Information and Software Technology 55 (2013) 412–427

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Interpretative case studies on agile team productivity and management

Claudia de O. Melo [a,*], Daniela S. Cruzes [b], Fabio Kon [a], Reidar Conradi [b]

[a] Department of Computer Science, University of São Paulo, São Paulo, Brazil
[b] Department of Computer and Information Science, NTNU, Trondheim, Norway

ARTICLE INFO

Article history:
Received 1 April 2012
Received in revised form 5 September 2012
Accepted 10 September 2012
Available online 21 September 2012

Keywords:
Agile software development
Team productivity factors
Team management
Thematic analysis
Industrial case studies

ABSTRACT

Context: The management of software development productivity is a key issue in software organizations, where the major drivers are lower cost and shorter time-to-market. Agile methods, including Extreme Programming and Scrum, have evolved as "light" approaches that simplify the software development process, potentially leading to increased team productivity. However, little empirical research has examined which factors do have an impact on productivity and in what way, when using agile methods.

Objective: Our objective is to provide a better understanding of the factors and mediators that impact agile team productivity.

Method: We have conducted a multiple-case study for 6 months in three large Brazilian companies that have been using agile methods for over 2 years. We have focused on the main productivity factors perceived by team members through interviews, documentation from retrospectives, and non-participant observation.

Results: We developed a novel conceptual framework, using thematic analysis to understand the possible mechanisms behind such productivity factors. Agile team management was found to be the most influential factor in achieving agile team productivity. At the intra-team level, the main productivity factors were team design (structure and work allocation) and member turnover. At the inter-team level, the main productivity factors were how well teams could be effectively coordinated by proper interfaces and other dependencies and avoiding delays in providing promised software to dependent teams.

Conclusion: Teams should be aware of the influence and magnitude of turnover, which has been shown negative for agile team productivity. Team design choices remain an important factor impacting team productivity, even more pronounced on agile teams that rely on teamwork and people factors. The intra-team coordination processes must be adjusted to enable productive work by considering priorities and pace between teams. Finally, the revised conceptual framework for agile team productivity supports further tests through confirmatory studies.

1. Introduction

The management of software development productivity is a key issue in software organizations, where the major drivers are lower cost and shorter time-to-market [18,100]. To manage productivity effectively, it is important to identify the most relevant difficulties and develop strategies to cope with them. Agile methods, including Extreme Programming [10] and Scrum [89], have evolved as approaches to simplify the software development process, potentially leading to better productivity. They aim to shorten development time and handle the inevitable changes resulting from market dynamics [50,82].

Considerable research has been directed at identifying factors that have a significant impact on software development productivity [100,106]. In general, the studied productivity factors were related to product (specific characterization of software), personnel (team member capabilities, experience, and motivation), project (management aspects, resource constraints), or process issues (software methods and tools). Continuously evaluating productivity factors is important, as factors may change under new software engineering practices [81].

Although the industry has extensively adopted agile methods, little research has empirically examined the software development agility construct regarding its dimensions, determinants, and effects on software development productivity and performance [54,95,8,35,92]. Understanding the factors that affect productivity could help determine where to concentrate management efforts (and related financial resources) from a practical standpoint and

* Corresponding author. Tel.: +55 11 97027-3090.
E-mail addresses: claudia@ime.usp.br (C. de O. Melo), dcruzes@idi.ntnu.no (D. S. Cruzes), fabio.kon@ime.usp.br (F. Kon), conradi@idi.ntnu.no (R. Conradi).
URLs: http://www.ime.usp.br/~claudia (C. de O. Melo), http://www.idi.ntnu.no/~dcruzes (D. S. Cruzes), http://www.ime.usp.br/~kon (F. Kon), http://www.idi.ntnu.no/~conradi (R. Conradi).

0950-5849/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2012.09.004

where to focus research efforts from an academic perspective [86]. However, to the best of our knowledge, no study in the literature has investigated the major factors influencing *agile team* productivity.

In a preliminary study [69], we investigated general factors influencing agile team productivity using two industrial case studies. The studied teams reported that the main general factors influencing agile team productivity were (1) team composition and allocation, (2) external dependencies, and (3) staff turnover. They also reported that, among the agile practices, pair programming and collocation most impacted productivity variations.

In the current paper, we extend and refine that work. Our objective is now to provide a better understanding of some factors (and mediators) that impact agile team productivity. We have thus conducted a new industrial, multiple-case study that draws on the general team effectiveness literature. We investigate the following research question:

RQ: Which factors do have an impact, and in what way, on team productivity when using agile methods?

In this extension, we added one large company and two data sources for all three companies: observational field notes and documentation from team retrospectives. We performed a more comprehensive analysis based on these sources and developed more detailed explanations on how the most perceived factors impacted agile team productivity. We developed a conceptual framework for agile team productivity, which provides clarity and focus in the study and drives further discussion around the results [70]. Finally, data from multiple-case studies were analyzed thematically and presented in a thematic map.

As the answer to our research question, we found that *agile team management* is the most influent factor on agile team productivity. Team design choices and staff turnover emerged as relevant intra-team management issues, while inter-team coordination emerged as an important inter-team management issue. In this paper, we present a conceptual framework to support the factor analysis and refine it to incorporate new information based on our findings.

The remainder of this paper is organized as follows. Section 2 presents a review on software productivity definitions and a conceptual framework for agile team productivity. Section 3 describes our research question and method in detail. Section 4 presents results from a multiple-case study performed in three Brazilian companies. Section 5 contains a discussion of our findings, and Section 6 concludes and provides suggestions for further work.

2. Background: productivity definitions and a conceptual framework to study agile team productivity

In this section, we give a short introduction to software productivity and present the conceptual framework that is the basis for our work.

2.1. Defining software productivity

Although productivity has been studied intensively, it remains a controversial issue [100]. First, several concepts are involved in its definition, including effectiveness, efficiency, performance, generating misunderstandings, and term overload [100]. Second, the meaning of productivity varies according to the context [98] and perspective [81]. Finally, there is no consensus on the best way to measure software productivity, both in traditional and agile software development teams, because software development is a human-based activity with extreme uncertainties from the outset, leading to many difficulties in achieving a reliable software productivity definition [100]. The diversity of these aspects hinders any precise approach to define and measure software productivity.

Furthermore, agile teams contain knowledge workers (KWs) working together and sharing the same goal. The product of a KW is typically intangible: knowledge is the addition of meaning, context, and relationships to data or information [19]. Even if there are no universally accepted methods to measure KW productivity, it has been studied in its productivity dimensions, including a KW's perception of his productivity, customer satisfaction, quantity of work, innovation, creativity, timeliness, product quality, absenteeism, profitability, and team efficiency and effectiveness [85]. However, most companies measures productivity in different ways, which makes comparison impossible. In this study, we have analyzed agile team productivity using the team's perception as one potential dimension to understand their overall productivity. Through perceptions, we were able to establish a single dimension for the three different companies.

2.2. Conceptual framework for studying agile team productivity

Though team productivity has been studied in the software development field, the most mature theoretical models addressing teamwork components and productivity outcomes come from organizational behavior area (e.g., [42,27,109,64,87,65]. According to Salas et al. [88], *teamwork* is "a set of interrelated thoughts, actions, and feelings that combine to facilitate coordinated, adaptive performance and the completion of taskwork objectives". Software development, especially agile software development, relies predominantly on teamwork [97]. This literature is, therefore, relevant for studying agile team productivity.

One well-known theoretical model for teamwork effectiveness is the *Input–Process–Outcome* (IPO). In this framework, team effectiveness is a function of input factors and group processes, where both team inputs and processes have important and differentiated impacts on team performance [37]. In software development, IPO frameworks have been applied to analyze the impact of input factors (e.g., team development stage, team characteristics) and group process factors (e.g., coordination, communication quality, knowledge sharing) on team effectiveness, performance, and other outcomes as software quality and job satisfaction [37,93,4,60]. Also in agile development, IPO frameworks (and their evolution) have recently been adopted as an underlying conceptual framework in both quantitative and qualitative studies (e.g., [108,97].

In our work, we adapted three IPO teamwork effectiveness frameworks from Cohen and Bailey [27], Yeatts and Hyten [109,64], aiming to describe a more coherent conceptual framework of agile team productivity. Based on these frameworks, we selected inputs, processes, and outcomes related to the agile values and principles [11]. Fig. 1 presents the novel conceptual framework we use to support the data analysis of agile team productivity. In this new conceptual framework, we classify input into five subgroups (I1–I5), one subgroup to explain group processes (G1), and two subgroups to explain productivity outcomes (O1 and O2). We describe them below.

2.2.1. Input factors

Individual and Group characteristics (I1) describe member and team types. Most theoretical team performance frameworks have included team design characteristics, including team size and composition [109]. Teams sufficiently well designed to perform adequately are more likely to be given additional authority over their work, more supporting resources, and more challenging goals [105]. According to Bell [14], team design could be related to team performance, as it affects the amount of knowledge and skills that team members must apply to the team task.

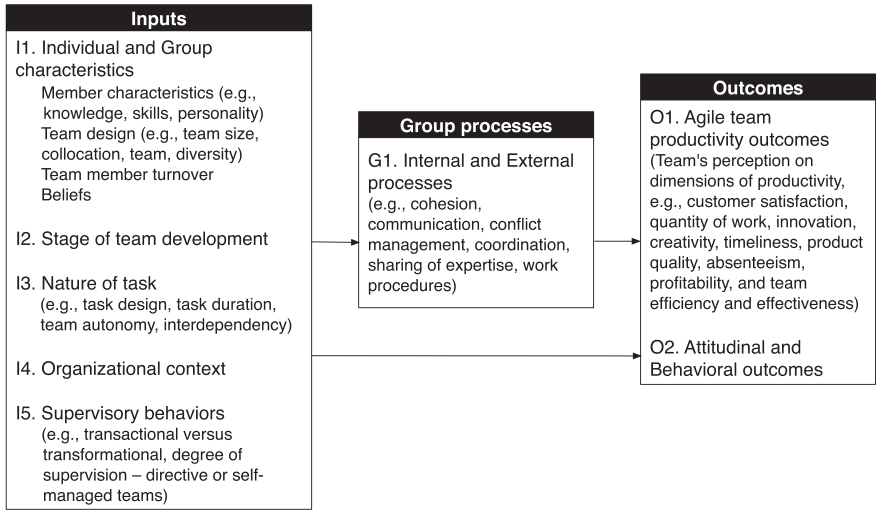

Fig. 1. Our agile team productivity conceptual framework.

The most relevant team member characteristics are knowledge, skills, and personality, while team characteristics include size, diversity, staff turnover, and shared beliefs. Team capabilities and skills are the most significant personnel characteristics that influence software productivity [66,96] and usually play a moderator role in frameworks that aim to explain productivity variations [15,100]. Agile development is essentially people-centric and recognizes the value of team member competencies when bringing agility to development processes [77,54]. Getting the right people with appropriate skills and empowering them are critical for agile development success [25,45]. Team diversity is key for agile development [77], being an XP principle [10]. Teams with broader experience are positively associated with project performance [61]. These claims suggest that group characteristics are an important factor impacting on agile team productivity.

