Agile software development, the people factor". Computer, vol. 34, no. 11, pp. 131–133, 2001.
13. Petersen, K., Wohlin, C. and Baca, D. "The Waterfall Model in Large-Scale Development" In: Product-Focused Software Process Improvement, vol. 32, Bomarius, F. et al. eds. Berlin, Springer, pp. 386–400, 2009.
14. Serrador P. and Pinto, J.K. "Does Agile work? – A quantitative analysis of agile project success". International Journal of Project Management, vol. 33, no. 5, pp. 1040–1051, 2015.
15. McConnell, S. Software estimation: demystifying the black art. Redmond, Microsoft Press, 2006.