Information and Software Technology 56 (2014) 79–101

Contents lists available at ScienceDirect

Information and Software Technology

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

Motivation in software engineering industrial practice: A cross-case analysis of two software organisations

A. César C. França [a,b], Fabio Q.B. da Silva [a,*], Adelnei de L.C. Felix [a], David E.S. Carneiro [a]

[a] Centre for Informatics, UFPE Cidade Universitária, 50.540-740 Recife, PE, Brazil
[b] Faculdade de Filosofia, Ciências e Letras de Caruaru, FAFICA, Rua Azevedo Coutinho, S/N – Petrópolis, Caruaru, PE, Brazil

ARTICLE INFO

Article history:
Available online 24 July 2013

Keywords:
Motivation
Software engineer
Qualitative research
Multi-case replication design
Meta-ethnography
Software engineering

ABSTRACT

Context: The research about motivation in software engineering has provided important insights into characterizing factors and outcomes related to motivation. However, the complex relationships among these factors, including the moderating and mediating effects of organisational and individual characteristics, still require deeper explanatory investigation.
Objective: Our general goal is to build explanatory theories of motivation in different software organisations and to integrate these local theories towards a comprehensive understanding of the role of motivation in the effectiveness of the individuals and the teams in which they work. In this article, we describe the integrative synthesis of the results of two case studies performed with software organisations in different business contexts.
Method: We performed two case studies using a multiple-case, replication design, focusing on the software engineers as the unit of analysis. For 13 months, we conducted semi structured interviews, diary studies, and document analyses, and analysed the collected data using grounded theory procedures. The results of the two cases were synthesized using a meta-ethnography supported process.
Results: We built translations of the concepts and propositions from the two studies into one another. We then used the translations to build a central story of motivation that synthesizes the individual stories. This synthesis is contextualized by the differences in organisational and individual characteristics.
Conclusion: The differences in organisational contexts and in the characteristics of the software engineers in each study provided rich explanations for contrasts in perceptions and feelings about motivation in both organisations. The theory that emerged from the synthesis, supported by these explanations, provides a deeper understanding of the interplay between motivators and outcomes, and the needs and personal goals of the software engineers. This theory also characterises the role of team cohesion in motivation, advancing previous models about motivation in software engineering.

1. Introduction

We are interested in understanding why some software engineers seem to apply higher levels of effort in doing their tasks than others, since the level of effort is closely tied to the performance outcomes of the task [1]. We are also interested in explaining why different levels of effort are found among employees that work in the same organisational context and perform similar tasks. In other words, we are interested in understanding and explaining the varying contextual factors and individual characteristics that affect the amount of effort a software engineer is willing to apply directly in performing his or her work in the practice of software development in industry.

Individual effort, i.e., the amount of human energy expended directly on doing some task, is closely related to motivation and commitment [1]. According to Hackman [1], when organisational context and the design of the tasks are motivating, individuals feel more satisfied and, thus, more willing to apply higher levels of effort to the job. This, in turn, increases performance and decreases absenteeism, loafing, and turnover. Therefore, understanding motivation of software engineers has important implications for industrial practice.

However, the study of motivation is complex, since motivation seems to be affected by a diverse set of environmental conditions [1]. Further, the effect of motivating factors on individuals are likely to be moderated by personality and individual values, beliefs, and needs [3]. Therefore, contextualized and explanatory theories of motivation are necessary to account for the diversity of factors and contextual conditions that may affect motivation. Further, the integrative synthesis of these local theories can contribute

[*] Corresponding author. Tel.: +55 81 85944085; fax: +55 81 34297921.
E-mail address: fabio@cin.ufpe.br (F.Q.B. da Silva).

0950-5849/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2013.06.006

to a deeper understanding of motivation in software practice and help practitioners and researchers to define and apply motivational strategies aiming at increasing commitment and effectiveness of software engineers in software organisations.

The research about motivation in software engineering has been developed for more than 30 years. However, the vast majority of the previous studies followed a quantitative approach based on survey research as the main research methodology [4,5]. Few qualitative studies were found in the literature and none of these studies provided a synthesis of results found in different industrial contexts.

More recently, based on a comprehensive systematic literature review initially presented by Beecham et al. [4], Sharp et al. [6] reviewed the conceptual models of motivation found in the literature and proposed a new and general model of software engineers' motivation called MOCC (acronym for *Motivators*, *Outcomes*, *Characteristics* and *Context*). Although the MOCC was built from empirical evidence found in the scientific literature, the model itself has not been empirically tested so far. Hall et al. [7] presented a review of the use of classical theories of motivation in software engineering research. Finally, Hall et al. [8] presented a summary of the main findings of the literature reviews, describing and categorizing factors that affect motivation and how they could be managed in practice to improve job outcomes. We describe the main findings of this set of articles [4,6–8] in Section 2.4 and compare our findings with their results in Section 5.1.

This body of research has compiled evidence about several factors related to motivation in software development and the outcomes that are usually related to motivated behaviour in general. Models like MOCC [6] and its summary presented by Hall et al. [7] increased our understanding of the complex phenomenon of motivation and provided strong foundations for further research. However, although they emphasize the role of environmental and individual characteristics as moderators and mediators of the effect of motivators in general terms, these models do not explicitly explain differences in these moderating and mediating effects in specific organisational contexts. The purpose of our research is to complement these previous studies with a deeper understanding of how differences in environmental conditions lead to differences in the relationships between motivators and the perceived outcomes of motivated behaviour. More specifically, our goal is to find answers to the following research questions:

> *How do organisational, individual, and task related factors affect the motivation of software engineers in the workplace, and what are the perceived outcomes of motivated behaviour?*

Our research was built on the main findings of the work presented by Sharp et al. [6] and the MOCC model was used in some of our initial studies [9]. With the goal of building rich descriptive and context sensitive accounts of the phenomenon of motivation, our present research strategy involved the development of multiple, replicated case studies in different industrial contexts. Consistent with the constructivist philosophical stance, we used interpretive and qualitative research methods. Therefore, in each case study we built contextualized theories from the data, using grounded theory methods. Then, we synthesized the results of studies using meta-ethnography supported procedures.

In this article, we present the integrative synthesis of two case studies that have been individually reported [10,11]. In these studies, we investigated motivation in the context of a large and mature public organisation [10,12] and of a small and young private company [11,13]. Both organisations are located in the city of Recife, in Brazil, and thus share the same business, social, and educational environment.

We summarize the main findings of each study and then build a cross-case analysis that extends these findings in two complementary ways. First, it provides confirmatory evidence for several factors and relationships that are present in both studies. Second, for the contradictory findings, the synthesis provides explanations based on demographic and contextual information related to each specific study. In both ways, we believe that this synthesis increases the reliability of the individual studies and provides a more comprehensive view of the phenomenon. In this sense, each study preserved its integrity as a whole piece of knowledge and their synthesis expanded this knowledge by explaining the similarities and divergences between the two studies. We also compare our findings with the MOCC model presented by Sharp et al. [6] in Section 5.1 and in specific points throughout the text. Based on two recent systematic reviews already cited above [4,5], this is the first study that attempts to synthesize two qualitative case studies about motivation in software engineering using meta-ethnography procedures.

This article is structured as follows. In Section 2, we present a conceptual framework for motivation and a discussion of related work. In Section 3, we describe the research design and specific aspects of the research methods used for the individual case studies and their synthesis. In Section 4, we describe the results of the synthesis of the two studies. We discuss the threats to validity and the implications of these results to research and practice in Section 5. Finally, in Section 6, we present the conclusions and some directions for future work.

2. Conceptual background and related work

2.1. Human motivation

The first attempts to scientifically understand and explain motivation came from the fields of psychotherapy, psychometrics, and learning theories in the early 1900s [14]. After the 1930s, the School of Human Relations disseminated the study and application of motivational strategies in business management. Latham and Ernst [15] pictured the history of workplace motivation as a set of clear distinct evolutionary stages. In its initial stages, there were only isolated initiatives in studying human motivation, most of which were disconnected and lacking deeper theoretical foundation. By the 1960s, psychologists became aware of "the importance of (1) taking into account a person's needs (...), (2) creating a job environment that is likely to facilitate self-motivation (...), and (3) ways to directly modify, that is, to directly increase or decrease another person's behaviour by administering environmental reinforcers and punishers contingent upon a person's response (...)" [15].

In the 1990s, other motivational principles were added to those articulated earlier: (4) setting specific high goals that are judged by employees to be attainable (concept closely related to self-efficacy [16]), (5) and ensuring that the attainment is tied to rewards that are valued and appraised as equitable by the employee. Closer to the end of the 20th century four new principles were added to the pre-existing list discussed above: (6) "Understand the outcomes people expect and you will understand their behavior; change the outcomes people expect, and you will change their behavior"; (7) "Increase a person's or group's efficacy, namely the conviction of 'can do' versus 'can't', through enactive mastery, modelling, persuasion by a significant other, or one or more combinations of these three methods"; (8) "Take the steps necessary for ensuring feelings/perceptions of procedural and interactive justice among the workforce" [15].

These principles for understanding human motivation have been derived from a large amount of theoretical and empirical research developed during this range of time and, according to Latham and Ernst [15], they have proved to be stable and timeless. Nevertheless, the definition of motivation is still polemic and the choices among competing theories have stimulated researchers and practitioners in a long lasting debate (see Katzell and Thompson [17] for a review). It is out of the scope of this article to provide definitions and deeper discussions about different types of motivation theories. However, it is important that we base our research on a consistent set of concepts, as we will discuss below.

Despite the diversity and complexity of the different theories and definitions of motivation, they seem to converge to a common set of characteristics. First, as discussed by Rogers, Lundington, and Graham, motivation is an internal feeling, which pushes someone towards doing something [18]. Being an *internal feeling* poses a challenge to operational definitions, leading to the need to use proxies to understand the possible external signs of motivated individuals, and the external factors that influence internal motivation. Models usually try to explain motivation by showing a causal relationship between these factors and outcomes [6]. Further, direct observation of motivation is not feasible, only the observation of behaviour attributed to motivation.

Steers, Mowday, and Shapiro discuss the idea that theories and definitions of motivation "… are all principally concerned with factors or events that energize, channel, and sustain human behaviour over time" [19]. Motivation has been understood sometimes as a psychological factor or set of factors, and sometimes as a process through which those factors lead to choose, instigate, and start an objective-oriented behaviour. The distinction between factors and processes leads to the classification of theories of motivation into at least two groups. The Content theories focus on the factors that satisfy human needs and the Process theories are mainly interested in the dynamic processes through which motivation occurs. Important instances of Content theories are Maslow's Hierarchy of Needs Theory [20], Herzberg's Motivation-Hygiene Theory [21], and McClelland's Needs Theory [22]. Skinner's Stimulus-Response Theory [23], Vroom's Expectancy Theory [24], Locke's Goal-setting Theory [25], and Hackman and Oldham's Job Characteristics Theory [2] are classic examples of Process theories.

It is the investigation of motivational factors, their inter-relationships, and their effects on the outcomes of the software engineer in the work practice that guide our research.

Some of the classical theories and models of motivation group these factors into many different categories, such as by their level of priority [20], their contributory or inhibitory effects in motivation [21], or their intrinsic or extrinsic characteristics [26]. These classifications are not straightforward and there is little agreement in general about how to use these classifications [26]. In fact, despite their theoretical importance, these definitions have limited impact on managerial practices. We, therefore, will not attempt to categorise the factors identified in our studies in these categories, leaving this problem to future research. Consistent with our constructive stance and the use of grounded theory, the categories will be constructed from the data collected from the field.

2.2. Research about Motivation in Software Engineering

In software engineering, motivation has been studied since the 1980's. A seminal work of Couger and Zawacki [27] claimed that computing personnel have higher growth needs and lower social needs than the average population, implying that specific strategies of job design [2] could be employed for this specific type of professional, in order to increase their job's motivational potential and consequently their performance at work.

Two recent systematic reviews analysed 140 scientific papers, published between 1980 and June of 2010 [4,5]. This body of literature has addressed motivation in different types of organisations and from different theoretical perspectives, and the main contributions are the consolidation of a general set of factors that attempt to predict the motivation of the software engineer. This body of research has been dominated by quantitative studies (90/140) and the vast majority of the quantitative studies (85/90) used survey research as the main research methodology. Most of the results of these surveys point to the software engineering job itself as the main motivational factor, while performance improvements (e.g. product quality, productivity, and job excitement) and intention to leave the organisation are highlighted as the most important outcomes of, respectively, high and low motivation.

However, these literature reviews also identified three important gaps in the research that must be addressed:

- Although there are many existing theories of motivation, only some of these studies (71%; 100/140) was appropriately underpinned by classical motivation theories. Further, these studies adopt different theories, which may "overlap, address issues at different levels of abstraction, and occasionally contradict each other" [7], thus making it difficult to compare and integrate findings from different studies.
- Some researchers claim that software engineers differ from the general population with respect to personality, needs, and other individual characteristics, and for this reason, what motivates software engineers is likely to be different from what motivates the population in general. However, the difficulties in verifying these claims and the lack of understanding of exactly how these differences (if they exist) affect motivation make these claims not very useful in practice.
- Most of the results of the surveys about motivation identify the software engineering job itself as the main motivator. The core limitation of these results is that they do not provide explanations on what it is about the job that motivates software engineers and how this effect happens.

Theoretical and empirical research from different areas indicates that motivation is context dependent and varies from one individual to another. Therefore, to address the gaps identified above, it is necessary to produce theories capable of explaining why some software engineers seem to be motivated to perform their work while others do not, in particular regarding the effects of organisational and personal factors. Moreover, it is important to identify what aspects of software engineering are motivating and how these aspects affect the motivation of the individual in the working place.

2.3. Related case studies

Among the 140 studies mentioned in Beecham et al. [4] and França et al. [5], it is possible to identity only a few cases studies that provide a deep analysis of specific industrial cases, using qualitative interpretive approaches focused in the motivation of software engineers. In this section, we briefly describe these case studies.

Beecham et al. [28], based on the general research claim that agile methods lead to higher motivation, carried out a case study with five mature XP teams and concluded that agile practices are somewhat supportive to the motivation of software engineers regarding peer relationships. However, the XP environment did not support individual needs of recognition, clear career progression, and variety of work. Our results are consistent with these findings in traditional (non-agile) contexts, reinforcing the generalizability of the results.