Stage of team development (I2) relates to team maturity. Team members must learn new behaviors and skills to improve their work and create a high-performance team. Several models describe the team development stage, such as the Tuckman [101] classic model of forming, storming, norming, and performing. In the forming stage, team members attempt to create social and task structures to guide their interactions. When they realize that it is difficult to create consensus on a certain approach, they shift to a storming stage, in which different members compete for influence. The team evolves and reconciles differences by setting norms to guide their interactions. Once the norms are well established, members can focus on achieving common goals. As agile methods focus on teamwork, which varies according to the development stage, we included this subgroup in the framework.

Nature of task (I3) includes task design, task duration, the degree of autonomy to execute the tasks, and task interdependencies. Cohen and Ledford [28] argue that enriched tasks allow variety, significance, autonomy, and feedback, which result in high responsibility, motivation, satisfaction, and team performance. In software development, proper task assignment is clearly considered to impact productivity [17], because it can influence team member motivation [83]. Agile teams should have sufficient autonomy to determine which tasks must be performed, demonstrating results at the end of each iteration [7]; we thus included this subgroup in our conceptual framework.

Organizational context (I4) includes variables such as rewards, culture, training, and resources. Collective rewards help motivate groups whose tasks were made interdependent, while individual rewards acknowledge members whose performed tasks reflect individual responsibilities [27]. Several agile teams (including those studied here) work within an organizational environment. We thus included this subgroup as input.

Supervisory behaviors (I5) rely on leadership style whether it is transactional or transformational and whether it guides the team directly or encourages self-management. Transactional leaders usually set goals, obtain team agreement on what is to be accomplished, and monitor team performance. Transformational leaders are inspiring and stimulating, providing followers with a sense of purpose, articulating shared goals and mutual understanding, and an attractive future. To do so, they consider the maturity level, capabilities, and subordinates' needs by treating employees as unique individuals [39]. We added *Supervisory behaviors* because self-management and empowerment are considered key for agile development, supported by agile practices and essential for agile culture [45,90,51].

2.2.2. Group processes (G1)

Interactions among team members and interactions with other teams, customers, and suppliers directly affect team performance [109]. Group processes also mediate the relationship between inputs and outcomes. Team interpersonal processes and work procedures are considered group processes. Examples of group processes are team cohesion, team communication, conflict management processes, and how they coordinate their activities (coordination processes). Moreover, agile methods and their practices are work procedures played by team members that may affect productivity directly or, at least, mediate the relationship between input factors and productivity outcomes. Because agile methods focus on people, teamwork, and their interactions through agile practices, all those processes may have a significant influence on team productivity and were included in our framework.

2.2.3. Outcomes (O1 and O2)

As output, there are some expected outcomes, including *agile team productivity (O1)* and *attitudinal and behavioral indicators (O2)*. As productivity is hard to measure (Section 2.1), we considered agile team productivity as the team's own perception of their overall productivity. Considering that team's perception may vary substantially over time, we added all knowledge worker productivity dimensions as possible outcomes in the model. Attitudinal and behavioral outcomes (e.g., trust and commitment) were included because of their importance in establishing agile teamwork and self-organization [72,74].

3. Research method

We investigated our research question "Which factors do have an impact, and in what way, on team productivity when using agile methods?" To answer this research question, we performed a multiple-case study in the Brazilian IT industry. Case studies [41,104] have high potential for analyzing performance improvement, and they are appropriate for studying complex performance issues [76].

The criteria for case selection included the following: (1) companies using agile methods (XP [10] or Scrum [89]) for at least 2 years; (2) companies in different business segments, geographical location, size, structure, and culture; (3) agile projects with at least four co-located developers and in progress for at least 6 months.

Data collection was carried out in three Brazilian companies, from September 2010 to February 2011. The unit of analysis is a set of three development projects, one in each company. We chose to follow the teams for 6 months because the influence of some productivity factors may change over time, depending on the project context. The staff turnover problem may be noticeable only immediately after a member's dismissal. If we collect the data in a single point in time, this event may not be mentioned during the interviews.

We signed a non-disclosure agreement with the companies; this step was important to establish a formal link between researchers and companies and ensure data confidentiality, so the companies would feel more comfortable with our presence observing their internal activities.

Fig. 2 summarizes the overall research steps. We performed the research steps A1–A4 in our preliminary study [69], here called P1. The current paper details the steps A5–A10, in which we included Company 3 and additional data sources from all three companies. Data were analyzed thematically, then contrasted with a conceptual framework for agile team productivity. We finally revised the conceptual framework and presented agile team productivity and management factors.

3.1. Data collection

Table 2 describes the company and project profiles, considering guidelines provided by Kitchenham et al. [52]. Table 3 shows the agile practice adoption level for each project. If a team used one practice fully, we assigned the term *full*. If they used just a few recommendations of the practice, we assigned *partial*. When they did not use it, we assigned the phrase *Do not use*.

Company 1 is a large financial corporation with over 500 IT employees, which had previously used plan-driven development processes. The company managers decided to adopt agile methods to increase team productivity, and they have been using them for 2 years. The organizational structure and coordination are primarily vertical [40], where project managers usually implement coordination processes. Project 1 is a re-development of an existing system for the financial market involving several institutions. The project started in March 2010 and is estimated to last for approximately 2 years. The team adopted several XP [10] and Scrum [89] practices and used 1-week iterations.

Company 2 has been delivering e-commerce and infrastructure services for over 10 years and has used only agile methods to develop software. It employs approximately 120 developers. The organizational structure and coordination are primarily horizontal [40], where coordination processes are usually provided by an individual team member who communicates directly with other members or users on a one-to-one basis. Project 2 is a new development of an e-commerce service in a market with other competitors. The project also started in March 2010 but does not have a specific deadline, as they are developing software as a service, with continuous improvement and new functionalities. The project adopts several XP, Scrum, and Lean principles and practices.

Company 3 is an important player in Internet content and access provision in Brazil. The organizational structure and coordination are primarily vertical [40], but the hierarchy is smaller than Company 1. The IT department employs approximately 200 developers and had also previously used plan-driven development processes. They have applied agile methods since 2008. Project 3

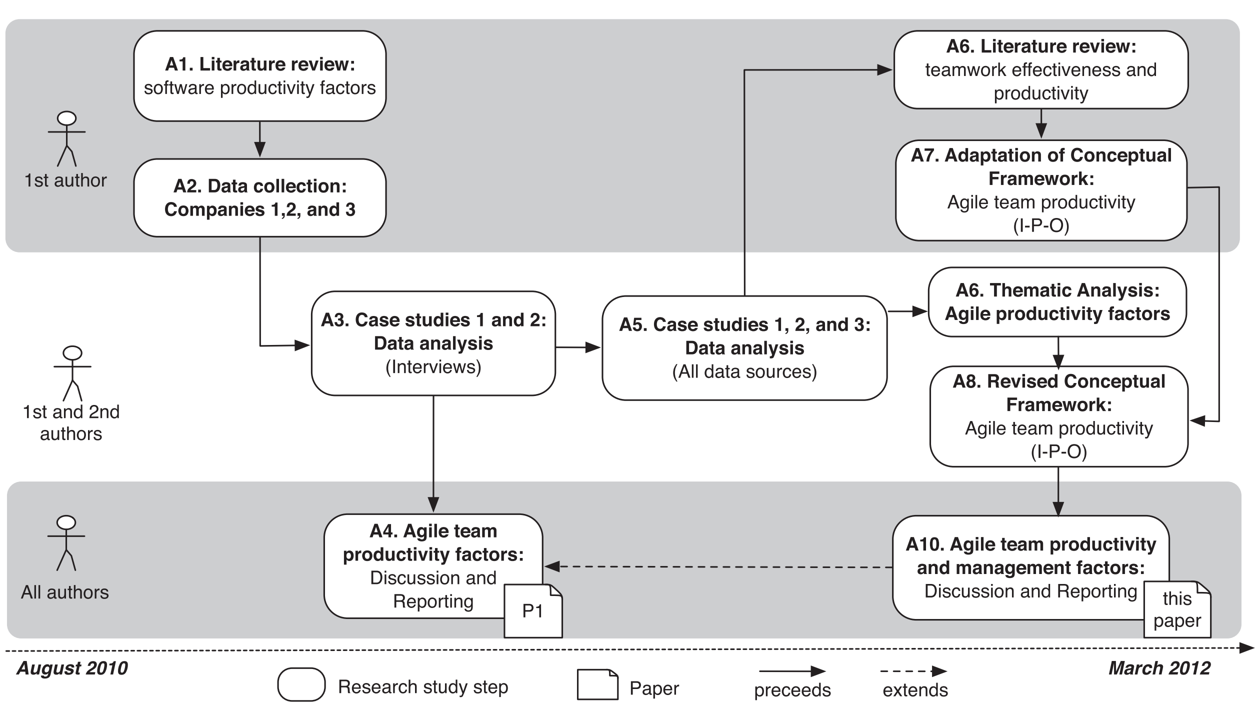

Fig. 2. Overall research steps.

Table 1
Description of the data sources in our study (adapted from Yin [110]).

Data source	Company/Project 1	Company/Project 2	Company/Project 3
Retrospective documentation	27 (1-week) iterations	15 (3 or 4-week) iterations	18 (3-week) iterations
Interviews	3 Full-time developers, 1 part-time developer, 1 product owner, 1 scrum master, 1 project manager Total = 7	1 Project manager/coach, 1 product owner, 4 developers Total = 6	3 Full-time developers, 1 test specialist, 1 webmaster, 1 scrum master Total = 6
Direct observation	Visiting the open office spaces. Field notes taken in the regular work sessions		

is the maintenance of a recommendation system for products from several virtual stores. The project adopts mainly Scrum practices and some XP and Lean principles and practices.

3.1.1. Data collection instruments

The main data collection methods were semi-structured interviews, non-participant direct observations, face-to-face discussions with project leaders, and document analysis (Table 1). The data were collected over a period of 6 months, gathering opinions from different stages of the project. Since we played different roles in the data collection and analysis, from now on we make a distinction between researchers to make each one's participation clearer.

Interviews were semi-structured (Appendix A provides the interview guide) to understand the factors impacting project productivity in the team's perception and how they impacted. The first author of this paper conducted the interviews. This researcher has experience conducting interviews due to her background in requirement elicitation in real projects. Each interview lasted approximately 1 h, and the interviewees were informed about the audio recording and its importance to the study.

We conducted interviews with 19 team members within the three companies, including developers, project managers, and product owners, also considering different experience profiles. We informed all participants of the main research goal but did not give further details, which could have biased their opinions on the research subject.

We developed a protocol (Appendix B) to guide the non-participative observation and register observations about factors impacting team productivity and any other exceptions during the project/team follow-up. The protocol contains questions answered regularly by the observer (daily or per iteration), which was the first author of this paper. We also collected retrospective documentation all the teams maintain such information in spreadsheets or wiki. Table 1 summarizes the data sources used in the study.

