Management ambidexterity: A clue for maturing in agile software development

Rafaela Mantovani Fontana[1,2], Victor Meyer Jr.[1], Sheila Reinehr[1], and Andreia Malucelli[1]

[1] Pontifical Catholic University of Paraná (PUCPR)
R. Imaculada Conceição, 1155, 80215-901 Curitiba, PR, Brazil
[2] Federal University of Paraná (UFPR)
R. Dr. Alcides Vieira Arcoverde, 1225, 81520-260 Curitiba, PR, Brazil
rafaela.fontana@ufpr.br,victormeyerjr@gmail.com,sheila.reinehr@pucpr.br,
malu@ppgia.pucpr.br

Abstract. Organizational ambidexterity is the ability to be aligned and efficient in combining current resources and demands (exploitation) as well as adaptive and innovative due to changing conditions and demands (exploration). Maturity in software development is defined over exploitation – through processes definition and control. We argue in this study that mature agile software development is also exploratory – adaptive and innovative. Thus, our objective is to verify how ambidexterity occurs in mature agile software development. The research approach is a single case study with analysis of qualitative data. Our findings show how a mature team is managed by ambidextrous strategies.

Key words: ambidexterity, maturity, agile software development

1 Introduction

Organizational ambidexterity represents an essential ability in organizations that wish to prosper in high-velocity environments [1], and requires efficient management of current demands (exploitation) with simultaneous adaptation to a changing environment (exploration) [2], [3].

Our previous work shows evidence that an exploitative focus on process definition and control is not the means to mature in agile software development [4];[5]. However, solely focusing on exploratory practices may lead to "too many undeveloped new ideas and too little distinctive competence" [3, p. 71]). This is the reason we argue mature agile software development management presents ambidextrous abilities.

Our research question is therefore, the following: *How does ambidexterity occur in mature agile software development?* The research approach is a single case study with qualitative data collection and analysis. This study contributes to the research by integrating organizational theory and agile software development. For practitioners, our findings offer suggestions for strategic management actions that support maturing in agile software development.

This paper is organized as follows. Section 2 briefly presents the organizational ambidexterity concept. Section 3 describes the research approach. Section 4 presents the results and, finally, Section 5 discusses the findings and concludes the paper.

2 Organizational ambidexterity

The ability to simultaneously pursue exploitation and exploration [3, p. 71] has currently been termed "organizational ambidexterity" [6]. Ambidextrous organizations are successful because they are aligned and efficient in their management of current demands while simultaneously adaptive to environmental change [2]. Alignment is characterized by the ability to maintain individuals working towards the same goals and adaptability refers to the ability to quickly adjust activities as a response to changes in the environment [7].

The challenge in describing ambidexterity is that organizations adopt idiosyncratic implementation strategies to become ambidextrous [7]. Gibson and Birkinshaw [7], for example, have identified contextually ambidextrous behavior by analyzing the ability of a business unit to be aligned and adaptable simultaneously. Tiwana [8] has identified ambidexterity in information flows among individuals, which can be "strong ties" or "bridging ties". Bridging ties lead to diversity of accessible knowledge, while strong ties lead to knowledge integration. Tiwana has also characterized ambidexterity as a combination of formal and informal (clan) controls [9].

Organizational ambidexterity is currently evolving as a research paradigm in organizational theory [2], is recognized as an adequate approach to the uncertainty of the software product market [10], and is verified in agile software development teams [11];[12]. It is relevant, then, to investigate how ambidexterity is accomplished in mature agile software development.

3 Research design

The research question that drives this study is *"How does ambidexterity occur in mature agile software development?"*. We chose the single case study as research approach because it is an effective method to "understand the dynamics present within single settings" [14, p. 534].

The unit of analysis was an agile software development firm. We selected a Brazilian agile software development company, that developed and customizes a product which is currently used by a single big customer. This case is specifically interesting to study ambidexterity because company's manager has more than ten years of experience in managing agile teams (is one of the introducers of agile methods in Brazil). The firm has forty-five employees.

Our data collection and analysis were conducted in October/November, 2014 and guided by the proposition: *Management simultaneously combine formal and clan mechanisms in mature agile software development.*

To verify this proposition we applied a qualitative approach through interviews with three team members (a team leader, a functional analyst – named "feature owner" –, and a developer). The protocol for data collection was based on the study presented by Tiwana [9]. After the interviews had been conducted, they were transcribed and analyzed using thematic networks analysis [13]. With the objective to have descriptive statistics of the perception of maturity and ambidexterity, we also collected quantitative data through a questionnaire. It measured ambidexterity perception, based on [7] and [8], and maturity perception, based on [4].

The threats to validity in this study are internal and external. Internal validity is threatened by the interpretation of interviewees responses. To minimize this threat, we presented the results for evaluation by the manager of the firm, who did not participate in the interviews. The external validity is threatened by the findings from the single case study. In this case, we did not pursue statistical generalization, but analytical generalization instead [14].