Tessem and Maurer [29] carried out semi-structured interviews with a large industrial SCRUM team (70 people), aimed at understanding how the size of the development team could possibly affect the presence of the task design characteristics that are hypothesized to affect motivation: Autonomy, Variety, Significance, Feedback, and Task Identity [2]. Their results showed that, despite the relatively large team size, the essential values and practices of agile teams were present, such as small sub teams, open communication, participation in decisions, and autonomy in task assignment. The existence of these practices supported the perception that motivating task design characteristics existed. As a result, as predicted in the Job Characteristics theory, job satisfaction and motivation also appeared to have been maintained among the team, which was signalled by the willingness to work hard to complete their tasks within the defined sprints, and low turnover.

More recently, Sach et al. [30] interviewed 13 software engineering practitioners from an agile software company, in situ, regarding their motivation to work. Doing "work that is useful", "producing good software" and "solving problems" emerged as the most cited motivating factors. In our current study, task design characteristics and factors related to the problem solving content of the tasks were also found to be important.

In summary, these studies have provided evidence that motivation is highly directed by task related characteristics as well as by organisational characteristics and managerial practices. Our results support these findings and extend them with a deeper understanding about how these factors interact in different contexts.

It is important to notice that motivation has received little attention in research related to public sector organisations and motivation relevant characteristics of public sector jobs are not well documented either [31]. This is also true in software engineering and, consistently, we did not find any study reporting case studies about software organisations in the public sector. According to Perry and Porter [32], some characteristics of the public sector may influence motivation, including the notion that government is perceived primarily as a service provider rather than a goods producer, and that governmental organisations pursue diffuse and conflicting goals, since the political environment hinders goal directed activity. Our study of a public organisation confirms these influences. Reid et al. [33] surveyed American IT employees for antecedents of voluntary turnover in the public sector, and found that organisational support, leader-member exchange, role ambiguity, and task variety significantly explained the variance in employee commitment and job satisfaction.

2.4. The MOCC Model

Sharp et al. [6] proposed a new model of motivation in software engineering built from empirical findings compiled in their prior systematic literature review [4] (which is summarized by Hall et al. [8]) with the goal of clarifying the complex relationships among the factors that may affect the motivation of software engineers. This new model, called MOCC (acronym for *Motivators*, *Outcomes*, *Characteristics* and *Context*) (Fig. 1) summarizes a large amount of empirical evidence and serves as a reasonable framework to organise and guide future research on this topic because of its compatibility with several motivation theories.

In summary, the MOCC Model postulates that motivators (which are classified as intrinsic or extrinsic according to the definitions of Herzberg et al. [21]) affect certain outcomes of the job, such as job retention, productivity, project delivery time, budget, absenteeism, and general project success. Further, the model also states that this effect is moderated and mediated by individual characteristics and contextual factors. The model expresses the relationships among categories of factors and outcomes in a general way, providing a high level of abstraction that is useful for a generic understanding of the phenomenon. This understanding has provided important guidance in the design of specific studies, like the ones carried out by our research group.

However, the relationships between specific factors and their direct effect on a given outcome are not explicitly modeled in the MOCC. Furthermore, the model is a synthesis from several studies in different contexts and the specific relationships occurring in any given particular context are not expressed in the model. The application of the model in a particular situation, thus, requires careful interpretation and instantiation. Besides, as mentioned by MOCC authors, "the impact of context is more complex than initially suggested" and "in order to revise and augment the model (...) further understanding of the software engineering-specific issues that underpin motivation is needed" [6]. We believe that case studies and integrative synthesis, as presented in our current study, can contribute to close these gaps.

Therefore, in Section 5.1, we compare our findings with MOCC and other models from the literature. We discuss what particular relationships among factors were found and use contextual information to explain the discrepancies between the models and our findings.

3. Research methods

The research on motivation in software engineering has been largely dominated by quests for general results that would apply across a large number of different organisational contexts, technological contingencies, and types of individuals [4,5]. Consistently, the preferred research method used has been survey research, with emphasis on quantitative analysis. In our research, we are interested in understanding how individual software engineers interpret their experiences in the workplace, how these interpretations shape the meaning of motivation, and why certain combinations of contextual and individual factors lead to more or less motivated behaviour. Our research goals and our philosophical stance call for a qualitative and interpretive research method.

3.1. Study design

In this research, we take a constructivist or interpretive philosophical perspective that "assumes that reality is socially constructed, (...), there are multiple realities, or interpretations, of a single event" [34]. Accordingly, we use a qualitative approach. According to Merriam [34] and Denzin and Lincoln [35], "qualitative researchers study things in their natural settings, attempting to make sense of, or interpret, phenomena in terms of the meanings people bring to them."

Consistent with our intention of investigating individuals in specific working contexts, we used a qualitative case study, understood as "an in-depth description and analysis of a bounded system" [34], as the research procedure. We performed multiple independent case studies, using a multi-case replication design [37], choosing cases of contrasting nature to increase data diversity. We classify our case studies as instrumental, since our goal is to understand the construct and build theories. Multiple cases were synthesized and integrated to extend our understanding of the phenomenon in different contexts, and to explain similarities and discrepancies among individual results. We used procedures derived from meta-ethnography to synthesize the studies and to build a new theory of motivation that explains similarities and discrepancies found in both studies. Fig. 2 describes the multi-case method.

In this article, we present the results of two case studies performed in different organisations [10,11], and develop the synthesis of their results. As far as we are aware, the combination of qualitative case studies using meta-ethnography is novel in the literature about motivation in software engineering. We consider

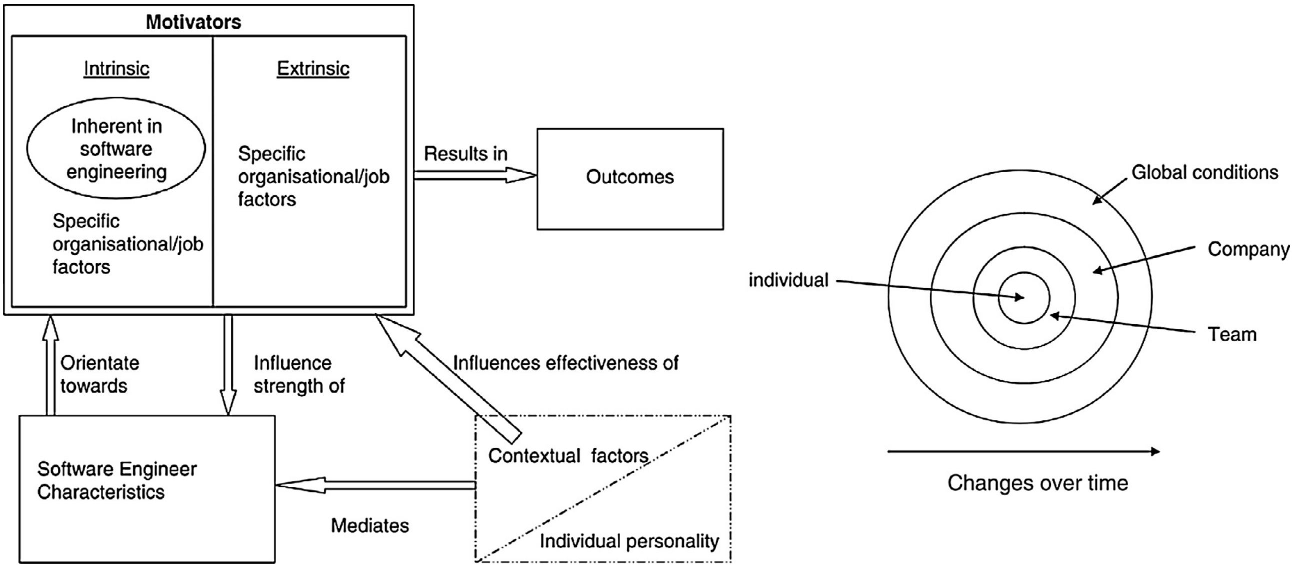

Fig. 1. The MOCC Model and its contextual layers (extracted from Sharp et al. [6]).

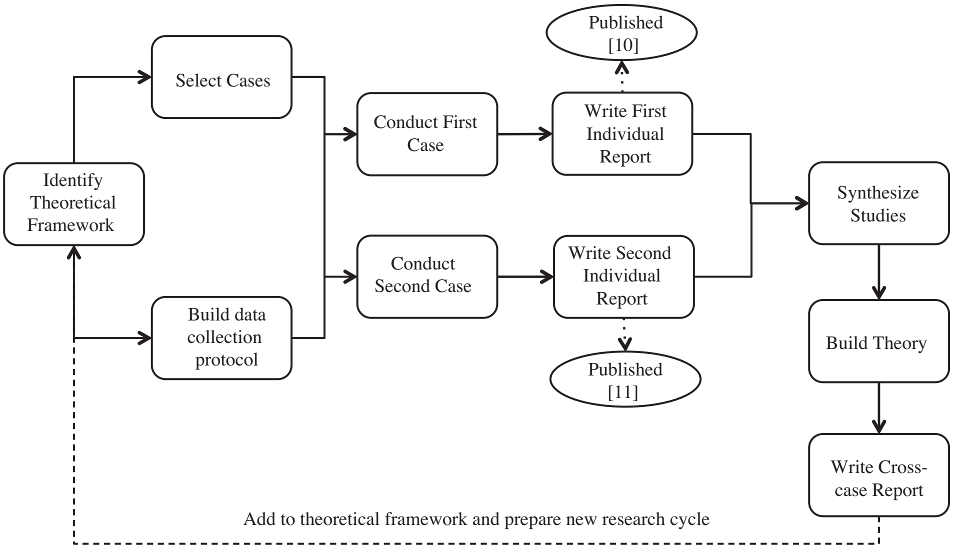
Published [10]
Select Cases
Conduct First Case
Write First Individual Report
Identify Theoretical Framework
Synthesize Studies
Conduct Second Case
Write Second Individual Report
Build data collection protocol
Build Theory
Published [11]
Write Cross-case Report
Add to theoretical framework and prepare new research cycle

Fig. 2. The multi-case design.

that this is also a contribution of this article to empirical software engineering research.

3.2. The samples in the multi-case study

Usually, two levels of sampling are necessary in qualitative case studies. The first is the selection of the cases that will be investigated or, in other words, what constitute the bounded system of interest for the research. Then, we must sample participants and other sources of data within each case.

In this research, the bounded system of interest for each case study is a software organisation. In our multi-case design, we deployed a maximum variation strategy. This sampling strategy was first identified by Glaser and Strauss [38] for whom a "grounded theory would be more conceptually dense and potentially more useful if it had been 'grounded' in widely varying instances of the phenomenon" [34]. Therefore, we chose two distinct software organisations in the same business environment (although in different business sectors) and in similar social and educational contexts. We chose organisations with different business nature (public and private), different sizes and maturity, and distinct business goals, but kept the overall context (social, cultural, economic, and educational) the same. In this way, we achieved large variation regarding the individual organisations but kept the environmental and social variables the same to facilitate comparison and integration of findings. Specifically, with respect to the size and maturity dimensions, we developed case studies on two polar opposite organisations: a mature government agency and on a small and young private software company.

Data was collected directly from participants using the techniques explained in Section 3.3. The primary type of participant was the software engineer in each organisation, because we were interested in studying motivation from a personal and subjective perspective. However, as explained below we collected data from project managers and directors of the organisation for triangulation of factual events. That is, we checked with managers and directors the information about organisational aspects and about process definition and other resources available to the software developers. We also analysed documentation to complement or triangulate the data collected from participants.

In the second level of sampling we needed to choose the software engineers in each case study. We tried to achieve a good coverage of age, background and education, years of professional experience, years of employment in the organisation, participation in different projects in the organisation, work on different activities in software development and maintenance, among other factors, to ensure a fertile sample. This sampling strategy was used since patterns that emerge based on great variation are likely to have more value in capturing the central perceptions of the phenomenon of interest, as advocated by Merriam [34], which in this case is the motivation of software engineers in a specific organisational setting.

3.3. Data collection preparation

According to Merriam [34] (p. 85), “the idea that we ‘collect’ data is a bit misleading. Data is not ‘out there’ waiting for collection.” Collecting data involves selecting data and the most suitable data collection techniques, which will ultimately affect what constitute data for the purposes of the research. Further, qualitative data collection is more dependent on the skills and experience of the researcher, than the collection of quantitative data, in general. Practice with the subject and with the techniques is essential to reduce the risks of introducing bias in the data collection process. For this reason, great care is usually taken in the preparation of the data collection process.

Qualitative data are mainly obtained through the use of one or a combination of more than one of the following techniques: interviews, observations, and document analysis. Several factors influence the choice of technique, including the nature of the phenomenon that is being studied [34]. The difficulties in observing motivation and the feelings and personal opinions about what actually constitute motivated behaviour, led to the decision to not use observation in this study. We therefore used interviews, complemented by diary studies, and document analysis for data collection.

One important issue in qualitative research is the language in which the data is collected. In our cases, the native language of the participants was Brazilian Portuguese. We, thus, conducted all data collection in this language. This is consistent with obtaining data that is richer in opinions, feelings, and emotions that is easier to express in one’s native language. However, as it will be discussed later, great care is necessary when reporting the findings in other languages, as in this article.

3.3.1. Interviews

According to Merriam [34], interviews are effective at eliciting information about things we cannot observe (such as feelings, thoughts, and intentions). We used semi-structured interviews, which is a type of interview in which a script, composed of open or less structured questions, is designed to guide the interviewer, but some freedom on the sequence of the questions and their exact wording is allowed so that the interviewer can extract as much useful and rich information as possible.

We interviewed software engineers using an interview guide composed of 43 open-ended questions (see Table 1 for a sample of the questions). The interview guide included questions aimed at exploring experience and behaviour, opinion and values, feelings, knowledge and background. We also included demographic questions at the end of the script. We also interviewed project managers and the directors of the organisation to triangulate the data collected from software engineers. As discussed before, individual motivation is not an observable phenomenon, so we used peer data to triangulate only factual information, or events, such as the characteristics of the organisation and software engineering processes. Specific interview guides were designed, one with 53 questions for the project managers and another with 12 questions formulated for the directors.

The questions in all three guides were presented in a funnel format, beginning with general questions and moving towards more specific ones [36]. All positive questions (e.g. ‘what do you like about...?’) had a corresponding negative one (e.g. ‘what do you dislike about ...?’). Each guide was pre-tested with two pilot interviews each. As a result, only minor changes were identified as necessary to improve the interview guides, such as better wording of some questions.

3.3.2. Diary studies

This is a method in which “participants are asked to record their daily activities on a pre-printed log form” [39]. It is a method of understanding participant behaviour and intent, in situ, which minimizes the effects of observers or interviewers on participants. Diary studies can be used to complement and/or to triangulate

Table 1
Interview guide excerpt (*translated to English from original in Portuguese*).