3.2. Data analysis

We used thematic analysis to analyze the data, a technique for identifying, analyzing, and reporting standards (or themes) found in qualitative data [20,21,33]. It is a way to recognize patterns in textual data, where emerging themes become categories for analysis [38]. To support the data analysis, data analysis, we used a tool, NVivo 9 [78], which enables information classification into searchable codes.

Thematic analysis has limited interpretative power beyond mere description if it is not used within an existing conceptual framework [21]. We thus adopted the conceptual framework on team productivity (Section 2.2) to anchor our analytic claims. Conceptual frameworks are useful as supports to better delineate qualitative studies and provide some clarity and focus; they can also be used to drive further discussion around the results [70]. Other qualitative studies on agile team effectiveness [74] and agile team

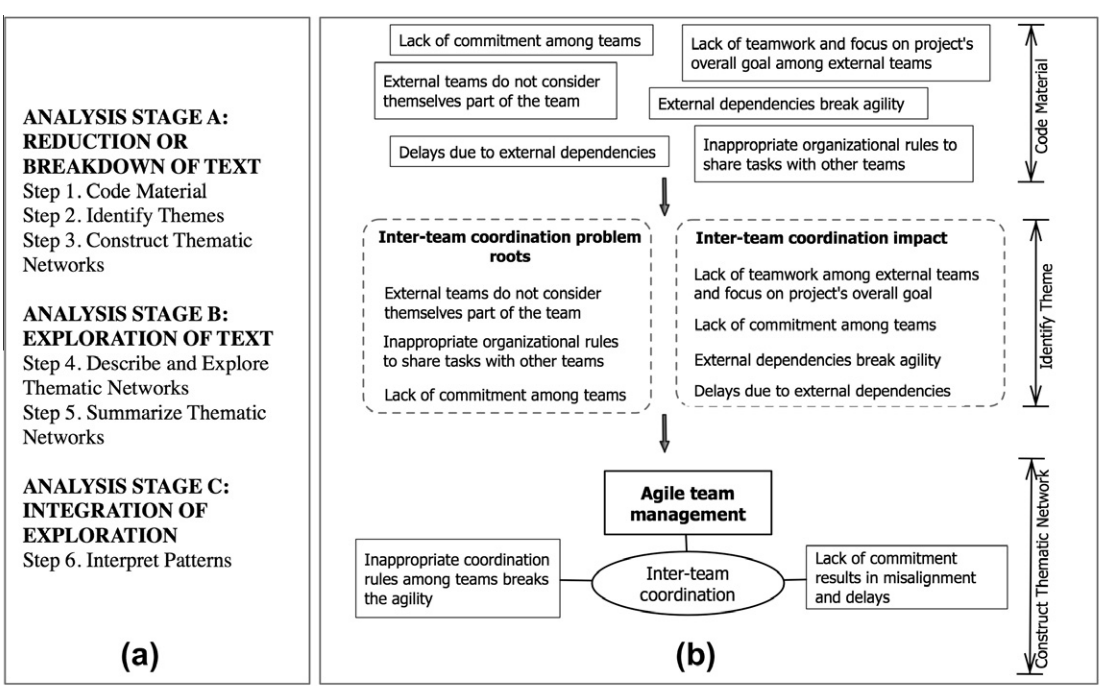

Fig. 3. (a) 3-Phase A–B–C qualitative method, starting with coding and ending up with thematic maps and (b) example of analysis Stages A and B.

communication [82] have also presented conceptual frameworks as strategies to explain their results. It is important to note that we did not use the conceptual framework to guide the thematic analysis, but to discuss and report results. We thus use it to complement, extend, and verify our findings, an approach described by Corbin and Strauss [30] in qualitative research.

To perform and thoroughly describe the thematic analysis, we used thematic networks that summarize the main themes constituting a piece of text [6]. Thematic network analysis contains three main sequential stages: *Stage A – reduction or breakdown of the text*; *Stage B – exploration of the text*; and *Stage C – integration of the exploration*. While they all involve interpretation, a more abstract analysis level is accomplished at each stage. In this section, we describe all stages and the details of how our themes emerged. Fig. 3a summarizes these three stages and subsequent steps.

3.2.1. Stage A – reduction or breakdown of text

Step 1 is to reduce the data, or *Code Material*. This may be performed by dissecting the text into manageable and meaningful text segments using a coding framework. This is a common procedure in qualitative research (e.g., [70,31]. This step in the analytic process is rather rudimentary, but it is imperative that it be completed with great rigor and attention to detail [6].

After transcribing the interviews, the two first authors of this paper performed data coding, naming all possible productivity factors mentioned by the respondents. At this stage, 98 codes were generated. These authors discussed each code before including it in the data collection tool (NVivo 9). After code generation, we reviewed each code in the raw information nature context.

After all the text was coded, *Step 2* involved going through the text segments in each code (or group of related codes) and extracting the salient, common, or significant themes in the coded text segments. We had 12 themes after this step. We next went through the selected themes and refined them further into themes that are (i) specific enough to be discrete (nonrepetitive) and (ii) broad enough to encapsulate a set of ideas contained in numerous text segments.

Finally, the identified themes provided guidance for the thematic networks: *team member turnover*, *team design choices*, and *inter-team coordination*. Fig. 3b illustrates the first stage of thematic analysis and the emergence of the *inter-team coordination* organizing theme and its connection to the global theme found, *agile team management*.

3.2.2. Stage B – exploration of text

In this stage, we returned to the original text, reading it linearly, theme by theme. The goal was to describe and explore the network (Step 4), supporting the description with text segments. All authors of this paper discussed the rationale behind each theme. Once a network has been described and explored, the next step is presenting a summary of the main themes and patterns characterizing it (Step 5). The objective here is to summarize the major themes that began to emerge in the network description and make explicit the patterns emerging in the exploration. Fig. 4 describes the main themes and related patterns.

3.2.3. Stage C – integration of exploration

In this stage, the researcher brings together the deductions in the summaries of all networks and the relevant theory to explore the significant themes, concepts, patterns and structures that arose in the text (Step 6). The aim was to return to the original research questions and theoretical interests underpinning them and address them with arguments grounded on the patterns that emerged from exploring the texts. According to Attride-Stirling [6], this is a complex and challenging task that is difficult to explain procedurally. We address this step in Section 5.

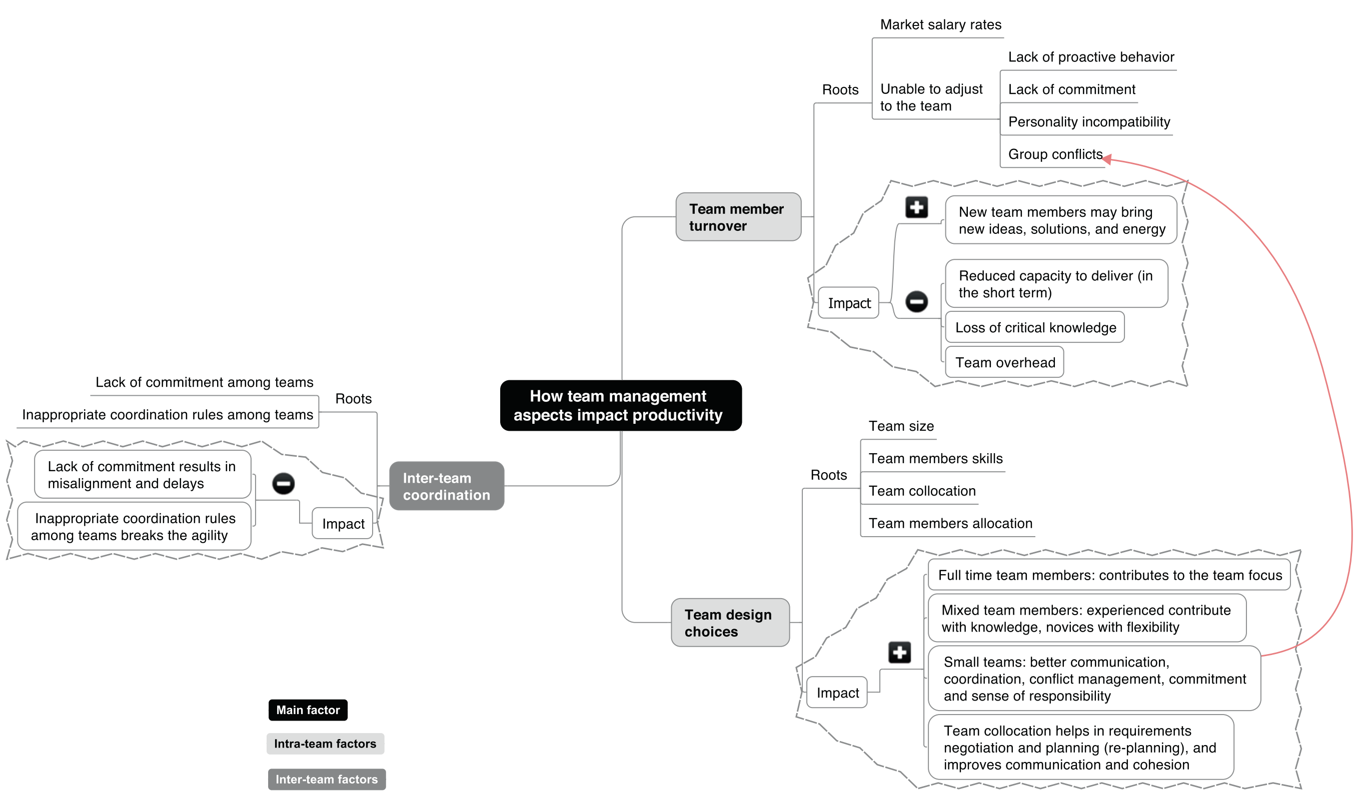

Fig. 4. Thematic map on agile productivity factors.

4. Agile team management and productivity

We describe below results from the thematic analysis, providing a short description for each theme and presenting quotations and other evidence that support the findings.

Fig. 4 presents the results from the thematic network analysis of agile team productivity factors. In the following sections, we describe the organizing themes, *team design choices*, *team member turnover*, and *inter-team coordination*, the main factors impacting agile team productivity, all related to the global theme *agile team management*. The organizing themes have related roots and impacts on agile team productivity.

4.1. Team member turnover

Team member turnover occurred as a factor impacting team productivity. There was some degree of staff turnover in the three teams studied, as Table 2 shows. The teams perceived reduced productivity due to staff turnover.

Turnover can be defined as a type of membership change that involves the departure or arrival of a formally designated team member [5,56]. There are many categories of turnover expenses: separation costs (exit interview, administrative procedures); advertising and recruiting expenses; new employee orientation and training; and decreased productivity until the new employee is ready to contribute [24,102].

In Project 1, at one specific time of the project, many team members left the project at once. Projects 2 and 3 experienced lower, but frequent turnover. Developers said:

> *"There are things that impact the project negatively, etc. One is the staff turnover. People who started the project are no longer here. Everybody now is new."* (Developer, Project 1).
>
> *"There was a drop in productivity at the end of the year when two developers left the project."* (Project manager, Project 2).