4 Results

The measurement of maturity and ambidexterity showed that the majority of the respondents have a perception that the company is mature and that the management is ambidextrous. Twenty-one employees (46% of the total) responded the questionnaire. On average, 84% of the respondents agree with the agile maturity evidences in their work processes; 73% agree that management system encourages people to challenge outmoded traditions, is flexible and responds quickly to changes (exploration); and 95% agree that management works coherently to support the overall objectives of the organization (exploitation).

The analysis of the interviews resulted in a number of evidences that represent observations of firm ambidextrous behavior. In summary, we identified that ambidexterity is enabled by two main management strategies 1) exploiting enough to add value to the customer and the team; and 2) processes automation and results visibility. To accomplish the first initiative, we observed that management chases definition and execution (exploitation) in such a way that added value is realized by the customer (outcomes and estimates) and the team (definition of roles and estimates), but space is left for variety and discovery in estimates, in the relationship with the customer and within team. For the second initiative, we observed that management defined a few indicators concerning the process, and discussions and solutions are left to emerge from the informal interactions among team members.

Next subsections present the evidences for these management strategies.

4.1 Exploiting enough to add value for the customer and for the team

Deliberate loose estimates. Estimates at this company are based on past data but are also deliberately loose: "... for example, nobody says that we have

four or five hours to finish a task. We classify tasks as small, medium or big. There are some statistics that state the range of hours for small tasks, medium tasks and big tasks."(Developer, discussing loose estimates). Because deliveries are fixed every week, there are always features being delivered to customer. When required, long-term planning is made through milestones, leaving space for flexibility in the activities for each milestone. *We see here planning deriving from the combination of exploitation, through long-term milestones planning and estimates based on past data; and exploration from loose delivery dates, and considering individuals' experience on tasks classification.*

Customer care through people and perceived outcomes. The customer of this company is not significantly involved with the team, but the customer feels supported by a focal point, the feature owner: "We work a lot with the customer to define the requirements he needs." (Feature owner, about helping customer). The customer also receives weekly deliveries, which are validated and formally accepted. *In this relationship we see exploration evidenced by the customer sensing that the team assists in the processes, and exploitation from a defined process for deliveries, which provides the customer with a sense of outcome as a result of the team's work.*

Team atmosphere is friendly and established. The atmosphere among team members is pleasant. The team members do not typically perform overtime and, if they must, the overtime is planned and communicated. Whenever situations of concern occur in the team, the roles of the leaders able to assist in problem solving or decision making is clear. The teams under each feature owner are small, which facilitates informal communication and problem solving. *We see, then, exploration enabled through the close relationship between individuals and exploitation facilitated by well-defined problem-solving roles.*

4.2 Processes automation and status visibility

Task work out with checkpoints and flexibility. This firm leads task accomplishment with a high level of automation. Repetitive and time-consuming development tasks are automated in an internally developed toolkit. Functional tests and code integration are also automated. The status of the repository, the tests suite, and other status in the environment are visible to the team through a number of dashboards on monitors distributed around the rooms. At the same time, tasks completion is flexible. Whenever problems occur with tasks execution, discussions for the solution are collective, and the decision to delay the task is also collective. *This extensive automation and status visibility aligns developers' work with processes, leading to exploitation. The discussions concerning the tasks and the status that is visible to all facilitate exploration in task work out.*

Process enduring through people and automation. The team members were highly involved in the definition of the process when the agile method was adopted. With the current established, stable process, people are encouraged to automate anything they consider may ease their work. The process is simple with few manual steps (coding, reviewing and testing). *Thus, we consider that*

automation retain exploitation in the process, while discussions and freedom to change guarantee exploration.

Communication arena based on status visibility and conversation. All data concerning what needs to be done, what is being done, environment and tests status is shown in real time for everyone. This high visibility and the red indicators when something malfunctions stimulates people to discuss and engage in problem solving. *We find, then, that exploitation is derived from status visibility and exploration from intense, informal, and close communication.*

5 Discussion and conclusions

This study aimed to identify how ambidexterity occurs in mature agile software development. We have qualitatively shown, based on a single case study, the management strategies used to allow ambidexterity in agile software development.

The benefits of combining exploitation and exploration have already been identified in agile software development. Boehm [15] emphasized the importance of combining plan-driven and agile approaches. Baskerville et al. [16] previewed the future of agility as the search for the dual objective of agility and alignment. These benefits and mechanisms have been empirically verified in globally distributed software projects [17].

Ramesh et al. [18] identified a number of balanced practices where exploitation and exploration (e.g., formal structures with flexibility, trust but verify etc) were simultaneously applied in agile software development. However, the research focus was on distributed development, which has an explicit need for discipline [18]. Our study adds to their findings showing that even a collocated team, when maturing, develops ambidextrous abilities – through deliberate management strategies – to maintain team work aligned and at the same time flexible and responding to changes.