Q1. Talk about yourself, your education, professional trajectory, etc.
Q2. Why did you decide (*or what has brought you*) to work as a software engineer?
...
Q5. Why did you decide (*or what has brought you*) to work in this organisation?
Q6. What are the characteristics of this organisation that stimulates you to do a good job?
...
Q11. Imagine an extraordinarily good day, in which everything goes perfectly. Describe it in detail, from the early morning to late at night.
Q12. Among your daily activities, what are those you like more?
Q13. What is it about these activities (Q12) that make you like them the most?
...
Q22. What do you think are the strongest points of your current team?
...
Q24. Describe a situation in which you really felt part of the team.
...
Q36. What does the organisation do, or offer, to stimulate the software engineers’ motivation?
Q37. How do these actions/offers affect your particular work?
...
Q40. What could the organisation do (but currently does not do) to stimulate the software engineers’ motivation?
...
Q41. Projecting your career five years towards the future, what do you wish you were doing?
...
Q43. What does the term motivation mean to you?

data collected using other techniques. Our diary study data collection was designed to occur in three steps:

1. In the beginning of each week, participants received a blank notepad, in which they should annotate any event that affected (positively or negatively) his or her motivation at the moment the event occurred.
2. Every day, participants would fill out an on-line form, in which they would list all relevant events occurred in that day, and give detailed information about how and why the events affected their work.
3. At the end of each week during the study, the researchers carried out a retrospective interview to clarify and complement information submitted on the online form.

3.3.3. Document analysis

Documents are an important source of data for qualitative research because they are usually produced for reasons other than the research and therefore are not subject to the same limitations and biases. Further, the analysis of documents does not intrude on the daily activities of participants in the same way as interviews, diaries, and observations do [34]. In this study, we mined documents related to human resources and norms that regulate employee-organisation relationships, among others. The main purpose was to perform triangulation, to verify and corroborate evidence raised in interviews and diary studies.

3.4. Data collection process

Potential participants were initially contacted by e-mail, and invited to participate. The interviews were scheduled and conducted individually, on the organisation's premises. All interview sessions were audio recorded with the consent of the participants. Participants in the diary studies were randomly selected from those that participated in the interviews and also sampled from other engineers not involved in the interviews. We followed the suggestions given by Dearman et al. [40] to avoid low participation in the diary studies. Because the diary study required an additional effort from the participants, they had to be constantly stimulated and receive feedback to continue contributing. In fact, six participants were selected to participate in the diary studies in Case 2, but three of them decided to resign from this task. Further retrospective interviews pointed to their high workload as the main cause of low participation.

Tables 2 and 3 summarize the amount of data collected in both cases. In Table 11, we present demographic information describing the software engineers that participated in the data collection.

3.5. Data analysis in each case study

The objective of qualitative analysis is to systematically consolidate, reduce, and interpret data obtained from various sources, and make sense of them. According to Merriam [34], in qualitative research, data collection, analysis, and reporting often occur simultaneously, in incremental and iterative steps that are adapted as the research proceeds and results emerge. In this study, several iterations of data collection and analysis were performed, using the techniques described in this and the previous sections.

We used the methods and techniques of grounded theory [41] to code, categorize, and synthesize data, towards the construction of a theory of motivation in each case study. Initially, all audio from the interviews was transcribed verbatim. We used QSR NVivo 8 to support the data analysis and synthesis. Data analysis began with open coding of the transcripts. Post-formed codes were constructed as the coding progressed and were attached to particular pieces of the text. Then, the codes arising from each interview (and diary study of the same participant, when applicable) were constantly compared to codes in the same interview and from other interviews. From the constant comparisons of the codes, we grouped codes into categories that represent factors affecting motivation and the outcomes of high and low motivated behaviour. Fig. 3 shows an example of the category building process.

Table 2
Data collected in Case 1.

Interviews	Participants	Duration of the interviews
Between August and December 2010		
	Director #1	0:15:32
	Director #2	0:23:56
	Director #3	0:21:14
	Director #4	0:32:38
	Director #5	0:47:42
	Project Manager #1	0:34:08
	Project Manager #2	0:32:31
	Project Manager #3	1:00:47
	Sw. Engineer #1	0:33:50
	Sw. Engineer #2	1:07:31
	Sw. Engineer #3	0:28:13
	Sw. Engineer #4	1:03:06
	Sw. Engineer #5	0:32:41
	Sw. Engineer #6	1:12:23
	Total	9 h 26 min
Diary studies	Participants	# of events
Between February and March 2011		
	Software Engineer #3	12
	Software Engineer #7	42
	Software Engineer #8	18
	Software Engineer #9	14
	Total	86

Table 3
Data collected in Case 2.

Interviews	Participants	Duration of the interviews
During May 2011		
	Director #1	00:07:06
	Director #2	00:10:01
	Project Manager #1	00:33:51
	Project Manager #2	00:47:19
	Project Manager #3	00:38:13
	Sw. Engineer #1	00:38:51
	Sw. Engineer #2	00:44:19
	Sw. Engineer #3	00:33:16
	Sw. Engineer #4	00:25:51
	Sw. Engineer #5	01:16:06
	Sw. Engineer #6	00:25:29
	Sw. Engineer #7	00:58:23
	Sw. Engineer #8	00:26:40
	Sw. Engineer #9	00:30:16
	Sw. Engineer #10	00:41:25
	Total	8 h 57 min
Diary Studies	Participants	# of events
Between May and June 2011		
	Sw. Engineer #2	4
	Sw. Engineer #4	2
	Sw. Engineer #11	4
	Total	10

As the process of data analysis progressed, relationships among categories were built, leading to explanatory propositions. Finally, core categories were chosen according to their general explanatory power, propositions emerged and a narrative was created to describe the central story of the case. The motivators, the propositions, and the central story constitute the elements of a grounded theory for each case study.

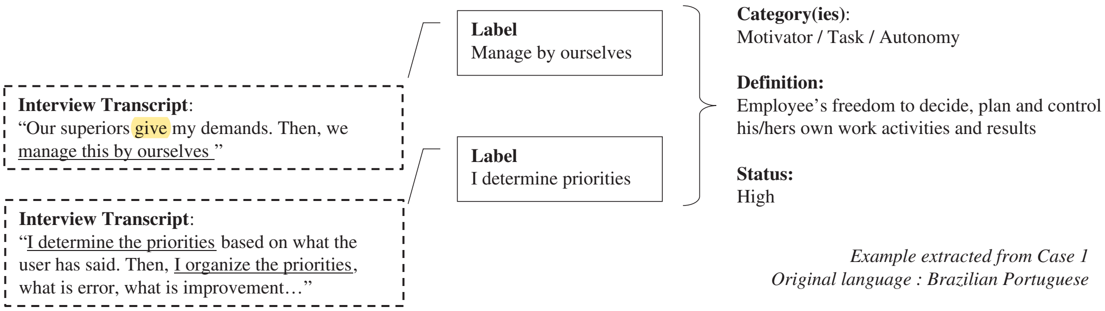
Interview Transcript:
"Our superiors give my demands. Then, we manage this by ourselves "
Interview Transcript:
"I determine the priorities based on what the user has said. Then, I organize the priorities, what is error, what is improvement…"
Label
Manage by ourselves
Label
I determine priorities
Category(ies):
Motivator / Task / Autonomy
Definition:
Employee's freedom to decide, plan and control his/hers own work activities and results
Status:
High
Example extracted from Case 1
Original language : Brazilian Portuguese

Fig. 3. Open coding category building example.

3.6. Cross-case analysis and synthesis

Research synthesis is a term used to describe a family of methods for summarizing, integrating, combining, and comparing the results of different studies which are mainly interested in similar or related research questions or topics [42]. There is a debate among qualitative researchers on whether qualitative synthesis is appropriate or even feasible, without subverting the very principles of qualitative inquiry [43]. Our position on this debate follows the view that the findings of qualitative research that uses multiple case studies are likely to be stronger than those of only one case as long as the research synthesis is properly and carefully conducted [37,43].

We used the principles of meta-ethnography in our cross-case analysis and syntheses, which is an approach that "enables a rigorous procedure for deriving substantive interpretations about any set of constructive studies" [44]. A meta-ethnography is a systematic comparison of studies, intended to be an interpretive approach to research synthesis. It involves some degree of induction and interpretation, which is consistent with the nature of the qualitative case studies we want to analyse and synthesize. Besides, instead of summarizing data for the purpose of achieving generalizations, its primary concern is "to achieve synthesis through subsuming the concepts identified in the primary studies into a higher-order theoretical structure" [42]. This is also consistent with our goal of building theories of motivation that explain similarities and contradictions among different studies.

Noblit and Hare [44] suggest the following seven steps for conducting a meta-ethnography:

1. Getting started.
2. Deciding what is relevant to the initial interest.
3. Reading the studies.
4. Determining how the studies are related.
5. Translating the studies into one another.
6. Synthesizing translations.
7. Expressing the synthesis.

As explained by Britten et al. [45], the primary interest of the founders of meta-ethnography was to work with metaphors and meta-ethnography was conceived as a way of understanding and transferring such metaphors across studies. However, the method is also applicable to studies that are not ethnographies. In our case, we use the procedures of meta-ethnography to translate and synthesize concepts (motivators and outcomes of motivated behaviour) and their relationships in the form of theoretical propositions. Translation,[1] in meta-ethnography, involves treating the cases as analogies. According to Noblit and Hare [44], translations are adequate to "maintain the central metaphors and/or concepts of each account *in their relation to other key metaphors or concepts* in that account".

In our cross-case analysis, steps 1 and 2 were conducted as part of the design and development of each individual case study, as explained before. The remaining steps are described below.

Step 3 involved carefully reading each study to identify and fully understand the main concepts and their relationships in each individual study. At this step we recorded context information about the organisations and participants that would be important to contextualize the individual results during the translations and the synthesis. The published articles and the technical reports of the studies were the source of data in this step. Results of this step are presented in Section 4.1.

In step 4, we considered the relationships between the concepts, propositions, and the central stories across studies. We interpreted and summarized the core similarities and discrepancies between the two case studies, and the results are presented in Section 4.2.

In step 5, we were interested in translating concepts and propositions of the two studies into one another. We started with the translations of concepts, that is, motivators and the outcomes of motivated behaviour. We present the results of the translations of concepts in Section 4.3.1. We then translated the propositions that were part of the grounded theory in each case study, and described the results in Section 4.3.2.

In step 6, we construct a new central story of motivation that synthesizes the two case studies. We used the differences in the context of each study to achieve a richer explanation of the relationships among motivators and outcomes. In the process, we tried to reach an integrative account of the phenomenon, while preserving the individual meanings from each study as much as possible. The level of interpretation and induction in this step is higher than in the translations and it cannot be explained as a sequence of well-defined steps. In fact, the construction of the synthesis evolved through several iterations of interpretation and induction using the translations and the results presented in each case study, as one would perform in a primary study during qualitative synthesis [34].

This paper is the result of step 7, in which we attempt to express the synthesis in a way that it is relevant for both academics and practitioners.

3.7. Addressing validity and reliability

Being able to trust results from research is important for the adoption and dissemination of the results both in academia and in the industrial practice. However, validity and reliability assessments used in causal or correlational studies do not apply directly

[1] To avoid ambiguities, we used the term 'translation' to refer to the meta-ethnography process of translating the studies into one another and the term 'idiom translation' when we needed to refer to the translation of the terms and transcriptions from Portuguese to English language.

in interpretive, qualitative research. We address below the validity and reliability of our results from the three perspectives proposed by Merriam [34]:

- *Credibility or internal validity*: Internal validity is related to the issue of whether the research results consistently represent reality. Thus, the central problem is how to provide evidence that the findings are credible and represent reality as faithfully as possible, as the data is presented. To increase credibility, we used triangulation by having data collected from participants with different roles for factual data and by using multiple data collection techniques when collecting personal data. We then used member checking, *also known* as respondent validation, to avoid misinterpretations of what participants said. A long engagement time in data collection and analysis allowed the identification of contradicting evidence and complementary explanations. Finally, the entire study was constantly scrutinized in follow up meetings.
- *Consistency or reliability*: An important question in qualitative research is whether the findings are consistent with the data collected. To increase consistency, we used triangulation in data collection and analysis. We also kept research diaries and process logs that can be used as audit trails by external reviewers.
- *Transferability or external validity*: It is a common understanding in qualitative research that it is the reader or user of the study that should primarily engage in the generalization of research findings [34]. In this sense, the reader or user can decide to what extent the findings can be applied to other situations. The researcher has to enhance the possibility of someone else "transferring" the results. Two strategies were employed to enhance transferability. First, we tried to provide rich description of the research method, the context in which the research was performed, and the results themselves (with the side effect of increasing the length of this paper). Second, we sampled the participants to achieve maximum variation since this would provide richer data and, consequently, a richer theory, as advocated by Glaser and Strauss [38].

3.8. Ethics

According to Merriam [34], "the validity and reliability of a study depend upon the ethics of the investigator", because understanding how the investigator addressed ethical issues helps to increase his or her credibility.

In this case study, we followed the ethical norms of Resolution 196/96 of the Brazilian National Health Council. The organisations signed a Terms of Authorization and Commitment to the Research, which granted the researchers access to facilities, to the participants, and to necessary documents. It also authorized the participants to use work hours for the interviews and diary studies. The participants perceived these approved terms as indicating that the organisation was involved and interested in the research. We believe that it facilitated data collection and prevented participants from concealing information that they might think was sensitive for the organisation.

Each participant signed an Informed Consent Form that explained the overall objective and importance of the research, guaranteed confidentiality of the data provided, the anonymity of the participation, and the right to withdraw from the research at any moment.

3.9. Reporting the individual case-studies and their synthesis

Reporting qualitative studies has an important impact on credibility and transferability of the findings. Rich and thick descriptions of the research method, the context in which the research was performed, and its results are important to allow the reader or user of the research to evaluate its credibility and to understand conditions under which the findings can be transferred to another context or situation.

In the report of the individual case studies, we followed the guidelines described by Merriam [34]. In this article, we followed a different structure, consistent with the primary goal of presenting a cross-case analysis of two case studies using meta-ethnography procedures. We start the presentation of the results by describing the context and central story of each case study and a synthesis of the core distinctions between them. We then present the mutual translations of the two case studies. Finally, we explain the reciprocal and refutational translations using contextual data and build a more contextualized and robust explanation of motivation of software engineers.

One important aspect of reporting qualitative case studies and their synthesis is related to language. The studies were conducted in the native language of the researchers and the participants, which is Brazilian Portuguese. All data collection, analysis, and the individual reporting of each case study were conducted in this language. We then performed an idiom translation of the reports of the individual case studies into English. We carefully conducted these idiom translations using more than one idiom translator fluent in English and Portuguese to verify the accuracy of the process and also compared the (idiom) translated terms with their meaning in the literature about motivation. The published articles were written based on the (idiom) translated reports, eliminating the risk of introducing new idiom translation error in this stage.