The turnover caused negative impact on team productivity and usually occurred due to job offers with better salaries or when a team member was unable to adjust to the team. Teams usually expect that newcomers early adapt to their fast-paced work. When it does not occur, the teamwork might be compromised, affecting productivity:

> "*One of the problems of productivity today is the 'new' guy. This is a productivity issue. He is here about 3 months, but he did not get into the rhythm of the team yet*." (Developer, Project 3).

In Project 2, there was a tension between the QA (Quality Assurance) person and the other developers concerning work procedures. When the QA joined the team, he wanted to change some quality assurance procedures adopted by the team. In fact, the team was struggling with configuration management and testing tasks. The tension not only caused team members dissatisfaction, but also raised many conflicts in the meetings, which could not be completed on time. In the end, the company terminated the QA, which was considered both positive and negative by the team members.

In Project 3, a developer joined the team but disagreed with some team procedures. The team did not accept the proposed changes, and the developer lost motivation. Here, we are not discussing the change merit, but the conflict origin. After a while, the developer left the company. In both cases, the newcomers tried to make changes in work procedures established by teams formed for at least 6 months (Section 3.1) and faced barriers that led to the turnover.

Conversely, team members also mentioned a positive staff turnover influence on their productivity: the opportunity for the

Table 2
Company and project profiles.

Characteristics	Company/Project 1	Company/Project 2	Company/Project 3
Company business	Financial	E-commerce and Infra-structure services	Internet content and provider
Company structure	Vertical	Horizontal	Vertical
Number of IT employees	400	120	200
Project description	Financial system	E-commerce service	Recommendation system
Team composition	6 Full-time developers, 2 part-time developers, 1 scrum master, 1 project manager, 1 product owner (Total = 11 members)	4 Full-time developers, 2 part-time developers, 1 project manager/coach, 1 product owner (Total = 8 members)	4 Full-time developers, 1 webmaster, 1 test specialist, 1 scrum master, 1 project manager, 1 product owner (Total = 9 members)
Language	Java	Ruby	Java
Non-functional requirements	Reliability, availability, performance	Reliability, availability	Performance, availability and auditability
Reuse	High – software product lines, components and other systems	High – open source project and other systems	High – components and other systems
Requirements stability	High stability	Medium stability	Medium stability
Staff turnover	33.3% – Considered medium by the project manager	40% – Considered medium by the project manager	35.3% – Considered medium by the project manager

Table 3
XP and Scrum practices adopted by the projects.

Practices	Project 1	Project 2	Project 3
Code and tests	Full	Full	Partial
Continuous integration	Full	Full	Partial
Daily deployment	Do not use	Partial	Partial
Daily meeting	Full	Full	Full
Energized work	Partial	Full	Partial
Incremental design	Full	Full	Full
Pair programming	Full	Partial	Partial
Real customer involvement	Full	Full	Full
Shared code	Full	Full	Full
Single code base	Full	Full	Full
Sit together	Partial	Full	Full
TDD	Partial	Full	Partial
Ten minute build	Full	Full	Partial
Negotiated scope contract	Do not use	Partial	Partial
Planning game	Full	Partial	Full
Retrospectives	Full	Full	Full
Root cause analysis	Do not use	Full	Do not use
Slack	Full	Full	Full
Stories	Full	Full	Full
Team continuity	Partial	Partial	Partial
Weekly cycle	Full	Partial	Partial
Whole team	Partial	Partial	Partial

team to improve and grow. New team members can bring new ideas and experiences, leading the team to a more mature level. This relates both to the team's ability to handle turbulent environments and the continuous learning ability.

> "*We see new people's arrival on the team in a different perspective. Maybe their proposals seem to be awkward for the team and they [the team] need to mature to a point in which maybe the proposed ideas will be good*." (Developer from Project 2).

In the retrospectives of the three companies, we found evidence supporting the positive side of turnover: new people bring more energy to the group, especially when the team motivation was deficient. However, the results were stronger in Projects 2 and 3, where the turnover wad medium but somewhat frequent. In Project 1, the turnover was also medium but happened once a year. There are many occurrences of positive notes in the subsequent retrospectives after team member turnover. In a retrospective session, teams explicitly divide negative and positive aspects of the previous iteration to recognize positive actions and discuss improvements for the negative ones. We identified positive notes greeting new members, sometimes referring to "*new blood in the team*" – a Brazilian expression that connotes a feeling of renewed energy.

4.2. Team design choices

We found that *Team design choice* is a factor impacting agile team productivity. Team design choices are the member attribute configurations in a team [57]. Our findings indicate desirable team design attributes: full-time allocation, diversity (mixed teams), team member skills, team size, and collocation (physical proximity).

Team composition with different profiles and knowledge levels was considered positive for team productivity, especially in Project 1.

Considering the organizational context and business, Project 1 clearly contains high-ability (business experts, experienced software architects) and low-ability workers (novices). One possible explanation is that the benefits of having mixed team members on such teams may be more noticeable than for the other teams. Experienced team members contributed to the work by adding knowledge, while the others contributed by being flexible, as stated by a developer:

> "*Some things contribute to productivity. We have very experienced people here and less experienced ones that are more flexible*" (Developer, Project 1).

The respondents mentioned that small teams lead to better communication and alignment. In addition, conflict management and coordination among team members are easier to handle. As team size increases, the number of necessary communication links between team members increases and there will be more potential conflicts to manage. In Project 2, some members left the project, and the remaining team members were allocated full-time to the project. A Developer said the following:

> "*As we reduced the team, we are starting to focus more on what we want. Before, it was a bit messy and people did not perform certain tasks because they thought that other people would do it*" (Developer, Project 2).

In the same project, the product owner also noticed the benefits of the full-time allocation:

> "*After the reorganization, nobody needs to move to other activities outside the project. So, they are more focused. Everyone knows everything in the project*." (Product Owner, Project 2).

In Project 1, the team size increased over time, and some team members noted it as a negative factor impacting team productivity. From the team members perspective, small teams also enable a better understanding of the product's big picture because fewer people need to learn and keep up-to-date on the product scope. In addition, respondents said that small teams help increase the team's sense of responsibility and commitment.

Our respondents also mentioned team collocation as an attribute. This result was stronger for Project 1, considering the frequency of references during the interviews. Team members in this project explained that collocated work helps overcome the invisible barriers between teams in a hierarchical company. They noticed improvements in requirement negotiations and risk mitigations through the socializing atmosphere provided by collocation.

Respondents mentioned that their productivity depends on their workspace layout. Both projects were radically collocated [99], but they mentioned that it is not enough to be in the same space. The layout (including desk positions and proximity) may also impact their productivity.

This factor was mentioned more by respondents in Company 1 than in Company 2, possibly because Company 2 has invested in a customized layout that benefits working in pairs, while Company 1 has kept their traditional workspace infrastructure.

4.3. Inter-team coordination

We found that *inter-team coordination* impacts agile team productivity (Fig. 4). There are several kinds of dependencies in a project. Shared resources, prerequisite constraints, simultaneity constraints, and the relationship between tasks and subtasks are common examples of dependencies among activities in a project [62]. Coordination processes are commonly used for managing dependencies among activities [62,63], enabling teamwork among different teams.

In large organizations, software development teams often depend on other teams to accomplish their tasks. These include external customers not collocated in the project; operation teams helping publish versions of the system or data models across different environments (integration, homologation, production); external QA teams verifying compliance between the developed system and organizational rules; and other development teams providing reusable assets to the project.

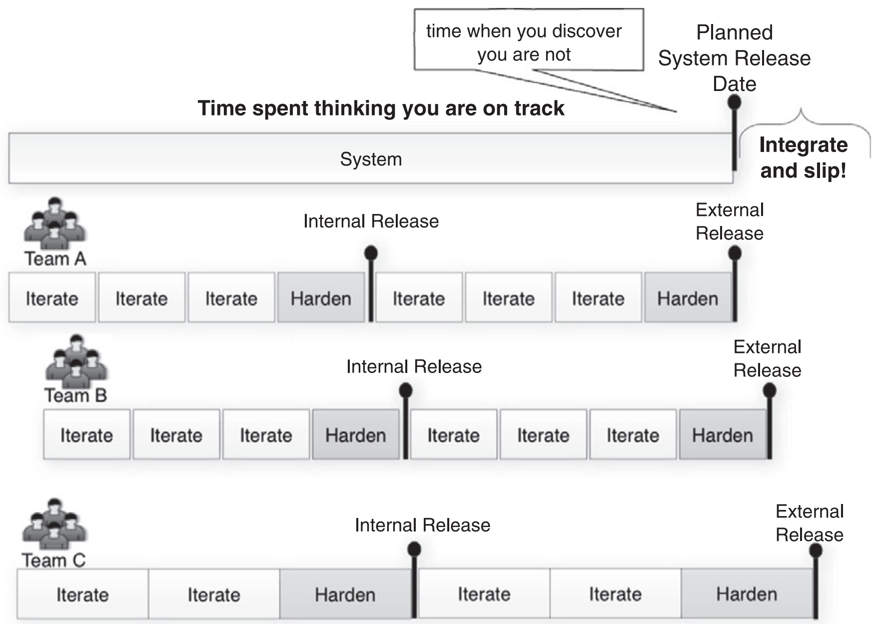

Fig. 5. Unsynchronized agile release pattern (adapted from Leffingwell [55]).

Through the interviews and retrospectives, we have identified documentation that external dependency management represents a recurrent problem in the three studied organizations. Our direct observation field notes corroborate this factor and there were references to this during the daily meetings and plenty of tasks waiting for impediment resolution resulting from external dependencies.

Companies coordinate dependencies among teams or resources using certain strategies. When Project 1 delivers the system to the test environment, it must submit some artifacts, such as data models, to the QA team. The QA team is an example of a resource shared among all enterprise projects in the company. It implements a "first come/first served" coordination process [62] to manage requests from other teams. According to the team, this kind of coordination solution is misaligned with the pace of the agile project:

> "*The other teams are not working at the pace of the project, they are not working in the (same) way, etc. The organization is not ready yet*" (Developer, Project 1).

A similar problem occurs when Project 2 publishes the system to the corporate environment:

> "*Currently, there is one factor that we are dealing with after a lot of feedback from our retrospective, which is about the relationship among the boundaries of development, testing, and production. Whenever we cross these boundaries, a bottleneck occurs.*" (Project Manager, Project 2).

Another external dependency problem occurs when an agile team decides to reuse components or existing systems, building a system of systems. External components and systems are often evolving and have their own lifecycle. Agile teams must sometimes wait for some components, which may compromise a timely delivery, the "unsynchronized agile release pattern" [55]. Fig. 5 illustrates the lack of synchronization among teams that are working on the same project. The main team (Team A) requests components or services from external teams, such as Teams B or C. While waiting for the requests to be completed, the main team thinks it is on track. However, when the planned system release date arrives, the team slips into integration problems.

Project 1 reuses several components and must implement interfaces to communicate with other systems. The problem with the integration with components and systems is thus:

> "*The productivity at the end of the first phase of the project will be somewhat lower because we will have solved a lot of problems, etc., which are the integration with other systems, publishing to a server with other business components.*" (Project Manager, Project 1).