This study also addresses a concern of the companies in the adoption of agile methods, which is the lack of managerial control [19]. We assume what companies fear loss of control based on processes exploitation, which leads to the search for stability. We demonstrated that management in agile methods remains exploitative, but is also exploratory.

Our findings are limited by the context of the studied firm, but may serve as a reference for practitioners that wish to improve management practices, and as a subject for further replication and validation for researchers that are interested in applying organizational theories to agile software development.

References

1. Tushman, Michael L.; O'Reilly III, Charles A.: Ambidextrous organizations: Managing evolutionary and revolutionary change. California Manag. Review, 38, 8-30 (1996)

2. Raish, Sebastian; Birkinshaw, Julian: Organizational Ambidexterity: Antecedents, Outcomes and Moderators. J. Manag., 34, 375-409 (2008), doi:10.1177/0149206308316058
3. March, James G.: Exploration and Exploitation in Organizational Learning. Organiz. Sci., 2 (1991)
4. Fontana, R. M.; Fontana, I. M.; Garbuio, P. A. R.; Reinehr, S.; Malucelli, A.: People over processes: how should agile software development be defined? J. Syst. Softw., 97, 140-155 (2014), doi:10.1016/j.jss.2014.07.030
5. Fontana, Rafaela M.; Reinehr, Sheila; Malucelli, Andreia: Maturing in Agile: What Is It About? In: Proceedings of the 15th International Conference, XP 2014, Rome, Italy, May 26-30, 94-109 (2014), doi:10.1007/978-3-319-06862-6_7
6. Tuner, Neil; Swart, Juani; Maylor, Harvey: Mechanisms for Managing Ambidexterity: A Review and Research Agenda. Int. J. Manag. Reviews., 15, 317-332 (2013), doi:10.1111/j.1468-2370.2012.00343.x
7. Gibson, Cristina; Birkinshaw, Julian: The antecedents, consequences, and mediating role of organizational ambidexterity. Academy Manag. J., 47, 209-226 (2004), doi:10.2307/20159573
8. Tiwana, Amrit: Do bridging ties complement strong ties? An empirical examination of alliance ambidexterity. Strat. Mgmt. J., 29, 251-272 (2008), doi:10.1002/smj.666
9. Tiwana, Amrit: Systems Development Ambidexterity: Explaining the Complementary and Substitutive Roles of Formal and Informal Controls. J. Manag. Inf. Syst., 27, 87-126 (2010), doi:10.2753/MIS0742-1222270203
10. Harris, Michael L.; Collins, Rosann Webb; Hevner, Alan R.: Control of Flexible Software Development Under Uncertainty. Inf. Syst. Research, 20, 400-419 (2009), doi:10.1287/isre.1090.0240
11. Vidgen, R.; Wang, X.: Coevolving Systems and the Organization of Agile Software Development. Inform Syst Res., 20, 355376 (2009), doi:10.1287/isre.1090.0237
12. Boehm, B.; Turner, R.: Balancing Agility and Discipline: Evaluating and Integrating Agile and Plan-Driven Methods. In: Proceedings of the 26th International Conference on Software Engineering, 23-28, 718-729 (2004), doi:10.1109/ICSE.2004.1317503
13. Attride-Sterling, Jennifer: Thematic networks: an analytic tool for qualitative research. Qualitative Research, 1, 385-405 (2001), doi:10.1177/146879410100100307
14. Eisenhardt, K.: Building Theories from Case Study Research. Academy Manag. Review, 14, 532-550 (1989) Available at http://www.jstor.org/stable/258557
15. Boehm, Barry: Get ready for agile methods, with care. IEEE Comp, 35, 64-69 (2002), doi:10.1109/2.976920
16. Baskerville, Richard; Pries-Heje, Jan; Madsen, Sabine: Post-agility: What follows a decade of agility? Inf. Soft. Tech., 53, 543-555 (2011), doi:10.1016/j.infsof.2010.10.010
17. Lee, Gwanhoo; Delone, William; Espinosa, J. Alberto: Ambidextrous Coping Strategies in Globally Distributed Software Development Projects: Strategies for enhancing flexibility and rigor. Comm. ACM, 49, 35-40 (2006), doi: 10.1145/1164394.1164417
18. Ramesh, Balasubramaniam; Mohan, Kannan; Cao, Lan: Ambidexterity in Agile Distributed Development: An Empirical Investigation. Inf. Syst. Research, 23, 323-339 (2012), doi:10.1287/isre.1110.0351
19. Melo, Claudia de O.; Cruzes, Daniela S.; Kon, Fábio; Conradi, Reidar: Interpretative case studies on agile team productivity and management. Inf. Soft. Tech, 55, 412-427 (2013), doi:10.1016/j.infsof.2012.09.004