The synthesis of the two cases was conducted using the English idiom translation of the two reports and the two published articles that described the case studies. Assuming that the first idiom translations of the original reports were accurate, we believe that building the synthesis based on them reduced the chances of introducing new errors into the synthesis due to language problems.

4. Results

4.1. Reading the studies

Reading the two studies resulted in the first attempt to capture similarities and discrepancies, and to find ways to explain them. We present the context and the central story of each study exactly as they appear in the publications [10,11]. In the presentation of each central story, italic names between double angled quotation marks are the factors or outcomes of motivation that emerged from coding in each case. For instance, *«task significance»* in Case 1 and *«learning opportunities»* in Case 2. The precise meaning of each concept from the studies can be found in the technical reports that complement the publications [12,13] and are discussed when we present the (meta-ethnography) translations in Section 4.3.1.

4.1.1. Case 1: A government public software organisation

4.1.1.1. Context description. This case study was carried out in a government software organisation situated in Recife, Brazil, established in 1969 by Government of the State of Pernambuco. Its core mission was to provide Information Technology services to internal customers in several levels of the State Government administration and to the citizens of the State.

As a government owned organisation, it was regulated under the laws and norms of the Brazilian public sector, which have two relevant characteristics for this study. First, since the Brazilian Constitution of 1998, public employees must be hired through an open selection process with universal access, based on objective criteria. This rules out subjective interviews, personality and behavioural assessment, peer indication, and other forms of employee selection found in the private sector. Further, it slows down the process of hiring new employees and, therefore, makes it difficult to produce a timely replacement when someone leaves the organisation. Second, all public employees have job stability after a probation period of 3 years of work in the public sector (State Law No. 6.123/68).

The organisation was structured into 14 main units distributed in different locations throughout the State. Its employees, including software engineers, were distributed in the main units and in over 60 other public administration buildings. By the time this study was performed, the organisation had 2580 employees, 272 of which composed the software engineering work force.

The organisation explicitly stimulated the adoption of open source software in the State administration, and there was one open source project under development at the time of the study. Regarding development methods and practices, the organisation used traditional, process-oriented methods, with command and control style of management in most software projects, although some small and isolated agile initiatives could also be found. Consistent with our sample strategy of maximum variation, we selected participants working in traditional process-oriented projects, agile projects, and the open source project. Software engineers worked in teams, assigned to specific development or maintenance projects. The type of task (development or maintenance) and the composition of the teams were found to be important for motivation.

4.1.1.2. The central story. «Task significance» and *«job stability»* constituted the core forces that attracted software engineers to work in the company, and acted as forces to decrease their *«intention to leave»*. Both factors were mutually reinforcing with respect to motivation. On the other hand, frustrated *«growth needs»*, caused by poor *«career development support»* and other organisational factors, were the core forces that reduced motivation, contributing to increasing the intention to leave the organisation (in particular for other public organisations, thus keeping high job stability and task significance). The attraction forces caused by the job stability and the high task significance were in constant tension with the disruptive forces caused by the engineers' frustrated growth needs. Whenever the balance was broken, the individual left the organisation. Since it was difficult to make timely replacement because of legal and bureaucratic restrictions, the overall *«workload»* increased. Then, together with *«poor management»*, the high workload reinforced the disruptive force caused by frustrated growth needs on those that stayed in the organisation. These findings are in contrast with the contention posed by Hall et al. [8] that "extrinsic nature of general good management practice means that these practices will likely have limited impact on behaviour". In our research, the extrinsic factors identified by Hall et al., such as "good management", "rewards and incentives", and "job security" played a significant role in the outcomes associated with motivated behaviour, such as voluntary turnover and job satisfaction.

These central forces were moderated by other factors. High *«team synergy»* and perceived positive *«team member's attitudes»* attenuated the negative impact of the disruptive forces, at least temporarily. Political decisions overcoming technical criteria in *«goal and priority settings»* increased the effect of frustration on intention to leave. *«Task Variety»*, being contingent on the type of task (development of new systems or maintenance of legacy systems) and moderated by individual characteristics, acted in both directions as an attraction or a disruptive force according to these contingencies.

4.1.2. Case 2: A small private software development company

4.1.2.1. Context description. The second case study was carried out in a software company, formally established in 2006 by the initiative of five entrepreneurs from the Information Technology sector field, in Recife, Brazil. As a young company, its core mission was to support the development of people and organisations with software tools, by means of technical excellence and innovation.

This company specialized in software development for different platforms, with expertise in different programming languages (such as .NET Framework, Java family, LUA, and others). It focused on the on-demand development of information systems, operating in areas such as management, finance, mining, health, and others. In addition, it also developed its own products for internal use. Its flagship product, a corporate social network, provides support for intra-organisational innovation management. At the time of the study, the company served national and international customers, mostly medium and large companies. Internal products and external projects differed significantly in terms of requirements, type of management style, and time and effort pressures. Team members from both types of projects participated in this research.

The company followed an agile-like software development process, broadly adopting practices such as regular delivery of software, adaptive management style (SCRUM based), small teams, face-to-face meetings, and customer authority. The organisational structure was flat, and the directors sometimes worked as part of the development teams. The directors themselves, who had a software engineering background, also managed all organisational issues, including the human resources.

At the time of our study, the company was composed of 27 professionals, all under 30 years old (directors included), fulfilling functions in one of three types of teams: software development, research, and design areas. Some of these professionals were in the organisation for less than 6 months, while others had more than 3 years with their teams. As an organisational strategy, the company was closely tied to the academy, both physically (its location was near a University) and operationally, and its staff was composed of undergraduate students as well as graduate students in software engineering. As part of our maximum variation policy, we sampled participants representing all these clusters.

4.1.2.2. The central story. As described above, the company was a small, young, and successful software development organisation. Given the natural limitations of resources faced by organisations in this stage of business maturity, this company seemed to have adopted a strategy of being geographically close to a University, offering students opportunities to carry out their internships and to learn and develop their technical knowledge, serving as a basis for their future career. On the other hand, the company benefited from the low wages at initial career stages. This organisational strategy had a strong impact on the motivation of this company's employees.

The essential driver of the motivated behaviour in this organisation was *«learning opportunities»*. Factors such as *«intellectual challenge»*, *«autonomy»*, *«collaborative work»*, and *«teammates' technical competence»* created an effective learning environment, which benefited the goal commitment of the software engineers. However, as time passed, their activities started to get repetitive and with low motivational potential in later phases of the projects, regardless of whether they were working for the internal products or for external customers.

This learning environment seemed to contribute to the engineers' job performance factors. On the other hand, three factors generated a large amount of requirements changes and rework, which negatively affected the engineers' job performance: (1) unclear *«task identity»* (the identification of individual work as part of a whole and identifiable piece of work) due to the confused requirement management process; (2) the low *«quality of intermediary artefacts»*; and (3) the low *«feedback»* due to distance between the end-user and the development teams.

The high *«sociable environment»* and *«team integration»* helped to create high *«team cohesion»*, building a very reciprocal work environment, filled with mutual help, shared experiences, and self-responsibility. *«Responsibility»*, in turn, was enhanced by high *«employee participation»*, supported by the structure of small teams and the direct participation of upper management in the development teams.

However, even with a highly agreed sense of *«commitment»* in the teams and the presence of several other contextual motivators, *«turnover»* rates were still high. Because the organisation did not offer long-term *«career opportunities»*, the *«growth needs»* of the software engineers had not been fulfilled, and external opportunities for career progression were constantly draining people off.

In summary, this company was positioned as a great experience for software engineers' first job, where they could take advantage of the effective learning environment to improve both their knowledge and general competence, and to build their *«reputation»* (through recognition for high job performance) in their professional networks towards future job opportunities and growth in other organisations. From the point of view of the company, there seemed to be no means to wrestle against the turnover of experienced professionals. In other words, the intention to leave in order to fulfil individual growth needs offsets the positive aspects of other motivators as soon as the individuals had achieved the initial goals of learning and reputation building.

4.2. Relating the two studies

The most important contrast between the case studies is related to the internal organisational characteristics. Consistent with a maximum variation sampling strategy, we chose two very different software organisations, but within the same economic, social, and educational environment. Therefore, similarities in the findings of the two studies would increase their reliability or external validity. The contradictory findings would open the opportunity for explanations based on the contextual differences, thus enriching our understanding of the phenomenon in highly distinct environments. The main similarities and contrasts between the study findings are briefly described as follows:

- *Organisational attractors*: The motivation for joining the organisations was distinct. In Case 1, task significance and job stability were the factors that motivated individuals to join and stay in the company. In Case 2, the learning opportunity was the most important factor that attracted individuals to join the company, while goal achievement and performance improvements were the outcomes of motivated individuals because they wanted to build a reputation that would be important for future job opportunities.
- *Intention to leave*: The single most important outcome of low motivated behaviour was intention to leave in Case 1, which was caused by frustrated growth needs due to poor career planning support in the organisation. Individuals tended to leave the company when this frustration was higher than the motivating power of task significance. When leaving the organisations, individuals also tended to move to other public organisations, keeping their job stability. Factors specific to the public sector, such as political influences overcoming technical decisions, also played a mediating factor in decreasing motivation and, thus, increasing intention to leave. As an interesting contrast, intention to leave did not emerge as an outcome from the data in Case 2. However, the company experienced high voluntary turnover explained by the fact that many software engineers reached a point in their career when learning was less important than career growth. Intention to leave in Case 2 was not associated with motivation by either the software engineers or the managers and directors, and turnover was considered part of an individual career progression. Although in both cases growth need was a determinant factor for employees to leave the organisation, in Case 1 this need was frustrated by the organisational context, thus negatively impacting motivation, while in Case 2, better external opportunities for growth were the primary cause.
- *Team cohesion*: team cohesion and its positive effect on teamwork conditions in general was the single most important moderator of the influence of negative factors on motivation in both contexts. In Case 1, team cohesion was a direct moderator in the tension between high and low motivation caused by other factors. In Case 2, team cohesion also played a moderator role by directly influencing the conditions for a good learning environment through trust and knowledge sharing.

For the core elements of the central stories in both cases, there are important similarities that support the validity of the results in more general terms. However, specific relationships among factors are distinct in each case and the cross-case synthesis will be used to understand and explain these distinctions. Further, the synthesis will also be important as a systematic method to map the similarities, thus increasing the reliability of the resulting theory of motivation.

4.3. Translating the studies into one another

We performed the translations between the studies o, two levels, directly related to the levels used in each individual case to build the grounded theories. Therefore, we translated motivators and outcomes of motivated behaviour and the propositions of each case into the other. The results highlight the confirmatory translations (in which findings related to the same concept in both studies) and the contrasting translations (in which results from one case did not hold in the other one).

4.3.1. Translating motivators and outcomes

Two types of concepts were translated, factors affecting motivation and the outcomes related to motivated behaviour. In the case studies, factors are classified in three categories: *characteristics of the task*, *team and teamwork*, and *organisational context*. Certain factors had a positive effect and increased motivation while others had a negative effect and decreased motivation. In this presentation, factors that have a negative effect are followed by a minus sign in parenthesis (−). Factors without a sign hold a positive effect.

Outcomes related to motivated behaviour are also presented in three groups: *emotional signs*, *attitudes* in the workplace, and *job performance*. Outcomes can be the result of the behaviour of an individual with high motivation or with low motivation. Outcomes associated with low levels of motivation are followed by a minus sign in parenthesis (−) and those associated with high levels of motivation are not marked.

Precise characterisations of factors and outcomes, with unambiguous and non-conflicting definitions, based on the data and also on the literature, were presented in two technical reports [12,13].

In the translation of factors and outcomes, we checked the name and definition of each concept presented in the published articles [10,11] and in the full reports of each case [12,13]. From reading the case studies (step 3 of the method), relating them (step 4), and from the literature on meta-ethnography, we anticipated possible situations that could occur when translating the concepts from one case to another, and proposed the solutions described in Table 4.

It is important to remark that our solution to generalizations could entail some degree of threat to validity. Further, it is

Table 4
Translation strategies for first level concepts.

Type of translation	Situation	Solution
Identical	The same label and definition was used for a concept in both cases studies	We simply repeated the concept as the translation
Renaming	Different labels were used for a concept in each study, but the definitions were equivalent	We chose the label that better expressed the meaning, consulting thesaurus, dictionaries, and the literature to support the choice
Generalization	Different concepts were found in each case, with different names and definitions, but one concept could be interpreted as a generalization, or abstraction, that included one or more concepts in the other study	We used the more general concept as the translation whenever it expressed the findings of both studies
Localization	A concept was found in one study but not in the other	We kept the concept as the translation with a remark that it was context dependent and associated the concept to its context
Refutation	A concept in one study contradicted a concept in the other study	We tried to understand and explain the refutation based on contextual data and added the explanation to the translation

Table 5
Example of translations: Task characteristics.

Case 1	Case 2	Type	Translation
Learning (domain knowledge) Opportunities	n/a	Localization	Learning (domain knowledge) Opportunities (Case 1)
n/a	Learning Opportunities	Localization	Learning (software engineering) Opportunities (Case 2)
Intellectual Problem Solving	Intellectual Challenge	Generalization	Intellectual Challenge
Autonomy	Autonomy	Identical	Autonomy
Task Variety	n/a	Localization	Task Variety (Case 1)
Task Significance	n/a	Localization	Task Significance (Case 1)
n/a	Task Identity	Localization	Task Identity (Case 2)
n/a	Self-efficacy	Localization	Self-efficacy (Case 2)
n/a	Lack of Skills Update (−)	Localization	Lack of Skills Update (−) (Case 2)

n/a: Not applicable.
(−) Negative effect, preventing motivation.

debatable that, from a pure interpretive or constructive stance, this type of generalization is appropriate. We therefore used this type of translation with parsimony and tried to be careful with the claims we based on them. Additionally, we anticipated that refutations could arise, as described by Noblit and Hare [44] p. 48, but did not find any instance in this current study.

In Table 5, we describe the translation of factors related to Task Characteristics. Identical and Localization translations were trivial. In the former case, the names and meanings of the concepts were the same, and the latter there was actually no translation because the concept is context dependent. Generalization was more challenging and this is illustrated by two situations in Table 5:

- First, *«Learning Opportunities»* from Case 2 would be a good candidate to generalize *«Learning (domain knowledge) Opportunities»* from Case 1. However, scrutinizing the results presented in the reports and also referring to the coded data, it was possible to identify that in Case 2, Learning referred to acquiring technical skills and competence in software engineering. Analysing the contexts, we concluded that the motivators are substantially different and would not be appropriate to make a generalization. In fact, we decided to annotate the concept from Case 2 to make this distinction clearer.
- In the second situation, *«Intellectual Problem Solving»* from Case 1 and *«Intellectual Challenge»* from Case 2 were combined. We reached this translation by analysing the quotes that resulted in coding *«Intellectual Problem Solving»* in Case 1. From the data, we concluded that problem solving was motivating because it entailed a high level of intellectual challenge. Therefore, it would be appropriate to use *«Intellectual Challenge»* as the factor that expressed the phenomenon in both cases.