Project 3 also faces the same inter-team coordination problem, even when the other team uses agile methods:

> "*Sometimes we have integration with other systems, internal but complicated ones. You're committed, you're on deadline to deliver, but the other team is not; or the other (team) can even be committed, but they are not able to respond on time. They are also working in some release, and probably will tell you: 'oh, to make such a change, just in the next sprint*'" (QA, Project 3).

These solutions for managing dependencies are thus not compatible with the team needs. For agile projects, it is not only important to manage dependencies, but also to resolve them in a timely manner. Thus, agile teams must be more synchronized to achieve this final goal [55].

Respondents also mentioned that external teams sometimes are not committed to the project goal, only with the execution of the requested task. In general, they do not consider themselves part of the team and tend to overemphasize their importance in the process. A Developer comments:

> "*There are a lot of roles, such as 'I am the system administrator', 'I am the QA.' They do not understand that we're a multidisciplinary team. Reducing the conflict between these roles can simplify our work process*" (Developer, Project 2).

The emphasis on their specific roles denotes not only a lack of teamwork spirit, but also an attempt to use their roles as a means to enforce the choice of practices to use by each team member. A

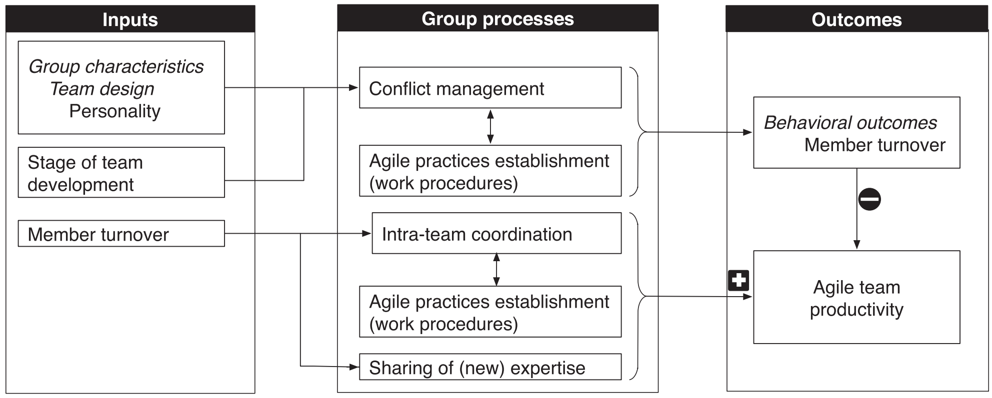

Fig. 6. Staff turnover factors and effects on agile team productivity.

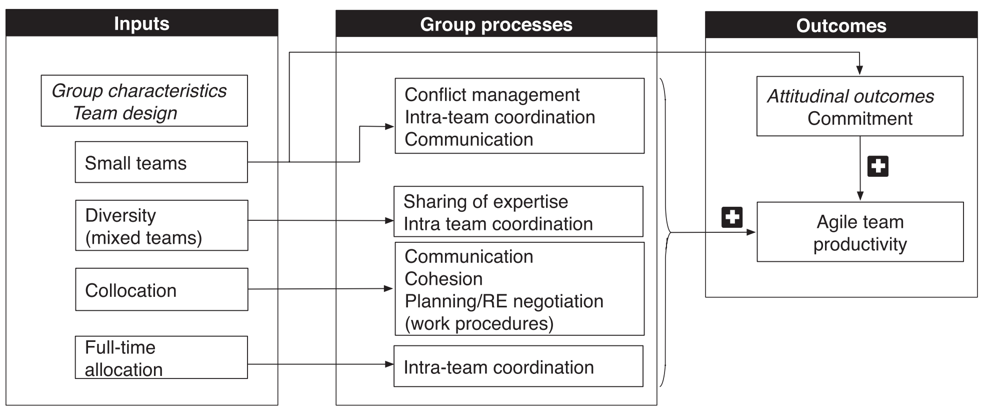

Fig. 7. Team design choices factors and effects on agile team productivity.

system administrator would thus use his/her "role" to determine how the team should proceed in administrative matters. This seems to cause more trouble than potential benefits in the inter-team coordination processes definition.

5. Discussion

Through an interpretative field study in three large companies in Brazil, we investigated factors that affect agile team productivity. We now discuss the cases in light of our research question, RQ: "Which factors do have an impact, and in what way, on team productivity when using agile methods?"

The productivity of the studied agile teams was more sensitive to *team management issues*. Agile teams often take responsibility for managing their own work and behaviors; while others usually make decisions about goals, team structure, and organizational supports [68]. However, often these "other" choices influence the team, and will later require attention from the team members. Our analysis indicated *team design choices* and *staff turnover* as two intra-team management factors with productivity impact. It also indicated *inter-team coordination* as a single inter-team management factor with productivity impact. The intra-team and inter-team factors are further discussed in Sections 5.1 and 5.2 respectively.

5.1. Intra-team management factors and a revised conceptual framework

We will now revise the initial conceptual framework (Fig. 1, Section 2.2), based on consolidated insights from further *multiple-case studies* in the mentioned three IT companies, all regarding agile team productivity factors. Figs. 6–8 depict the revised framework, where we have identified in detail how the three "new" team management factors namely *member turnover*, *team design choices*, and *inter-team coordination* provoke changes in agile team productivity.

We revised the framework by using the following process. For each theme from the thematic map (Fig. 4), we analyzed its roots and impact on productivity, as well as the existing links on the original conceptual framework. When the link already existed, we marked the impact as either positive or negative. Otherwise, we created the links between inputs and outcomes, also considering possible group processes mediating the relationship. Small teams generally led to better communication, easier conflict management and coordination, and, ultimately, agile team productivity. We thus created, in Fig. 6, a link between the input *Small teams* and related group processes, including *conflict management*, *coordination*, and *communication*. Afterwards, we assigned a link between these processes and the team productivity outcomes, because they appear in the context of factors that impacted the agile team productivity. Finally, we marked the type of the impact in this case, positive.

Member turnover The initial conceptual framework proposed staff turnover as an input factor that impacts team productivity, mediated or not by some group processes. In our findings, turnover is an input and outcome, serving as input back into the team processes.[1]

Staff turnover is a common team trouble spot for any software development project [2,107]. Coram and Bohner [29] state that

[1] In fact, turnover has occupied both an input role and an outcome role in traditional IPO models of teamwork effectiveness [103].

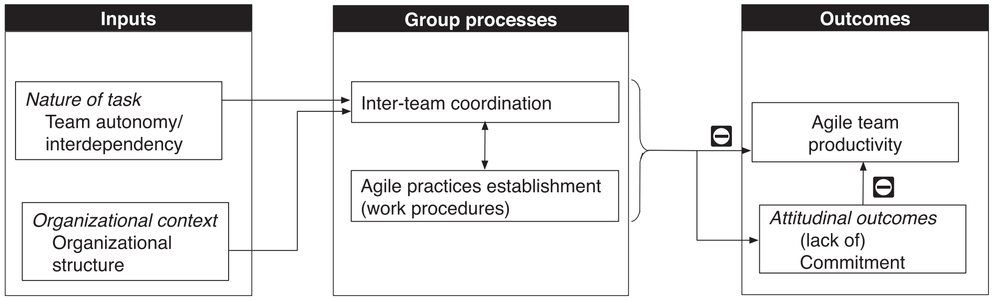

Fig. 8. Inter-team coordination factors and effects on agile team productivity.

high turnover in an agile team can lead to losing critical knowledge, due to the lack of documentation. The turnover may happen for reasons originating within the team, such as personal disagreements, or because of external circumstances, including retirement or job opportunities elsewhere. The team must anyhow adapt to such turnover, despite the reasons for it.

Our results show a negative impact of turnover on agile team productivity. Despite being considered medium in the three companies, staff turnover emerged as critical for team productivity in the interviews and retrospectives. Our observation notes also contain records on teams struggling to deliver stories in the subsequent iterations after the turnovers. Agile teams rely mostly on people and teamwork, and provide many techniques that foster communication, collaboration, and backup behavior, such as pair programming, daily meetings, and sit together. These are suited to reduce or mitigate the impact of turnover in a large degree. However, it was surprising that, even adopting such practices, the teams were visibly affected by medium turnover.

Most empirical research on turnover analyzes its impact on team productivity (and other group performance indicators) without clarifying possible mediation processes [103,56]. Knowledge regarding the specific group processes that might intervene with these indicators is therefore limited [22,49].

We observed (and the team members reported) many conflict episodes regarding agile work procedures, especially quality and configuration management. Our analysis suggests that *personality* and the *stage of team development*, mediated by the selected *conflict management processes* may result in increased *member turnover* and impact on *agile team productivity*. In general, teams in the storming stage [101] face conflicts, reconcile, and evolve by setting new norms. When the conflict management does not succeed, teams expect to observe commitment loss, leading to dismissals and turnover. This happened several times in the observed teams. On some level, they were not able to manage the conflicts regarding the work procedures, leading to turnover and decreased productivity in the short term (teams were unable to deliver for a while), as well as loss in both knowledge and team overhead after the turnover.

Our findings on the influence of *personality* in the conflicts arising in team work (Fig. 7) are consistent with Licorish et al. [58], who observed that the greater diversity of individuals involved in agile teams, combined with the less rigid nature of their involvement, may increase the incidence of personnel incompatibilities and, therefore, the potential for conflict. Hoda et al. [47] found that some team members may not have the desired attributes to be part of an agile team and are perceived to pose a threat to the proper functioning and productivity of a self-organizing agile team. *Personality* and individual practices were seen as root causes that led to turnover. This is somewhat consistent with our results, as *personality* was an input that influenced member turnover, which implies decreasing productivity. However, Hoda et al. [47] assume that the team was doing well and an individual caused malfunction. Our results, in contrast, suggest that both the team development stage and conflict management processes matter and interact with personality, resulting in higher team member turnover.

Though agile methods embrace conflict and dialectics [77], our results show that there is a limit of tension and conflicts that teams can tolerate. Empirical evidence from teamwork literature has supported the negative relationship between conflict and team productivity because it produces tension, antagonism, and distracts team members from performing their tasks [34]. Conversely, according to McAvoy and Butler [67], it is possible to maintain cohesion while introducing low levels of positive conflict in an XP team, which will guarantee better decision-making and learning outcomes. In such self-organized teams, some level of process conflict seems inevitable because there is no legitimate authority to enforce process rules or prevent process conflicts (disagreements about assignments of duties and resources) [13]. However, it is still not clear which degrees (and types) of conflict are healthy for agile teams. Our results suggest that process conflicts, including how to accomplish and divide work, may decrease agile team productivity by causing team member turnover.

Member turnover also plays an input role in our results (Fig. 7), as it generates other indirect effects on agile team productivity. The agile teams had to self-adapt and reorganize their routines, which took time and negatively impacted team productivity. Conversely, new team members may bring new ideas, solutions, and energy to *establish agile practices* and make improvements in the *team coordination processes*. Those changes positively impact agile team productivity. Our results confirm previous research on teamwork (not specifically on software development) [103] that acknowledges high turnover as a potential disrupter of routines, norms, group composition, and a way to introduce new ideas, all of which have important implications for team effectiveness and team productivity.