The need to look deeper into the data to find explanations that support the translations, in particular in the cases of generalisation, enriched our knowledge of both studies. This is an important advantage of conducting a cross-case analysis. These enquires would not naturally occur in the individual cases without the attempt to translate one case into another. We performed all translations of concepts in the same way. In Appendix A, we present the translations of the remaining factors and outcomes.

4.3.2. Consolidating the propositions

Propositions are causal relationships among concepts that offer a hypothetical explanation of the phenomenon of interest in the theory [46]. As such, they can and should be subject to empirical verification. Translating the propositions across the case studies entailed more reflection, interpretation and induction than with the concepts, and the process was similar to performing axial and selective coding in grounded theory [41], which was compatible with the method used to build the propositions in each individual study.

We built the translations in three stages. We first put side by side the propositions that related similar concepts. It was possible to identify relationships for most propositions in both studies. In the second stage of this translation, as proposed in meta-ethnography, we classified the translations of the propositions relating similar concepts as in Table 6. We did not find refutational instances in our study, but we believe it can be found in other cross-case analyses in general.

As the third and final stage in the translation of propositions, we renamed the motivators and outcomes in the resulting propositions according to the names that were built in the translation of the concepts. The numbers of the propositions follow their original numbering in Case 1 and Case 2. The translations received new sequential numbers in this current article.

Table 7 shows a straightforward example of a reciprocal translation. The two propositions hypothesized the moderating effect of Team Cohesion on the negative impact of poor organisational characteristics on motivation. Proposition 02 from Case 2 also relates employee participation with goal commitment, but because it is

Table 6
Translation strategies for cross-case propositions.

Type of translation	Situation	Solution
Reciprocal	Propositions related concepts by similar or comparable causal relationships	We identified the common aspects, and phrased a new proposition consistent with the two reciprocal propositions
Refutational	Propositions related the concepts in opposing or contradicting relationships	Contextual information would be used to explain the refutations enriching our understanding of the varying conditions under which the local propositions would hold or fail
Line-of-argument	Propositions related a different set of concepts, with enough intersection and without refutation, allowing the construction of a line-of-argument that could explain the different situations	Higher degree of inductive inference and interpretation was used to build the translations than in the reciprocal case

Table 7
Example of reciprocal translation of propositions.

Case 1	Case 2	Translation
Proposition 5: team *«cohesion and synergy»* act as moderators of the negative impact of *«poor organisational characteristics»* at least up to a certain level and for a limited period of time	**Proposition 02**: *«Team cohesion»* benefits the power of the enhancing effects and attenuates the disruptive effects of the association between the individual and the team motivations, and *«employee participation»* may help to maintain this association healthy by maintaining the *«goal commitment»* high	**Proposition 2**: Team Cohesion is a moderator of the negative effects of poor organisational factors on motivation

Table 8
Example of a line-of-argument translation of propositions.

Case 1	Case 2	Translation
Proposition 2: the motivating force of *«task variety»* and *«Intellectual Problem Solving»* is contingent on the type of software engineering task	**Proposition 01**: *«Intellectual challenge»*, *«autonomy»*, *«collaborative work»* and *«teammates' technical competence»* combined with *«skill variety»*, are conditions for the establishment of an effective permanent constructive *«learning job environment»*, which increases *«self-efficacy»* and *«goal commitment»*	**Proposition 1**: Intellectual challenge, autonomy, collaboration, and peer technical competence positively affect motivation through the creation of learning (domain knowledge and software engineering) opportunities that increases commitment to perform the job
Proposition 3*:* the motivating force of the need to acquire *«Knowledge from different domains»* is moderated by individual characteristics, being higher for those individuals that like *«constant learning»* and *«skill development»*	**Proposition 03**: A *«learning environment»* protects the engineers' *«self-efficacy»* through self-esteem and self-confidence mechanisms of overcoming individual failures, contributing to the constant increase of *«goal commitment»* and consequently *«job performance»* factors	**Proposition 1.1**: the positive effect of learning opportunities on motivation is moderated by individual characteristics **Proposition 1.2**: the positive effect of learning opportunities on motivation is moderated by the matching between the type or learning opportunity (domain knowledge or software engineering skills) and the learning needs of the individual

Table 9
Example of a line-of-argument using complementary information.

Case 1	Case 2	Translation
Proposition 6: *«Goals and priorities»* defined based on political instead of technical arguments act as a negative force on software engineer motivation that decreases organisational *«commitment»* and increases *«intention to leave»*	From the central story of Case 2: unclear *«task identity»*, the *«low quality of intermediary artefacts»*, … generate large . . . requirements changes and rework, which negatively affect engineers' *«self-efficacy»*	**Proposition 5**: Goals and priorities based on non-technical arguments and constant goal and priority changes negatively affect **motivation**, having a direct effect on commitment and intention to leave

not stated in Case 1, we did not include the relationship in the reciprocal translation.

Table 8 shows the construction of a line-of-argument translation from propositions that established similar, but not equivalent, relationships. With this translation, we identified the role of learning opportunities in both contexts, and the moderating effect of task type and individual characteristics on the motivating power of intellectual challenge and learning. In this line-of-argument, the first proposition emphasizes the positive effect of learning opportunities on motivation and the remaining propositions describe the moderating effect of individual characteristics and type of task.

Table 9 shows a line-of-argument translation for a situation in which a proposition in Case 1 had no similar explicitly stated proposition in Case 2. To construct the translation, we scrutinized the central story of Case 2, looking for propositions implicit in the story that had not been explicitly state in the case report. We found a sentence that related rework and changes due to unclear task identity as producing a negative effect on motivation. We then constructed Proposition 5 to express the argument that unclear goals or non-technical priorities have adverse effect on motivation, which is also supported in the case studies by the strong technical orientation of the software engineer.

As for the concepts, we performed all translations of propositions in a similar way. In Appendix B, we present the translations of the remaining propositions.

4.4. Synthesizing the translations: A higher level central story of motivation

The translation and synthesis of the factors, outcomes, and their relationships stated as propositions, forced us to perform a deeper investigation on contextual and individual characteristics that would explain differences in the findings of the two studies.

4.4.1. Describing context

It does not make sense to synthesize the two contexts. The very nature of the maximum variation sampling method resulted in sufficiently different contexts to increase the richness of data and results. However, it is exactly these differences that can be used to give a higher explanatory power to the new central story of motivation. We summarize the relevant differences in the two contexts in Table 10.

All characteristics presented in Table 10 are contrastingly different. Noteworthy are the age, size in terms of number of employees, and the relative percentage of software engineers with respect to the rest of the employees. The Case 1 organisation was more mature, was regulated by a more strict set of laws related to employee selection and job stability, and produced services and products that directly affect the life of citizens who depend on Government services. The Case 2 organisation was a typical technology intensive start up, with a high percentage of technical staff and developied solutions for clients from the private sector. These contrasts affected motivation both in terms of attraction to work in the organisations and to remain in the job, as will be explained below.

4.4.2. Describing the software engineer

As a result of choosing contrasting cases, we also found two groups of software engineers with high variation in some important characteristics. Although this was not intended during the case selection, the contrasting characteristics of the two groups of software engineers are also important to contextualize and enrich the central story of motivation. This is consistent with the main tenets of grounded theory [38,41]. In fact, based on the literature and on our interpretation of the evidence collected in the case studies, we contend that the differences in individual characteristics play a central role in the different views of motivation between the studies. In the MOCC model [6], the mediating role of individual characteristics on the effect of motivators on behaviour is also stated, albeit in a more general level. We summarize the characteristics of the two groups of software engineers in Table 11. We did not include directors and managers, because our unit of analysis was the software engineer.

We ordered the two groups of participants in Table 11 by their age. It is clear from Table 11 that the participants in each case had distinct characteristics related to age, years in the profession, years with the organisation, education, and their functional role in software development. Participants in Case 1 were older and had been

Table 10
Describing the contrasting organisational contexts.

Characteristics	Case 1	Case 2
Years of Existence	44	6
Nature of Business	Public Sector	Private Sector
Nature of Products	Services to Citizens and other Government Organisations	Products to other private companies
Number of Employees	2580	27
% of Software Engineers	10% (272/2580)	66% (18/27)
Recruiting and Selection	Open process with universal access based on objective criteria	Individual evaluation based on subjective and objective criteria
Job Stability	By law after 3 years probation	None

Table 11
Describing the software engineers.

Software Engineers	Gender	Age	Years in the Profession	Years in the Organisation	Education	Functional Role
(a) Characteristics of Participants in Case 1						
#3	Male	27	5	4	University Graduate	Tester
#2	Male	29	8	4	University Graduate	Programmer
#6	Female	29	8	2	University Graduate	Systems Analyst
#7	Female	29	7	4	University Graduate	Systems Analyst
#9	Male	30	8	3	Masters Student	Software Architect
#8	Female	31	9	2	Masters Degree	Software Architect
#1	Male	32	10	4	University Graduate	Systems Analyst
#4	Male	32	10	2	University Graduate	Software Architect
#5	Male	39	10	2	University Graduate	Systems Analyst
(b) Characteristics of Participants in Case 2						
#2	Female	21	<1	<1	Student	Tester
#8	Male	21	<1	<1	Student	Programmer
#4	Male	22	1	1	Student	Designer
#6	Male	23	<1	<1	Student	Programmer
#10	Male	23	1	<1	Student	Tester
#11	Male	25	4	1	University Graduate	Product Owner
#7	Male	26	7	5	Masters Student	Programmer
#9	Male	26	4	<1	Student	Programmer
#5	Male	27	9	1	University Graduate	Programmer
#3	Male	28	7	<1	Student	Designer
#1	Male	29	8	1	Student	Programmer

in the profession longer. As a consequence, most of them were in the functional roles of system analyst and software architect, with only one in the role of programmer. Participants in Case 2 were younger, several had 1 year or less of career experience and five of them were having their first job experience in the company. Consistent with age, all participants in Case 1 had at least a university undergraduate degree, while the majority of those in Case 2 were still completing their university studies (nevertheless they were hired as professionals not as trainees). Apart from one Product Owner (a SCRUM role), all other participants in Case 2 were working as programmers, testers, or interface designers.

We believe that this characterises two groups of individuals at different stages in their career, although we have not collected enough data to allow a formal characterisation of career stage [47]. Nevertheless, this could explain some of the differences in perception and their feelings towards motivation in the workplace. As will be discussed in the next section, although we found similar motivators and relationships in both cases, subtle but important variations in the strength and role of each motivator were found between the groups. We believe that the differences in contexts and in the characteristics of the participants can provide important insights into these differences. These findings reinforce the adequacy of maximum variation in sampling both the organisation at the level of the case study and the participants at the level of the unit of analysis.

4.4.3. Building the central story of motivation

We used the propositions and the central stories of the two cases to identify and select the common elements with strongest explanatory power in both cases. In the rest of this section, we first describe the influence of organisational factors and then explain the effects of task related characteristics on motivation. Finally, we explain the moderating effect of team cohesion and summarize the central story.

Fig. 4 summarizes the organisational factors that affected motivation in our case studies. We used dashed lines and boxes to relate the information to Case 1 and dotted ones to relate to Case 2.

Poor organisational characteristics were related to *«goal and priority definitions»*, *«feedback»*, and *«career progression support»*. Unclear goal definitions, non-technical priorities, and goal instability negatively affect motivation (Proposition 5 – Appendix B). The engineers in Case 1 emphasized the negative effect of politically instead of technically driven goal definition and in Case 2 the emphasis was on goal instability. The type of organisation explains the different emphases, with the organisation in Case 1 being more subject to political influences in goal and priority definitions.

In both cases, poor *«career progression support»* negatively affected motivation, which, in turn, increased *«intention to leave»* the organisation. The central difference in the two contexts was that in Case 1, intention to leave was directly associated to motivation while in Case 2 intention to leave was perceived as a natural career path and as influenced by *«external opportunities»* for growth outside the company. Both situations together offer complementary explanations for the intention to leave as the tension between internal growth opportunities (provided by clear career progression support) and external growth opportunities (provided by other companies). In this case, the influence of external opportunities was moderated by individual characteristics in that younger professionals tended to be more influenced by these opportunities than more mature professionals, for which job stability could compensate for poor career progression support. Therefore, the effect of motivation on retention (or on the intention of not to leave/stay)

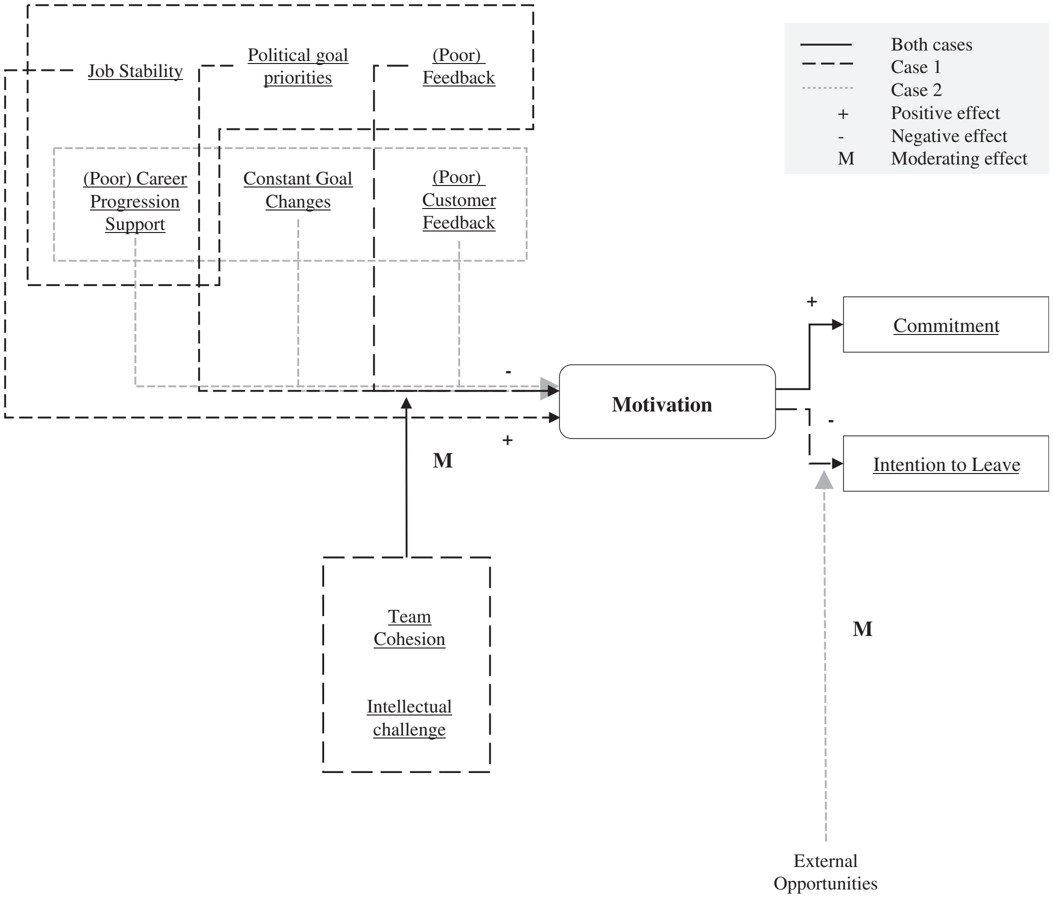

Fig. 4. How motivation is affected by organisational factors.

was limited if the career objectives of the individual included changing jobs to fulfil individual growth needs (Propositions 3 and 4 – Appendix B).

«Job stability» emerged as a motivator only in Case 1, in which stability was a legal job condition in the organisation. In addition to this organisational characteristics, engineers in Case 1 were in a later career point than those from Case 2, for whom stability seemed to be less important than *«learning»* and performing well to build a reputation and to be able to move to better job opportunities in the future. Further, job stability was a motivator to join the organisation (attractor) in Case 1 and its effect on outcomes of motivated behaviour, such as commitment to perform better, was not directly perceived in the study. Therefore, the effect of job stability on motivation seemed to cease as the individual joins the organisation, being of very small importance to create outcomes related to motivated behaviour. Hall et al. [8] consider job security as an extrinsic motivator, but they are not explicit as to whether this motivator has a direct effect on job outcomes or acts mainly as an attractor. In our study, this distinction was clearly observed.

Poor *«feedback»* was identified in both contexts and engineers in Case 2 emphasized absence of *«customer feedback»*. This can be explained by the need of younger and less experienced professionals to receive acknowledgement to increase their self-confidence. The negative effect of poor feedback on motivation was associated with a negative impact on *«goal and performance commitments»*, but not so much on *«intention to leave»* the organisation (Fig. 4).

Task related factors associated with motivation found in the studies were *«task significance»*, *«autonomy»* to choose how to perform the tasks, and the existence of conditions for *«learning»* as part of the task development (Fig. 5). Autonomy has been postulated as a motivator in most theories in the literature [1,48,49] and also appears in the MOCC model. Autonomy was identified as motivating in both case studies.

The creation of conditions for technical *«knowledge exchange»* on the job provided engineers with *«learning opportunities»* that were found to be motivating. The creation of these conditions was driven by *«collaboration»* among individuals, the *«technical competence»* of these individuals, the drive for *«intellectually challenging tasks»*, and *«autonomy»* (Propositions 1 and 1.1 – Appendix B). The effect of learning opportunities on motivation was identified in both cases. In Case 2, this effect was related to learning software engineering skills, which is consistent with the engineers being at an earlier stage in their career. In Case 1, the emphasis of the learning opportunities was on acquiring domain knowledge skills, which can be explained in two complementary ways. First, engineers in Case 1 had more professional experience and, for this reason, did not perceive the opportunities to acquire more technical skills as motivating. Second, the nature of the products and services provided by the organisation in Case 1 required knowledge of the application domains associated with public administration and learning this knowledge represented an opportunity for the engineers to increase their effectiveness and this was perceived to be motivating (Proposition 1.2 – Appendix B). Therefore, learning opportunities were important in both situations, and the type of skills perceived as more important was related to the nature of the products and services developed, as well as the career stage of the software engineers.

Finally, a highly *«sociable environment»* was identified in both cases, which created the conditions for high levels of *«cohesion»* in the software teams in the organisations. Cohesion and other team characteristics, such as *«collaboration»*, *«synergy»*, and *«effective communication»* have been postulated in the literature as important moderators of the effects of organisational characteristics on motivation [1]. High levels of cohesion tended to attenuate the negative effect of poor *«goal priority»* definitions and poor *«feedback»*, because the good sociable environment existent in

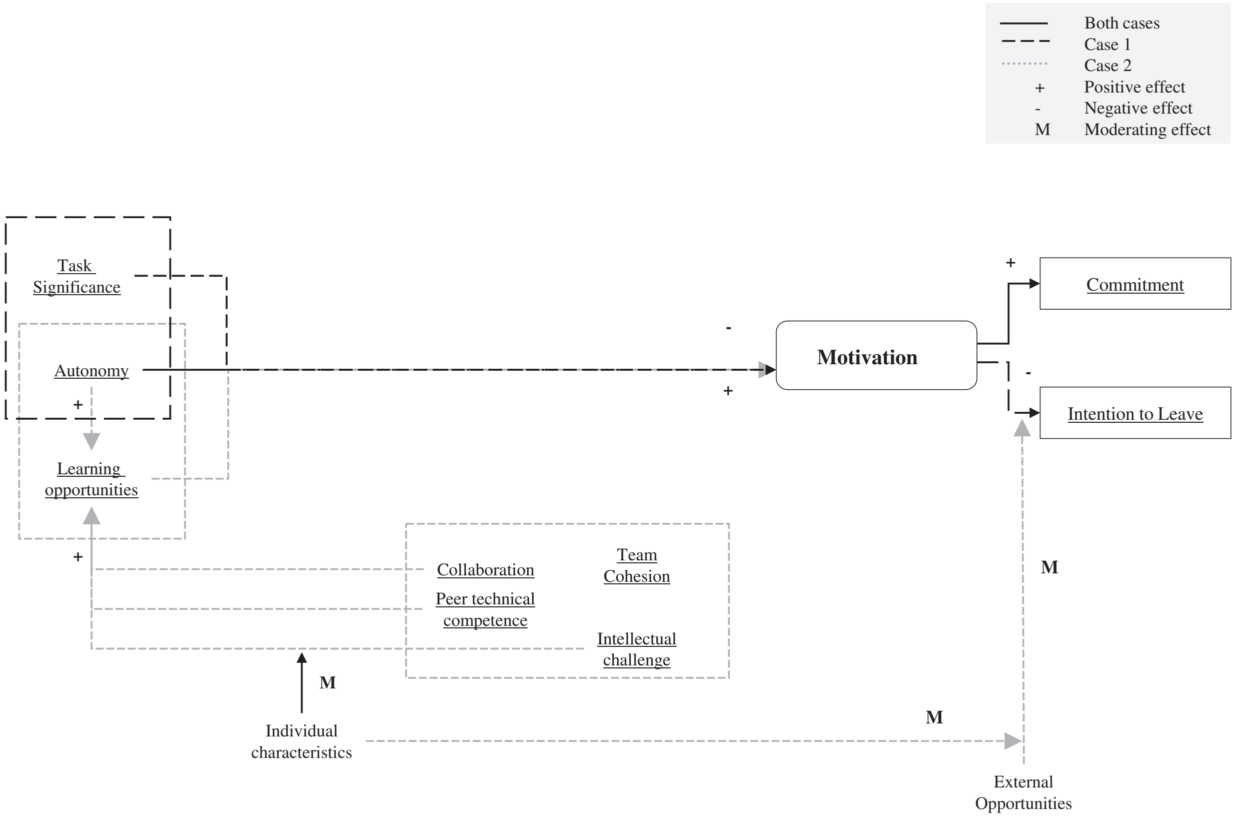

Fig. 5. How motivation is affected by task related factors.

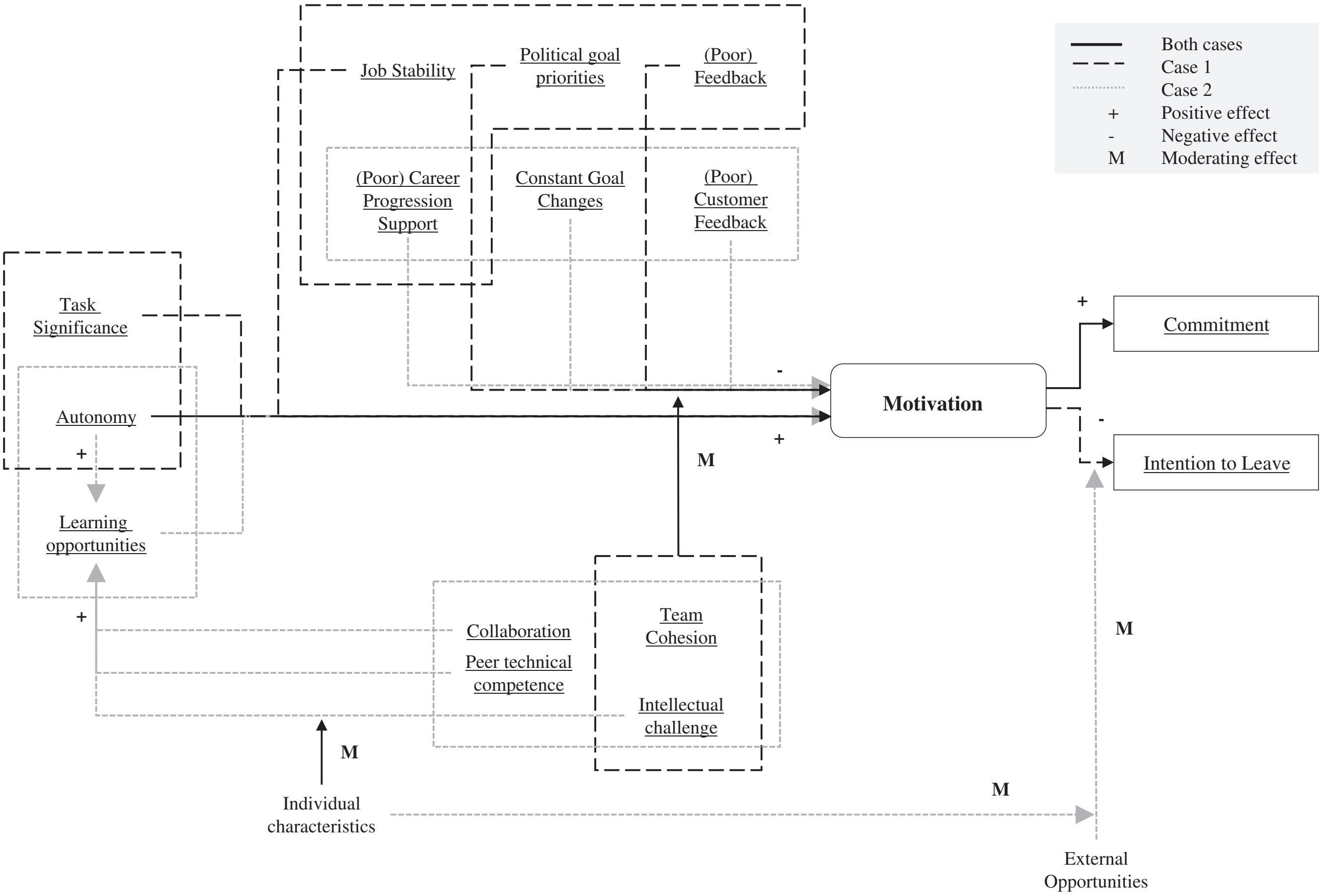

Fig. 6. The elements of the central story of motivation.

highly cohesive teams seemed to compensate these poor characteristics of the organisation (Proposition 2 – Appendix B). However, this moderating effect was temporary. As the poor conditions persisted in the organisation, cohesion seemed to lose its importance with respect to the other factors (see Fig. 6).

In summary, the central story that explains motivation was based on the balance between the negative effects of poor organisational factors and the positive effects of task characteristics, and these effects are moderated by the cohesion of the groups in the working environment. This interpretation is the core of the synthesis, and seems to be valid for both contexts. How these effects happened, including the moderating effects of individual characteristics and the external environment, had distinct explanations due to the differences in organisational contexts and the corresponding differences in the characteristics of the software engineers. By considering relationships that were common to both cases, we achieved a more general description of the motivation, and by adding context dependent interpretations of these relationships we kept the richness of each individual study. We contend that this synthesized theory of motivation increases our knowledge of the phenomenon beyond what was achieved with the individual studies.

5. Discussions

5.1. Reviewing previous models

Our research contributes to the body of knowledge on motivation in software engineering, first, by showing a theory that explains the complex interplay among motivational factors at the task, organisation, and individual levels, in which the role of team cohesion is emphasized. This theory is consistent with the Job Characteristics Theory of Hackman and Oldman [2], Hackman's model of team design and effectiveness [1], the models of self-managed team effectiveness of Cohen [49], and Yeatts and Hyten [48], and complements and extends previous descriptive models in software engineering research such as the work of Couger and Zawacki [27] and the MOCC [6].

Couger and Zawacki [27] claimed, in 1980, that software programmers and analysts had some unique differences from the general population; they had substantially higher growth needs and lower social needs. Regardless of whether this difference holds true after 30 years, in this cross-case study we provided evidence that the growth perspectives of software engineers are important for motivation. Regarding the low social needs, Couger and Zawacki said that it "does not mean that teams should not be utilized. It does indicate that DP [Data Processing] professionals are not actively seeking a team experience." [27] (p. 27). Although our data is convergent to Couger and Zawacki's, with respect to the satisfaction of the professionals with their co-workers, our findings clearly point to a different interpretation of this fact: team cohesion and a sociable environment clearly moderated the effect of almost all organisational factors on motivation. This difference may be attributed to the changes that the software engineering profession has undergone since that time, as anticipated in Beecham et al. [4], making teams a fundamental factor in the establishment of a motivating working environment.