Team design choices appeared to contribute a great deal to agile team productivity, corroborating previous research on teamwork [27,109,66,25,96]. Fig. 6 depicts our findings in light of the conceptual framework presented in Fig. 1, Section 2.2.

First, the respondents mentioned that *small teams* led to better *communication*, alignment, and *commitment*. In addition, *conflict management* and *intra-team coordination* were easier to handle. Such responses confirm findings from other studies [16,15,23]. As team size increases, the number of necessary communication links between team members increases, leading to more potential conflicts to manage. Having a small agile team thus leads to higher productivity. We also observed a relationship between the *small teams* and smaller turnover, probably due to a reduction of intragroup conflict. This result emerged in our thematic map (Fig. 4) and is better explained through the revised conceptual framework in Fig. 6.

Second, team *diversity* (mixed teams) emerged as a factor contributing to agile team productivity. Our results showed that agile teams with a mix of experienced and non-experienced workers may have a positive impact on productivity due to the knowledge and flexibility they can provide, respectively. This result suggests that there are benefits in maintaining agile teams with diversity in experience. Our results are consistent with previous research on manufacturing teams [43], where workers may have both technical and collaborative skills (such as flexibility) to be more productive. In the same direction, Lee and Xia [54] found that diversity is an important team variable to build team software development agility; however, they did not report a relationship between knowledge and flexibility in mixed teams.

Third, team *collocation* intends to improve communication and collaboration among team members. Both Scrum and XP recommend collocation as an agile practice. By adopting such a practice, companies also hope for productivity enhancement [99], but there are advantages and disadvantages in using collocation in software development [36,46,84,99,44]. Our results confirm previous research, as the projects reported significant productivity gains through improvements in *communication*, teamwork spirit (*cohesion*), *planning*, and requirements *negotiation* when collocated. Lack of privacy, work interruptions, lack of individual recognition, and some disconnection from the rest of the teams were mentioned as the negative side of collocation. Finally, *full-time allocation* of team members was considered positive for overall agile team productivity. It enhances team focus to complete tasks, decreases work interruptions and distractions, and increases team member awareness of the project situation. When team members were allocated part-time to the projects, they were not able to follow the project status well, even by participating in daily meetings. Our results complement previous research on agile team effectiveness [73], showing that developers working on two or more projects in parallel had to manage conflicts among different team goals or needs, damaging a self-managed team's potential.

5.2. Inter-team management factors and a revised conceptual framework

Recent research has been devoted to understanding agile intra-team coordination [94,91] and its relationship with agile team effectiveness and productivity [74,71]. However, agile teams are often embedded in large organizations, dealing with other teams to accomplish their goals. In large companies, such as the ones we studied, it is common to find agile projects with several teams, many of them sharing various projects. Despite initial success at the team level, some teams then find it difficult, if not impossible, to implement agile methods beyond their own boundaries [3].

Coordination between software development teams is one of the most difficult-to-improve factors of software engineering; its importance increases as software development becomes distributed [12]. In fact, teams do not function "in a vacuum", and all external activities (so-called boundary activities) may influence team performance and effectiveness [48]. Fig. 8 depicts our findings in light of the conceptual framework presented in Fig. 1, Section 2.2.

Our findings indicated that *team interdependencies*, mediated by *inter-team coordination* processes, result in decreased agile team productivity. The interdependencies range from QAs, operations, and external customers to maintenance and other project teams. The coordination strategies seem to be wrong, especially for handling the agile team pace and priorities. Using queues to request tasks between agile teams is not a good coordination strategy because it does not handle synchronization properly. Moreover, it was difficult to establish priorities in a timely manner between two agile teams due to the (natural) presence of uncertainty in their projects. Inappropriate coordination rules break team agility, resulting on delays and not achievement of the iterations goals.

The negative impact was more notable in Company 1, whose organizational structure is more rigid and coordination processes between units are primarily vertical (via supervisors, line managers, or other hierarchy representatives). In Companies 2 and 3, we observed both vertical and horizontal coordination. vertical coordination are usually implemented through project managers, while linkage in horizontal coordination is provided by an individual team member who communicates directly with other members or users on a one-to-one basis [79,40]. The organizational structure thus seems to accentuate the negative impact of some inter-team coordination processes on agile team productivity. However, the intensity of this relationship should be further explored.

Although agile teams follow the organizational procedures to send requests to other teams, they notice that the other teams are not really committed to their projects, which impacts agile team productivity. Our interpretation, based on all data collected and observations, is that the adopted inter-team coordination procedures do not favor establishing common goals among teams, which leads to the observed lack of commitment. Our results confirm previous research on team performance, where coordination process choices influence commitment and clear mission establishment, which, in turn, impact team performance [79]. Our results also shed light on the research topic suggested by Abrahamsson et al. [3] regarding synchronization practices of agile and non-agile functions.

5.3. Limitations

There are a number of limitations to this study. First, qualitative findings are highly context- and case-dependent [80]. Three kinds of sampling limitations typically arise in qualitative research designs: cases that are sampled for observation (because it is rarely possible to observe all situations); time periods during which observations took place (problems of temporal sampling); and selectivity in the people who were sampled either for observations or interviews or in document sampling. In pursuit of a trustworthy study, Lincoln and Guba [59] proposed four main characteristics to which a qualitative study should pay attention: credibility, transferability, dependability, and confirmability.

To promote *credibility*, we adopted well established research methods and developed an early familiarity with the organizations culture through preliminary visits. Although we have used a purposive sampling of informants, we tried to include as many participants as possible from each team, considering similarities, dissimilarities, redundancies, and varieties to acquire greater knowledge of the wider group. We also triangulated data from three different qualitative sources: interviews, direct non-participative observation, and retrospective's documentation. Interview data were our primary indicators of productivity factors. The other two sources, nevertheless, influenced the emergence of the main factors in a significant way.

Credibility of a thematic synthesis also considers how well codes and themes cover data, i.e., no relevant data can be inadvertently or systematically excluded or irrelevant data included [32]. We analyzed codes and grouped them systematically, adding more companies and different data sources to the analysis. We frequently referred to the data to ensure that codes were representative and check the relationship among codes and themes.

Confirmability is concerned with how the extracted data are coded and sorted and whether various researchers and experts would agree with the way those data were coded and sorted [59]. In this study, two researchers coded the data and agreed on each piece before adding them into the NVivo 9 tool for information classification.

Dependability concerns data stability, the degree to which data change over time, and adjustments made in the researchers' decisions during the synthesis process [59]. We described the changes that occurred in the companies, which helped us find some productivity factors. Cruzes and Dybå [32] suggest complementary coding methods and establishing an audit trail that will allow an external reviewer to examine the processes whereby data were extracted and coded. However, due to the non-disclosure agreements, we did not provide the audit trail for external researchers, but only those participating in the research.

Transferability refers to the extent to which the findings can be transferred to other settings or groups [32]. To promote transferability, we described the selection and characteristics of each case, including context and settings, data extraction, and synthesis process, as well as quotations with our major findings.

Another possible limitation is that we based much of our data on team perceptions. Once one productivity issue is solved, teams hardly notice its impact. For instance, all three projects substantially reuse software that is an acknowledged productivity factor [75,9]. However, teams had reached a good reuse level; its benefits were perceived, but with much less intensity than other factors emerging at the time of the study. To reduce the impact of this effect, we observed and interviewed teams over a period of 6 months, which allowed us to study the phenomena from different viewpoints as they emerged and changed. Participants might be influenced by turnover events outside the study timeframe. We thus triangulated perceptions, retrospectives notes and observation notes to overcome this limitation.

Finally, the IPO model has served as a valuable guide for researchers [65], where the input/outcome relationships are an important first step in any research program [103]. However, due to its limited and static perspective on team effectiveness and the dynamic processes that underline it [53], IPO-style investigations may be more of the exception than the rule in modern-day organizations [65]. Future studies should thus explore other contemporary theories that attempt to explain teamwork effectiveness, performance, and productivity in a more dynamic perspective.

6. Conclusion

We have conducted a multiple-case study of agile teams in three large Brazilian IT companies for 6 months. The results presented here are based on detailed and rigorous investigations of three teams, summarized in Table 2. This paper sheds light on under-researched questions pertaining to the productivity factors of agile teams.

The major original contributions of this paper are (1) to explore, in an industrial setting, a significant agile productivity factor: *team management*, (2) to analyze team management underlying mechanisms using a novel conceptual framework, and (3) to detail the conceptual framework by adding links among its components, enabling further tests through confirmatory studies. We give new insights on possible cause–effect relationships between staff turnover, team design choices, and inter-team coordination factors and the observed productivity outcomes.

In agile software development, few studies discuss the implications of staff turnover, considering both intervening processes and outcomes. For instance, conflict types and conflict management processes and their impact on group performance are issues that have received little research attention [13]. Our findings shed light on inputs, such as *personality* and *stage of team development*, and group processes, such as *conflict management* and *agile practices establishment*, which result in turnover in agile teams. We also offer possible explanations on the positive impact of turnover on agile team productivity, mediated by group processes, including *intra-team coordination*, *work procedures establishment*, and *sharing of new expertise*.

Team design choices remain an important factor impacting team productivity, and it is even more pronounced on agile teams that rely on teamwork and people factors. More research on tool support for agile team composition, such as that conducted by Licorish et al. [58], is needed. Our results indicated that *team size*, *diversity*, *personality*, *skills*, *collocation*, and *time allocation* are key factors to be considered when designing agile teams.

Our findings suggest that companies may need to review their organizational structures and determine the fit between their structure and agile teams. Company structure decisions directly influence the coordination process selection among teams. Achieving alignment between teams is challenging and, in our view, poses a problem for both corporate-level and team management. Inter-team coordination emerged as a productivity factor in agile teams, but the teams themselves cannot change organizational processes. The *intra-team coordination processes* must be adjusted to enable productive work by considering priorities and pace between teams. Based on our preliminary findings regarding pace and priority issues, there is an avenue for further research on the influence of inter-team coordination strategies and structure on agile team productivity.

Because agile methods are people- and team-oriented [1], establishing teamwork and managing people are crucial to deploy agile methods effectively. The most important implication to managers working with agile methods is that it places more emphasis on people factors in the project, so the "attention to the human issues gives agile projects a particular feel" [26]. People factors directly influence the ability to work in teams, and our results have shown that not only skills, but also diversity, size, collocation, and full-time allocation matter when discussing agile team productivity.

Teams should be aware of the influence and magnitude of turnover, which has been shown, in most cases, negative for agile team productivity. Turnover has a disruptive effect on teams, becoming a particular challenge to self-organized teams. Project managers and team members should learn how to recognize the signs of disruptive conflicts to prevent productivity threats. Teams should also invest time exploring different modes of conflict management to keep issues under control. Human resources processes may be helpful in helping teams on solve turnover and subsequent team design issues.