Our findings complement the more recent results of Sharp et al. [6] (summarized by Hall et al. [8]) in several ways:

- We added new evidence for previously predicted motivational factors (e.g. quality of management) and outcomes (e.g. intention to leave), strengthening the external validity of (at least part of) those previous results.
- We found evidence of the influence of new factors (e.g. peer technical competence) and new outcomes (e.g. emotional signs) were not present in the MOCC Model.
- We found that certain factors that are stated at a general level in the MOCC, such as ''learn new skills'', can carry quite distinctive meanings depending on context (in this case, learning domain knowledge and learning programming skills).
- We proposed a different structure to organise the set of motivators according to their relationship to the task, to the team and teamwork, or to the organisation. We also grouped outcomes according to their emotional, behavioural, or job related nature. These categories more closely related to established motivational [1] and effectiveness [48,49] models, and are easier to relate to software engineering practice.
- By showing the relationships among individual factors rather than only among categories of factors, as in the MOCC, we were able to show that some motivators have an indirect effect on outcomes through their direct effect on other factors. For instance, collaboration has an indirect effect on motivation through its direct effect on creating better learning opportunities. Therefore, we increase the explanatory power of the relationships described in the MOCC Model.
- Finally, teams and teamwork received a more explicit treatment in our grounded theory which is important not only because of the intrinsic importance of teamwork in current software practice but because team factors, like team cohesion, seemed to act as a moderator of the negative effect of poor organisational aspects, as predicted by Hackman [1] and Yeatts and Hyten [48].

5.2. Overall implications for research

The case studies presented in this paper were developed following a multi-case replication design with a precisely defined research protocol that has been tested in practice. Other researchers in other contexts and situations can develop similar studies based on our protocol and design. Further integration of results from other case studies would be important to improve our understanding of motivation in software organisations.

However, synthesizing qualitative studies is subject to debate in other sciences. On one side of this debate, advocates of the synthesis approach defend that qualitative synthesis enhances the value of individual studies and of the qualitative research approach in general. On the other side, those that oppose the idea contend that the principles of standardization and generalization that are at the heart of research synthesis, which ultimately aims at universal truth, violates the principles of contextualization and unconformity that are at the core of interpretive and constructivist enquiry.

We believe this is an important debate. As mentioned before, we stand on the side that believes that the findings of qualitative research that use multiple case studies are likely to be stronger than having only one case. However, our experience in performing this synthesis has shown that synthesis and integration has to be properly and carefully conducted, and that contextualized knowledge also plays a central role in our understanding of the phenomena of interest. Producing stronger results that could be transferred to a wider range of contexts will also be in conflict with deep and rich explanations of localized contexts that give us a better understanding of reality, albeit of a very small part of it. We believe that the community of empirical software engineering researchers interested and involved in conducting qualitative research and synthesis should engage in this debate.

5.3. Implications for industrial practice

This cross-case analysis reveal a number of findings that can have implications for software organisations in their quest to create better work environments and to improve the motivation and satisfaction of software engineers. We briefly discuss five findings that we believe have direct practical relevance.

Finding #1: Clear growth perspectives in the organisation are central to creating conditions to improve individual motivation and these conditions tend to have a positive impact on job retention. This result is supported by other more general theories [1]. Our study also emphasized that the effect of growth opportunities is contingent on career stage. Organisations should be aware that growth opportunities directed to early stages in career development are likely to be motivating to attract young professionals and to maintain them until their growth needs change in later stages of their development.

Finding #2: Goal clarity is a central motivator for software engineers given their strong technical orientation. Similarly, priority definitions and changes that are not clearly justified and, ultimately based on non-technical aspects, tend to diminish motivation and organisational commitment.

Finding #3: The role of motivation in increasing retention or, complementarily, in decreasing voluntary turnover is not as direct as postulated in other studies, as in the MOCC Model [6]. We showed that the effect of a motivating environment on retention is strongly moderated by the existence of external opportunities that provide the software engineers with perspectives in career growth beyond those that are possible in the company. Career stage seems to moderate the effect of the external opportunities, increasing their effect for younger professionals that value personal growth more than other factors such as job stability.

Finding #4: Team cohesion is an important factor in the establishment of a motivating working environment. Associated with a cohesive team, a highly sociable environment can exist that influences retention even in the context of poor organisational characteristics. However, the influence of cohesion on motivation is temporary and tends to decrease if the poor organisational characteristics persist. Therefore, strategies to improve team cohesion and the social environment within the organisation must be associated with changes aimed at the solution to improve organisational characteristics.

Finding #5: Motivation, alone, may not be enough to avoid turnover, although it clearly attenuates individual intention to leave, because of the permanent existence of attractive external growth opportunities. As portrayed in previous literature reviews, the largest proportion of research on motivation in software engineering aimed to address the turnover problem, which is indeed a relevant industrial challenge. However, other outcomes highlighted in our study are equally relevant for software engineering practice, such as commitment, productivity, quality of work outputs, creativity and communication. Thus, rather than taking these effects as peripheral benefits of motivation, managers could act more proactively on the motivation of software engineers towards these other outcomes.

6. Limitations and threats to validity

We analysed the threats to validity related to credibility, consistency, and transferability, as discussed in Section 3.7.

6.1. Credibility or internal validity

Credibility in qualitative research has to do with the question of how research results faithfully represent the reality investigated. Two issues related to credibility were relevant in this study. First, in each individual study we tried to ensure that the concepts that emerged from the coding process faithfully represented the meanings attributed by the participants in their answers to our enquires. We used data triangulation (different data collection techniques: interviews and diary studies) and member checking as the main strategies to increase credibility. According to Merriam [34], data triangulation means the use of multiple sources of data to compare and crosscheck the data collected at different times, from different people, possibly using different data collection strategies. Sampling participants with maximum variation and using interviews, diary studies, and document analysis contributed to increasing credibility at the level of each individual study.

At the level of the synthesis, one issue was related to the generalization strategy used in the translation of concepts. The threat was that by performing a generalization of two distinct concepts that emerged from different realities, the translated concept would not faithfully represent both realities or, worse, would represent neither of them. As mentioned and exemplified before, we used the coded data, including transcript excerpts, and the meanings of each concept formalized in the full reports of each case to perform this type of translation. We also consulted the literature to validate the translated names and meanings.

6.2. Consistency or reliability

Reliability is related to the extent to which the research findings can be replicated by the same or other researchers. Replication of research involving human aspects is known to be challenging [50,51], for not other reason than that "human behaviour is never static" [34]. We do not expect that the results of each individual study will be reproduced even if we repeat the same protocol, in the same organisations, with a similar set of participants, because their perception of motivation is likely to change and the organisational context is likely to have evolved. Therefore, what we hope to achieve with respect to reliability is the consistency between our findings and the data collected in each study. For this, we used data triangulation, peer review of the findings by other researchers external to our group, and creation of an audit trail of our data analysis and synthesis that can be used by external researchers to check the consistency of our findings.

In the synthesis of the studies, each level of translation entailed different levels of reproducibility. Translations of concepts were more objective than the translations of propositions, which entailed more induction and interpretation. We expect that the results of the translations of concepts would be easier to be reproduced by other researchers, while the translation of propositions and the synthesis of the translations into a central story could lead to different results given the different interpretations of other researchers. This is part of the richness of the qualitative paradigm, in which the investigator's background, belief, and positions towards the phenomenon can affect the interpretations and result in different, and possibly contradictory, findings. Provided these findings are still consistent with the reality investigated and are not the product of personal bias or prejudice, they are important to increase our knowledge of the phenomenon and to provide alternative theories that can be tested by further empirical studies.

6.3. Transferability or external validity

When concerned with external validity, we want to know the extent to which the research findings can be applied or used in contexts different from those in which the study was first conducted. This is related to how generalizable the results are. As mentioned in Section 3.7, in qualitative research it is the reader or user of the study that should primarily be engaged in the generalization of the research findings. In other words, the reader or user is best equipped to decide to what extent the findings can be applied to their own situations. To the researcher lies the responsibility to provide rich and detailed information about the studies to enhance the possibility of someone else understanding and "transferring" the results. Rich context descriptions are at the heart of transferability in qualitative research. It is by understanding the context and its influence on the results, that the reader can evaluate the extent of which the results can be applied or have to be adapted, or even reconstructed, in the new situation.

We addressed transferability at two levels. First, the research method was detailed and precisely defined, so that other researchers could use the data collection and analysis procedures to produce similar and comparable studies. We tested the research instruments by having independent research groups conducting the two case studies reported in this article, and the results demonstrated that by following the research protocol it is possible to achieve results that can be consistently compared and integrated.

On a second level, we addressed the transferability of the results by providing rich descriptions of each context investigated. These descriptions were individually and independently produced in each case study. Then, when building the integration of the studies, we conducted a deeper investigation of the differences between the two contexts and also in the characteristics of the participants. As discussed above, these differences were essential to explaining the synthesized findings. Furthermore, the descriptions that emerged from comparing and cross analysing the contexts provided richer contextual data that can inform the transference of our findings to other contexts.

7. Conclusions

In this article, we presented a cross-case analysis of two qualitative case studies that investigated motivation of software engineers in industrial practice. We investigated two organisations with distinct characteristics in order to achieve high data variation and richness of results, as advocated in grounded theory [38]. The first investigation targeted a mature and large software organisation in the public sector and the second studied a young and small, private software company. We then integrated the findings from each study using procedures based on meta-ethnography.

Our results showed that organisational and individual characteristics were central in explaining differences in the findings of the individual studies, and to support the development of a more general theory of motivation that was also rooted in each individual context. Further, our results emphasized the role of team cohesion in motivation, consistent with more general theories of work effectiveness [1,48,49]. This result extends related work in software engineering that did not explicitly included cohesion in their models [6,8].

We are seeking to understand why some software engineers seem to be more motivated to perform their work than others, because motivation is closely tied to performance outcomes [1,2]. We are also interested in explaining why different levels of motivation are found among employees that work in the same organisational context and perform similar tasks. We achieved the first steps in this direction by identifying that the influence of motivational factors are likely to be different in individuals of different age and at different career stage.

As part of our future research, we are finishing the development of three other case studies in distinct contexts: a research and development organisation, the information technology (IT) department of a large university, and one Open Source community. We plan to integrate the findings of this current study with these other cases using the same synthesis method presented in this article. We are also planning confirmatory case studies to test part of the resulting theories and the application of action research methods [52] to guide the transfer of our findings to industry in the form of motivational strategies, as part of a long-term research program.

Further, we are conducting a deeper investigation focused on the role of team cohesion in the effectiveness of software teams. As cohesion is perceived by some authors as one of the most important factors that affects team performance, the integration of the studies about cohesion and motivation shall produce a comprehensive understanding of the factors that can be managed to enhance effectiveness of software development in industry. Finally, our findings agree with researchers in other fields that career stage is an important factor to consider in the management of motivation and satisfaction in the work place [53]. We want to collect and analyse data in our future case studies that could provide stronger evidence about the moderating effect of career stage on the relationships between organisational and individual factors and motivation.

Our overall research strategy is to build and integrate local explanatory theories that were constructed from instrumental case studies in different industrial contexts. We believe that, by using this strategy, we will contribute to the general understanding of how to build and develop more effective software teams. Further, the investigation and the practical tests of research methods to produce individual studies, their replication, and integration and synthesis also represent important contributions to the advancements of empirical studies in software engineering.

Acknowledgments

Fabio Q.B. da Silva holds a research Grant from the Brazilian National Research Council (CNPq), process #314523/2009-0. A. César C. França has a doctoral scholarship from CNPq, process #141156/2010-4. We would like to thank all participants in all phases of this research and the organisations involved for granting access for data collection.

Appendix A. Translations of concepts across cases

In this appendix, we present the complete list of translations of concepts, motivators and outcomes, as well as their respective types. Translations can be one of the types described in Table 4, p. 20. The translations were organized with respect to their relationship with the characteristics of the task itself, aspects related to the team and teamwork, and the organizational context in general, reflecting the same structure as they were presented in the case studies [10,11]. Table 12 presents the translations of task-related factors.

Table 13 presents the organizational factors that were translated. As discussed in the text, we intentionally chose very

Table 12
Task-related translations.

Case 1	Case 2	Type	Translation
Learning (domain knowledge) Opportunities (+)	n/a	Localization	Learning (domain knowledge) Opportunities (Case 1) (+)
n/a	Learning Opportunities (+)	Localization	Learning (software engineering) Opportunities (Case 2) (+)
Intellectual Problem Solving (+)	Intellectual Challenge (+)	Generalization	Intellectual Challenge (+)
Autonomy (+)	Autonomy (+)	Identical	Autonomy (+)
Task Variety (+)	n/a	Localization	Task Variety (Case 1) (+)
Task Significance (+)	n/a	Localization	Task Significance (Case 1) (+)
n/a	Task Identity (+)	Localization	Task Identity (Case 2) (+)
n/a	Self-efficacy (+)	Localization	Self-efficacy (Case 2) (+)
n/a	Lack of Skills Update (−)	Localization	Lack of Constant Skills Update (Case 2) (−)

n/a: Not applicable.
(+) Positive effect, reinforcing motivation.
(−) Negative effect, preventing motivation.

Table 13
Organizational context translations.

Case 1	Case 2	Type	Translation
Career Progression Support (+)	Career Progression Support (+)	Identical	Career Progression Support (+)
Employee Participation (+)	Employee Participation (+)	Identical	Employee Participation (+)
Recognition (+)	Recognition (+)	Identical	Recognition (+)
Skill Development (+)	Skill Development (+)	Identical	Skill Development (+)
Low Customer Feedback (−)	Low Customer Feedback (−)	Identical	Low Customer Feedback
High Salary (+)	Low Salary (−)	Generalization	Salary (+)
Infra-structure (+)	n/a	Localization	Infra-structure (+)
Successful Company (+)	n/a	Localization	Successful Company (+)
Job Stability (+)	n/a	Localization	Job Stability (+)
Well Defined Working Time (+)	n/a	Localization	Well Defined Working Time (+)
High Workload (−)	n/a	Localization	High Workload (−)
Low Managerial Feedback (−)	n/a	Localization	Low Managerial Feedback (−)
Politically Driven Goals (−)	n/a	Localization	Politically Driven Goals (−)
Bureaucracy (−)	n/a	Localization	Bureaucracy (−)
Quality of Management (+)	n/a	Localization	Quality of Management (+)
n/a	Time–pressure (−)	Localization	Time–pressure (−)
n/a	Work Environment Flexibility (+)	Localization	Work Environment Flexibility (+)

n/a: Not applicable.
(+) Positive effect, reinforcing motivation.
(−) Negative effect, preventing motivation.

Table 14
Team-level related translations.

Case 1	Case 2	Type	Translation
Sharing Knowledge/Experience (+)	Collaboration (+)	Generalization	Collaboration (+)
Sociable Environment (+)	Sociable Environment (+)	Identical	Sociable Environment (+)
Cohesion (Integration) (+)	Cohesion (Integration) (+)	Identical	Cohesion (Integration) (+)
Peer Commitment (+)	Peer Motivation (+)	Generalization	Peer Motivation (+)
Peer Proactivity (+)			
Peer Trustworthiness (+)			
Peer Technical Competence (+)	Peer Technical Competence (+)	Identical	Peer Technical Competence (+)
Shared Goals (+)	n/a	Localization	Shared Goals (+)
Communication (+)	n/a	Localization	Communication (+)

n/a: Not applicable.
(+) Positive effect, reinforcing motivation.
(−) Negative effect, preventing motivation.