Finally, there are many possible directions for future research based on our results. Identifying productivity factors in a social–technical system is a challenge, but it should not be neglected. Research on productivity monitoring in agile teams may help the teams to learn more about their own capacity to deliver and work as a team. Team members should be educated to understand and cope with productivity factors on a daily basis because they are self-managed. Likewise, researchers and companies should investigate appropriate strategies for inter-team coordination that consider agile team adaptivity and continuous delivery. Teams pace and priorities are important properties to be considered in this modelling. More research is needed to identify links between theoretical teamwork components to establish enlightening cause–effect relationships that help keep or improve agile team productivity. Quantitative data would provide valuable insights regarding the strengths of these relationships.

Acknowledgements

We would like to thank the companies and their employees who contributed to this project. This research is supported by FA-

PESP, Brazil, proc. 2009/10338-3, CNPq, Brazil, proc. 76661/2010-2, and the Research Council of Norway, proc. 202657.

Appendix A

Interview guide

- What is your role in the project and how long have you been working on it?
- How is the team's 'way of work'? What is your role in the team?
- How does the prioritization of feature work?
- How does the team judge, during prioritization, the value to be delivered?
- In your opinion, has the customer realized the delivered value in each iteration and release? Does he report this?
- In your opinion, has the team delivered value to the customer?
- What is the project status (scope, cost, time box, etc.)?
- What is your opinion regarding the project productivity? How do you perceive this productivity?
- What is your opinion about project quality? How do you track the external quality? And internal? How do you handle the bugs?
- Is there any re-work on the project? How much?
- What do you think that most influences your team's productivity?
- In your opinion, which changes were recently done in the team way of work that can have influenced any productivity variation?
- Do you consider the project motivating?
- Is there anything demotivating you in the project? And in the company?
- Is there anything in the agile methods that motivates you in anyway?
- Is there any kind of waste that jeopardizes the project?
- If you could choose three things to increase productivity, what would them be?
- Do you think that the use of agile methods increases team productivity? Why? Is there something in Agile that helps your own productivity or the productivity of your team?
- Is there anything in the agile methods that decreases your individual productivity or the team productivity? If so, what is it? Why?

Appendix B

Observational protocol – questions and frequency

- Is there any mention about events that are affecting team productivity during the daily meeting? (Daily)
- Is there any mention about events that are affecting team productivity during the retrospective? (Per iteration)
- Is there any mention about events that are affecting the productivity during conversations with the team? (Daily)
- What is the suitability of the workplace to do creative work, e.g., windows, natural light, size of room and desk? (Per iteration)
- What are the ways in which all actors interact and behave toward each other? (Daily)
- Was there anything unexpected? (Daily)

Appendix C

Company and project profiles protocol – questions and scale

- What is the company business? *Open question*
- How the company is structured? *Scale: Vertically, Horizontally*
- What is the total number of IT employees in the company? *Open question*
- What does your project deliver? *Open question*
- What is your team composition? *Open question*
- What is the programming language used by the team? *Open question*
- What are the most important non-functional requirements in the project? *Open question*
- What is the software reuse degree in the project? *Scale: Low – we reuse few assets from the company; Medium – we reuse considerable assets from the company; High – we reuse many assets from the company*.
- Could you give some examples of the reused assets? *Open question*
- How stable are the requirements in the project? *Scale: Low, Medium, High*
- What is the staff turnover rate in the project? *Scale: Low, Medium, High*
- What is the staff turnover rate (considering just the studied period)? *Formula: average number of staff throughout the project divided by the number of leavers).*

References

[1] N. Abbas, A.M. Gravell, G.B. Wills, Historical roots of agile methods: where did agile thinking come from?, in: Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing, vol 9, Springer, Berlin, Heidelberg, 2008, pp. 94–103.
[2] T. Abdel-Hamid, S.E. Madnick, Software Project Dynamics: An Integrated Approach, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991.
[3] P. Abrahamsson, K. Conboy, X. Wang, 'lots done, more to do': the current state of agile systems development research, EJIS 18 (4) (2009) 281–284.
[4] S.T. Acuna, M. Gmez, N. Juristo, How do personality, team processes and task characteristics relate to job satisfaction and software quality?, Information and Software Technology 51 (3) (2009) 627–639
[5] H. Arrow, J.E. McGrath, Membership dynamics in groups at work: a theoretical framework, Research in organizational behavior 17 (1995) 373–411.
[6] J. Attride-Stirling, Thematic networks: an analytic tool for qualitative research, Qualitative Research 1 (3) (2001) 385–405.
[7] S. Augustine, B. Payne, F. Sencindiver, S. Woodcock, Agile project management: steering from the edges, Communications of the ACM 48 (December) (2005) 85–89.
[8] V. Balijepally, R. Mahapatra, S.P. Nerur, K.H. Price, Are two heads better than one for software development? The productivity paradox of pair programming, MIS Quarterly 33 (1) (2009) 91–118.
[9] V.R. Basili, Viewing maintenance as reuse-oriented software development, IEEE Software 7 (1) (1990) 19–25.
[10] K. Beck, C. Andres, Extreme Programming Explained: Embrace Change, second ed., Addison-Wesley Professional, 2004.
[11] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R.C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Manifesto for Agile Software Development, 2001. <http://agilemanifesto.org/>.
[12] A. Begel, N. Nagappan, C. Poile, L. Layman, Coordination in large-scale software teams, in: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. CHASE '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 1–7.
[13] K.J. Behfar, E.A. Mannix, R.S. Peterson, W.M. Trochim, Conflict in small groups: the meaning and consequences of process conflict, Small Group Research 42 (2) (2011) 127–176.
[14] S.T. Bell, Deep-level composition variables as predictors of team performance: a meta-analysis, Journal of Applied Psychology 92 (3) (2007) 595–615.
[15] J. Blackburn, G. Scudder, L.N. Van Wassenhove, Concurrent software development, Communications of ACM 43 (November) (2000) 200–214.
[16] J.D. Blackburn, G.D. Scudder, L.N. Van Wassenhove, Improving speed and productivity of software development: a global survey of software developers, IEEE Transactions on Software Engineering 22 (December) (1996) 875–885.
[17] B. Boehm, Software Engineering Economics, Prentice Hall, 1981.
[18] B.W. Boehm, Improving software productivity, Computer 20 (9) (1987) 43–57.
[19] P. Bosch-Sijtsema, V. Ruohomki, M. Vartiainen, Knowledge work productivity in distributed teams, Journal of Knowledge Management 13 (6) (2009) 533–546.
[20] R.E. Boyatzis, Transforming Qualitative Information: Thematic Analysis and Code Development, Sage Publications, 1998.

[21] V. Braun, V. Clarke, Using thematic analysis in psychology, Qualitative Research in Psychology 3 (2) (2006) 77–101.
[22] R. Brian, R.A.N. Dineen, The impact of team fluidity and its implications for human resource management research and practice, Research in Personnel and Human Resources Management 22 (2003) 1–37.
[23] F.P. Brooks Jr., The Mythical Man-month, anniversary ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
[24] F.W. Cascio, Managing Human Resource: Productivity, Quality of Work Life and Profits, third ed., McGraw Hill, New York, 1991.
[25] T. Chow, D.-B. Cao, A survey study of critical success factors in agile software projects, Journal of Systems and Software 81 (6) (2008) 961–971.
[26] A. Cockburn, J. Highsmith, Agile software development: the people factor, Computer 34 (2001) 131–133.
[27] S.G. Cohen, D.E. Bailey, What makes teams work: group effectiveness research from the shop floor to the executive suite, Journal of Management 23 (3) (1997) 239–290.
[28] S.G. Cohen, G.E. Ledford, The effectiveness of self-managing teams: a quasi-experiment, Human Relations 47 (1) (1994) 13–43.
[29] M. Coram, S. Bohner, The impact of agile methods on software project management, in: Proceedings of the 12th IEEE International Conference and Workshops on Engineering of Computer-Based Systems., IEEE Computer Society, Washington, DC, USA, 2005, pp. 363–370.
[30] J. Corbin, A.C. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, third ed., Sage Publications, Inc., 2007.
[31] J.M. Corbin, A. Strauss, Grounded theory research: procedures, canons, and evaluative criteria, Qualitative Sociology 13 (1990) 3–21.
[32] D.S. Cruzes, T. Dybå, Recommended steps for thematic synthesis in software engineering, in: Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2011, pp. 275–284.
[33] D.S. Cruzes, T. Dybå, Research synthesis in software engineering: a tertiary study, Information and Software Technology 53 (5) (2011) 440–455.
[34] C.K.W.D. Dreu, L.R. Weingart, Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis, Journal of Applied Psychology 88 (4) (2003) 741–749.
[35] T. Dybå, E. Arisholm, D.I.K. Sjberg, J.E. Hannay, F. Shull, Are two heads better than one? On the effectiveness of pair programming, IEEE Software 24 (November) (2007) 12–15.
[36] M. Eccles, J. Smith, M. Tanner, J. Van Belle, S. van der Watt, The impact of collocation on the effectiveness of agile is development teams, Communications of the IBIMA (2010) 1–11.
[37] S. Faraj, L. Sproull, Coordinating expertise in software development teams, Management Science 46 (12) (2000) 1554–1568.
[38] J. Fereday, E. Muir-Cochrane, Demonstrating rigor using thematic analysis: a hybrid approach of inductive and deductive coding and theme development, International Journal of Qualitative Methods 5 (1) (2006) 1–11.
[39] R. Flin, S. Yule, Leadership for safety: industrial experience, Quality Safety in Health Care 13 (Suppl. 2) (2004) ii45–ii51.
[40] J.R. Galbraith, Designing Complex Organizations, first ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1973.
[41] J. Gerring, Case Study Research: Principles and Practices, Cambridge University Press, 2006.
[42] R.A. Guzzo, M.W. Dickson, Teams in organizations: recent research on performance and effectiveness, Annual Review of Psychology 47 (1) (1996) 307–338.
[43] B.H. Hamilton, J.A. Nickerson, H. Owan, Team incentives and worker heterogeneity: an empirical analysis of the impact of teams on productivity and participation, Journal of Political Economy 111 (3) (2003) 465–497.
[44] J.E. Hannay, H.C. Benestad, Perceived productivity threats in large agile development projects, in: Proceedings of the 2010 ACM–IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM '10., ACM, New York, NY, USA, 2010, pp. 1–10.
[45] J. Highsmith, Agile Project Management – Creating Innovative Products, Pearson Education, Boston, 2004.
[46] P.J. Hinds, S. Kiesler (Eds.), Distributed Work, MIT Press, Cambridge, MA, USA, 2002.
[47] R. Hoda, J. Noble, S. Marshall, Organizing self-organizing teams, Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering – ICSE '10, vol. 1, ACM, New York, NY, USA, 2010, pp. 285–294.
[48] A. Joshi, N. Pandey, G.H. Han, Bracketing team boundary spanning: an examination of task-based, team-level, and contextual antecedents, Journal of Organizational Behavior 30 (6) (2009) 731–759.
[49] K.M. Kacmar, M.C. Andrews, D.L. Van Rooy, R.C. Steilberg, S. Cerrone, Sure everyone can be replaced . . . but at what cost? Turnover as a predictor of unit-level performance, Academy of Management Journal 49 (1) (2006) 133–144.
[50] D. Karlström, P. Runeson, Integrating agile software development into stage-gate managed product development, Empirical Software Engineering 11 (June) (2006) 203–225.
[51] A. Kelly, Changing Software Development: Learning to Become Agile, Wiley Publishing, 2008.
[52] B. Kitchenham, S. Pfleeger, L. Pickard, P. Jones, D. Hoaglin, K. El Emam, J. Rosenberg, Preliminary guidelines for empirical research in software engineering, IEEE Transactions on Software Engineering 28 (8) (2002) 721–734.
[53] S.W. Kozlowski, D.R. Ilgen, Enhancing the effectiveness of work groups and teams, Psychological Science in the Public Interest 7 (3) (2006) 77–124.
[54] G. Lee, W. Xia, Toward agile: an integrated analysis of quantitative and qualitative field data, MIS Quarterly 34 (1) (2010) 87–114.
[55] D. Leffingwell, Scaling Software Agility: Best Practices for Large Enterprises. The Agile Software Development Series, Addison-Wesley Professional, 2007.
[56] J.M. Levine, H.S. Choi, Impact of personnel turnover on team performance and cognition. Team Cognition: Understanding the Factors that Drive Process and Performance, American Psychological Association, Washington, DC, 2004. pp. 153–176.
[57] J.M. Levine, R.L. Moreland, Progress in small group research, Annual Review of Psychology 41 (1) (1990) 585–634.
[58] S. Licorish, A. Philpott, S.G. MacDonell, Supporting agile team composition: a prototype tool for identifying personality (in)compatibilities, in: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. CHASE '09., IEEE Computer Society, Washington, DC, USA, 2009, pp. 66–73.
[59] Y.S. Lincoln, E.G. Guba, Naturalistic Inquiry, Sage Publications, Newbury Park, CA, 1985.
[60] Y. Lu, C. Xiang, B. Wang, X. Wang, What affects information systems development team performance? An exploratory study from the perspective of combined socio-technical theory and coordination theory, Computers in Human Behavior 27 (2) (2011) 811–822.
[61] A. MacCormack, R. Verganti, M. Iansiti, Developing products on internet time: the anatomy of a flexible development process, Management Science 47 (January) (2001) 133–150.
[62] T. Malone, K. Crowston, J. Lee, B. Pentland, Tools for inventing organizations: toward a handbook of organizational processes, in: Proceedings of the Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1993, pp. 72–82.
[63] T.W. Malone, K. Crowston, The interdisciplinary study of coordination, ACM Computing Surveys 26 (March) (1994) 87–119.
[64] M.A. Marks, J.E. Mathieu, S.J. Zaccaro, A temporally based framework and taxonomy of team processes, Academy of Management Review 26 (3) (2001) 356–376.
[65] J.E. Mathieu, M.T. Maynard, T. Rapp, L. Gilson, Team effectiveness 1997–2007: a review of recent advancements and a glimpse into the future, Journal of Management 34 (3) (2008) 410–476.
[66] K.D. Maxwell, P. Forselius, Benchmarking software development productivity, IEEE Software 17 (1) (2000) 80–88.
[67] J. McAvoy, T. Butler, The impact of the Abilene Paradox on double-loop learning in an agile team, Information and Software Technology 49 (June) (2007) 552–563.
[68] O. McHugh, K. Conboy, M. Lang, Using agile practices to influence motivation within it project teams, Scandinavian Journal of Information Systems 23 (2) (2011) 85–110 (Special Issue on IT Project Management).
[69] C. Melo, D.S. Cruzes, F. Kon, R. Conradi, Agile team perceptions of productivity factors, in: Proceedings of the Agile Conference 2011 (AGILE'11), IEEE Computer Society, Salt Lake City, UT, USA, 2011, pp. 57–66.
[70] M.B. Miles, A.M. Huberman, Qualitative Data Analysis: An Expanded Sourcebook, vol. 2nd, Sage, 1994.
[71] D. Mishra, A. Mishra, Effective communication, collaboration, and coordination in extreme programming: human-centric perspective in a small organization, Human Factors and Ergonomics in Manufacturing and Service Industries 19 (5) (2009) 438–456.
[72] N.B. Moe, T. Dingsyr, T. Dybå, Understanding self-organizing teams in agile software development, in: Australian Software Engineering Conference (ASWEC), IEEE Computer Society, 2008, pp. 76–85.
[73] N.B. Moe, T. Dingsoyr, T. Dyba, Overcoming barriers to self-management in software teams, IEEE Software 26 (6) (2009) 20–26.
[74] N.B. Moe, T. Dingsoyr, T. Dyba, A teamwork model for understanding an agile team: a case study of a scrum project, Information and Software Technology 52 (5) (2010) 480–491.
[75] P. Mohagheghi, R. Conradi, Quality, productivity and economic benefits of software reuse: a review of industrial studies, Empirical Software Engineering 12 (October) (2007) 471–516.
[76] M. Mulder, Case studies in performance improvement, Advances in Developing Human Resources 1 (1) (1999) 83–94.
[77] S. Nerur, V. Balijepally, Theoretical reflections on agile development methodologies, Communications of the ACM 50 (March) (2007) 79–83.
[78] NVivo, QSR International, 2010. <http://www.qsrinternational.com/productsnvivo.aspx>.
[79] N. Parolia, S. Goodman, Y. Li, J.J. Jiang, Mediators between coordination and is project performance, Information and Management 44 (October) (2007) 635–645.
[80] M.Q. Patton, Enhancing the quality and credibility of qualitative analysis, Health services research 34 (5 Pt 2) (1999) 1189–1208.
[81] K. Petersen, Measuring and predicting software productivity: a systematic map and review, Information and Software Technology 53 (4) (2011) 317–343. special section: Software Engineering track of the 24th Annual Symposium on Applied Computing – Software Engineering track of the 24th Annual Symposium on Applied Computing.
[82] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, J. Still, The impact of agile practices on communication in software development, Empirical Software Engineering 13 (June) (2008) 303–337.
[83] J.D. Procaccino, J.M. Verner, K.M. Shelfer, D. Gefen, What do software practitioners really think about project success: an exploratory study, Journal of Systems and Software 78 (November) (2005) 194–203.

[84] F. Rafii, How important is physical collocation to product development success?, Business Horizons 38 (1) (1995) 78–84
[85] Y.W. Ramírez, D.A. Nembhard, Measuring knowledge worker productivity: a taxonomy, Journal of Intellectual Capital 5 (4) (2004) 602–628.
[86] R.H. Rasch, H.L. Tosi, Factors affecting software developers' performance: an integrated approach, MIS Quarterly 16 (1992) 395–413.
[87] E. Salas, Is there a big five in teamwork?, Small Group Research 36 (5) (2005) 555–599
[88] E. Salas, D.E. Sims, C. Klein, Cooperation and Teamwork at Work. Encyclopedia of applied psychology, vol. 1, Academic Press., San Diego, 2004. pp. 497–505 ().
[89] K. Schwaber, M. Beedle, Agile Software Development with Scrum, first ed., Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
[90] H. Sharp, H. Robinson, An ethnographic study of XP practice, Empirical Software Engineering 9 (December) (2004) 353–375.
[91] H. Sharp, H. Robinson, Three Cs of agile practice: collaboration, coordination and communication, in: T. Dingsyr, T. Dybå, N.B. Moe (Eds.), Agile Software Development: Current Research and Future Directions, Springer, Berlin, 2010, pp. 61–85.
[92] F. Shull, G. Melnik, B. Turhan, L. Layman, M. Diep, H. Erdogmus, What do we know about test-driven development?, IEEE Software 27 (November) (2010) 16–19
[93] K.J. Stewart, S. Gosain, The moderating role of development stage in free/open source software project performance, Software Process: Improvement and Practice 11 (2) (2006) 177–191.
[94] D.E. Strode, B. Hope, S.L. Huff, S. Link, Coordination effectiveness in an agile software development context, in: Proceedings of the 15th Pacific Asia Conference on Information Systems (PACIS), 2011, pp. 1–16.
[95] G.P. Sudhakar, A. Farooq, S. Patnaik, Soft factors affecting the performance of software development teams, Team Performance Management 17 (2011) 187–205.
[96] T. Tan, Q. Li, B. Boehm, Y. Yang, M. He, R. Moazeni, Productivity trends in incremental and iterative software development, in: Proceedings of 3rd International Symposium on Empirical Software Engineering and Measurement. (ESEM '09), IEEE Computer Society, Washington, DC, USA, 2009, pp. 1–10.
[97] X. Tang, R. Kishore, The antecedents and consequences of agile practices: a multi-period empirical study of software teams in time-bound projects, in: Proceedings of the International Conference on Information Systems (ICIS) and International Research Workshop on IT Project Management, Saint Louis, Missouri, USA, 2010, pp. 142–157.
[98] S. Tangen, Demystifying productivity and performance, International Journal of Productivity and Performance Management 54 (1) (2005) 34–46.
[99] S. Teasley, L. Covi, M.S. Krishnan, J.S. Olson, How does radical collocation help a team succeed?, in: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work CSCW'00, ACM, New York, NY, USA, 2000, pp. 339–346.
[100] A. Trendowicz, J. Münch, Factors influencing software development productivity – state-of-the-art and industrial experiences, Advances in Computers 77 (2009) 185–241.
[101] B.W. Tuckman, Developmental sequence in small groups, Psychological Bulletin 63 (6) (1965) 384–399.
[102] A. Tziner, A. Birati, Assessing employee turnover costs: a revised approach, Human Resource Management Review 6 (2) (1996) 113–122.
[103] G.S. van der Vegt, S. Bunderson, B. Kuipers, Why turnover matters in self-managing work teams: learning, social integration, and task flexibility, Journal of Management 36 (5) (2010) 1168–1191.
[104] J. Verner, J. Sampson, V. Tosic, N. Bakar, B. Kitchenham, Guidelines for industrially-based multiple case studies in software engineering, in: Third International Conference on Research Challenges in Information Science, 2009. RCIS 2009, 2009, pp. 313–324.
[105] R. Wageman, How leaders foster self-managing team effectiveness: design choices versus hands-on coaching, Organization Science 12 (September) (2001) 559–577.
[106] S. Wagner, M. Ruhe, A Structured Review of Productivity Factors in Software Development, Tech. Rep. TUM-I0832, Institut für Informatik – Technische Universität München, 2008.
[107] L. Wallace, M. Keil, A. Rai, Understanding software project risk: a cluster analysis, Information and Management 42 (2004) 115–125.
[108] E. Whitworth, R. Biddle, Motivation and cohesion in agile teams, in: G. Concas, E. Damiani, M. Scotto, G. Succi (Eds.), Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Computer Science, vol. 4536, Springer, Berlin/Heidelberg, 2007, pp. 62–69.
[109] D.E. Yeatts, C. Hyten, High-performing Self-managed Work Teams: A Comparison of Theory to Practice, Sage Publications, Thousand Oaks, 1998.
[110] R.K. Yin, Case Study Research: Design and Methods. Applied Social Research Methods, fourth ed., Sage Publications Inc., 2008.