Table 15
External signs of motivation–translations.

Case 1	Case 2	Type	Translation
Commitment (+)	Commitment (+)	Identical	Commitment (+)
Proactivity (+)	Proactivity (+)	Identical	Proactivity (+)
Absenteeism (−)	Absenteeism (−)	Identical	Absenteeism (−)
Quantity of Work Performed (+)	Quantity of Work Performed (+)	Identical	Quantity of Work Performed (+)
Quality of Work Performed (+)	Quality of Work Performed (+)	Identical	Quality of Work Performed (+)
Happiness (+)	Happiness (+)	Identical	Happiness (Case 1) (+)
Enthusiasm (+)	n/a	Localization	Enthusiasm (Case 1) (+)
Calmness (+)	n/a	Localization	Calmness (Case 1) (+)
Self-esteem (+)	n/a	Localization	Self-esteem (Case 1) (+)
Humor (+)	n/a	Localization	Humor (Case 1) (+)
Intention to Leave (−)	n/a	Localization	Intention to Leave (−) (Case 1) (+)
Collaboration (+)	n/a	Localization	Collaboration (Case 1) (+)
Punctuality (+)	n/a	Localization	Punctuality (Case 1) (+)
Communication Effectiveness (+)	n/a	Localization	Communication Effectiveness (Case 1) (+)
Complaints (−)	n/a	Localization	Complaints (−) (Case 1) (+)
Efficiency (+)	n/a	Localization	Efficiency (Case 1) (+)
Creativity (+)	n/a	Localization	Creativity (Case 1) (+)
Timely Deliveries (+)	n/a	Localization	Timely Deliveries (Case 1) (+)
n/a	Concentration (+)	Localization	Concentration (Case 2) (+)
n/a	Frustration (−)	Localization	Frustration (Case 2) (+)

n/a: Not applicable.
(+) Outcome of high motivation.
(−) Outcome of low motivation.

different organizations. This table reflects this decision, so that the most frequent type of translation is localization.

Table 14 presents the translations of team-level related concepts. Although the organizational contexts were very different, the team-level factors that affect motivation were similar in both contexts.

Table 15 presents the external signs of motivation translated. This table clearly shows that the participants in Case 1 reported twelve more signs of motivation than the participants in Case 2. This difference may be explained by the professional maturity of individuals in Case 1, which have reported significantly longer career experience.

Appendix B. Cross-case propositions

In this appendix, we present the complete list of propositions of each case, and then present the translations in Table 16. The six propositions emerged in Case 1 were:

Proposition 1. *«Task significance»* and *«job stability»* form the main attractors for software engineers to work and stay in the organization, and other motivating organizational characteristics reinforce the strength of this attraction, consequently increasing positive outcomes and lowering the *«intention to leave»* the organization.

Proposition 2. The motivating force of *«task variety»* and *«intellectual problem solving»* is contingent on the type of software engineering task.

Proposition 3. The motivating force of the need to acquire *«knowledge from different domains»* is moderated by individual characteristics, being higher for those individuals that like *«constant learning»* and *«skill development»*.

Proposition 4. Poor *«career development support»*, reinforced by other poorly designed organizational characteristics, frustrates the *«growth needs»* of software engineers, increasing the *«intention to leave»* the organization.

Proposition 5. Team *«cohesion and synergy»* act as moderators of the negative impact of poor *«organizational characteristics»*, at least up to a certain level and for a limited period of time.

Proposition 6. *«Goal and priorities»* defined based on political instead of technical arguments act as a negative force on software engineer motivation that decreases organizational *«commitment»* and increases *«intention to leave»*.

The five propositions that emerged in Case 2 were:

Table 16
Cross-case propositions.

Case 1	Case 2	Type	Translation
Proposition 2: The motivating force of *«task variety»* and *«intellectual problem solving»* is contingent on the type of software engineering task	**Proposition 1**: *«Intellectual challenge»*, *«autonomy»*, *«collaborative work»* and *«teammates' technical competence»* combined with *«skill variety»*, are conditions for the establishment of an effective permanent constructive learning job environment, which increases *«self-efficacy»* and *«goal commitment»*	Line-of-argument	**Proposition 1**: Intellectual challenge, autonomy, collaboration, and peer technical competence positively affect **motivation** through the creation of learning (domain knowledge and software engineering) opportunities that increases commitment to perform the job
Proposition 3: The motivating force of the need to acquire *«knowledge from different domains»* is moderated by individual characteristics, being higher for those individuals that like *«constant learning»* and *«skill development»*	**Proposition 3**: A *«learning environment»* protects the engineers' *«self-efficacy»* through self-esteem and self-confidence mechanisms of overcoming individual failures, contributing to the constant increase of *«goal commitment»* and consequently *«job performance»* factors		**Proposition 1.1**: The positive effect of learning opportunities on **motivation** is moderated by individual characteristics
			Proposition 1.2: The positive effect of learning opportunities on **motivation** is moderated by the match between the type of learning opportunity (domain knowledge or software engineering skills) and the learning needs of the individual
Proposition 5: *«Team cohesion and Synergy»* act as moderators of the negative impact of *«poor organizational characteristics»*, at least up to a certain level and for a limited period of time	**Proposition 2**: *«Team cohesion»* benefits the power of the enhancing effects and attenuates the disruptive effects of the association between the individual and the team motivations, and *«employee participation»* may help to keep this association healthy by maintaining the *«goal commitment»* high	Reciprocal	**Proposition 2**: Team Cohesion is a moderator of the negative effects of poor organizational factors on **motivation**
Proposition 4: Poor *«career development support»*, reinforced by other poorly designed organizational characteristics, frustrates the *«growth needs»* of software engineers, increasing the *«intention to leave»* the organization	**Proposition 4**: Given the high *«growth needs»* of software engineers, *«career opportunities»* attached to *«job performance»* factors are effective elements to increase their *«goal commitment»*	Reciprocal	**Proposition 3**: The quality of the career development support in the organization affects **motivation** with a direct effect on commitment to the job and to the organization
Proposition 1: *«Task Significance»* and *«Job Stability»* form the main attractors for software engineers to work and stay in the organization, and other motivating organizational characteristics reinforce the strength of this attraction, consequently increasing positive outcomes and lowering the *«intention to leave»* the organization	**Proposition 05**: *«Goal commitment»*, *«self-efficacy»*, *«learning»*, *«team cohesion»*, and *«job performance»* are not enough to prevent the *«intention to leave»*, if the *«growth needs»* of the software engineers have not been fulfilled, because of external opportunities for career progression	Line-of-argument	**Proposition 4**: The effect of **motivation** on the intention to (not) leave the organization (retention) is attenuated when the external environment provides more stimulating opportunities for career progression. This attenuating effect increases for individuals in early career stages
Proposition 6: *«Goals and priorities»* defined based on political instead of technical arguments act as a negative force on software engineer motivation that decreases organizational *«Commitment»* and increases *«intention to leave»*	From the central story of Case 2: unclear *«task identity»*, the low *«quality of intermediary artefacts»*, ... generate large . . . requirements changes and rework, which negatively affect engineers' *«self-efficacy»*	Line-of-argument	**Proposition 5**: Goals and priorities based on non-technical arguments and constant goal and priority changes negatively affect **motivation**, having direct effect on commitment and intention to leave

Proposition 1. *«Intellectual challenge»*, *«autonomy»*, *«collaborative work»* and *«teammates' technical competence»* combined with *«skill variety»*, are conditions for the establishment of an effective permanent constructive *«learning job environment»*, which increases *«self-efficacy»* and *«goal commitment»*.

Proposition 2. *«Team cohesion»* benefits the power of the enhancing effects and attenuates the disruptive effects of the association between the individual and the team motivations, and *«employee participation»* may help to keep this association healthy by maintaining the *«goal commitment»* high.

Proposition 3. A *«learning environment»* protects the engineers' *«self-efficacy»* through self-esteem and self-confidence mechanisms of overcoming individual failures, contributing to the constant increase of *«goal commitment»* and consequently job performance factors.

Proposition 4. Given the high *«growth needs»* of software engineers, *«career opportunities»* attached to job performance factors are effective elements to increase their *«goal commitment»*.

Proposition 5. *«Goal commitment»*, *«self-efficacy»*, *«learning»*, *«team cohesion»*, and *«job performance»* are not enough to prevent the *«intention to leave»*, if the *«growth needs»* of the software engineers have not been fulfilled, because of external opportunities for career progression.

References

[1] J.R. Hackman, The design of work teams, in: J. Lorsch (Ed.), Handbook of Organizational Behavior, Prentice-Hall, Englewood Cliffs, 1987, pp. 315–342.

[2] J.R. Hackman, G.R. Oldham, Motivation through the design of work: test of a theory, Organizational Behavior and Human Performance 16 (1976) 250–279.

[3] A. Furnham, A. Eracleous, Personality, motivation and job satisfaction: Hertzberg meets the Big Five, Journal of Managerial Psychology 24 (8) (2009) 765–779.

[4] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in software engineering: a systematic literature review, Information and Software Technology 50 (9–10) (2008) 860–878.

[5] A.C.C. França, T.B. Gouveia, P.C.F. Santos, C.A. Santana, F.Q.B. da Silva, Motivation in software engineering: A systematic review update, in: Proceedings of 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011), Durham, UK, 2011, pp. 154–163.

[6] H. Sharp, N. Badoo, S. Beecham, T. Hall, H. Robinson, Models of motivation in software engineering, Information and Software Technology 51 (2009) 219–233.

[7] T. Hall, N. Baddoo, S. Beecham, H. Robinson, H. Sharp, A systematic review of theory use in studies investigating the motivations of software engineers, ACM Transactions on Software Engineering and Methodology 18 (3) (2009) 1–29, http://dx.doi.org/10.1145/1525880.1525883.

[8] T. Hall, H. Sharp, S. Beecham, N. Baddoo, H. Robinson, What do we know about developer motivation?, IEEE Software 25 (4) (2008) 92–94 doi: 10.1109/MS.2008.105..

[9] A.C.C. França, F.Q.B. da Silva, Towards understanding the underlying structure of motivational factors for software engineers to guide the definition of motivational programs, JSS (2011), http://dx.doi.org/10.1016/j.jss.2010.12.017.

[10] A.C.C. França, A.L.C. Felix, F.Q.B. da Silva, Towards an explanatory theory of motivation in software engineering: a qualitative case study of a government organization, in: Proceedings of the 16th International Conference on Evaluation and Assessment in Software Engineering, EASE'2012, Castilla La-Mancha, 2012.

[11] A.C.C. França, D.E.S. Carneiro, F.Q.B. da Silva, Towards an explanatory theory of motivation in software engineering: a qualitative case study of a small software company, in: Proceedings of the 26th Brazilian Symposium on Software Engineering (SBES'2012) Natal, 2012.

[12] A.C.C. França, A.L.C. Felix, F.Q.B. da Silva, Appendices of: Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Government Organization, HASE Technical Report No. 2012-001, 2012. <http://goo.gl/Rs8AO>.

[13] A.C.C. França, D.E.S. Carneiro, F.Q.B. da Silva, Appendices of: Towards an Explanatory Theory of Motivation in Software Engineering: A Qualitative Case Study of a Small Software Company, HASE Technical Report No. 2012-003. <http://goo.gl/1dGn4>.

[14] R. Pritchard, E. Ashwood, Managing Motivation: A Manager's Guide to Diagnosing and Improving Motivation, CRC Press, 2008.

[15] P. Latham, C.T. Ernst, Keys to motivating tomorrow's workforce, Human Resource Management Review 16 (2) (2006) 181–198, http://dx.doi.org/10.1016/j.hrmr.2006.03.014.

[16] A. Bandura, Self-efficacy mechanism in human agency, American Psychologist 37 (1982) 122–147.

[17] R.A. Katzell, D.E. Thompson, Work motivation: theory and practice, American Psychologist 45 (2) (1990) 144–153.

[18] S. Rogers, J. Ludington, S. Graham, Motivation & Learning: A Teacher's Guide to Building Excitement for Learning & Igniting the Drive for Quality, third ed., Peak Learning Systems, Evergreen, 1997.

[19] R.M. Steers, R.T. Mowday, D.L. Shapiro, The future of work motivation theory, Academy of Management Review 29 (3) (2004) 379–387, http://dx.doi.org/10.5465/AMR.2004.13670978.

[20] A.H. Maslow, Motivation and Personality, Harper and Roe, New York, 1954.

[21] F. Herzberg, B. Mausner, B.B. Snyderman, Motivation to Work, second ed., Wiley, New York, 1959.

[22] D.C. McClelland, The Achieving Society, D. Van Nostrand Co., Inc., Princeton, NJ, 1961.

[23] B.F. Skinner, Operant behavior, American Psychologist (1963).

[24] V.H. Vroom, Work and Motivation, Wiley, New York, 1964.

[25] E.A. Locke, Toward a theory of task motivation and incentives, Organizational Behavior and Human Performance 3 (1968) 157–189.

[26] R.A. Guzzo, Types of rewards, cognitions, and work motivation, The Academy of Management Review 4 (1) (1979) 75–86.

[27] J.D. Couger, R.A. Zawacki, Motivating and Managing Computer Personnel, Wiley-Interscience, 1980.

[28] S. Beecham, H. Sharp, N. Baddoo, T. Hall, H. Robinson, Does the XP environment meet the motivational needs of the software developer? An empirical study, Proceedings of the AGILE 2007 (AGILE '07), IEEE Computer Society Washington, DC, USA, 2007.

[29] B. Tessem, F. Maurer, Job satisfaction and motivation in a large agile team, in: Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP'07). Lecture Notes in Computer Science, vol. 4536/2007, 2007, pp. 54–61. doi:10.1007/978-3-540-73101-6_8.

[30] R. Sach, H. Sharp, M. Petre, Software Engineers' Perceptions of Factors in Motivation, in: Proceeding of International Symposium on Empirical Software Engineering and Measurement (ESEM), Banff, AB, 2011, pp. 368–371.

[31] B.E. Wright, Public-sector work motivation: a review of the current literature and a revised conceptual model, Journal of Public Administration Research and Theory (J-PART) 11 (4) (2001) 559–586.

[32] J.L. Perry, L.W. Poter, Factors affecting the context for motivation in public organizations, The Academy of Management Review 7 (1) (1982) 89–98.

[33] M.F. Reid, M.W. Allen, C.K. Riemenschneider, D.J. Armstrong, Affective commitment in the public sector: the case of IT employees, in: Proceedings of the 2006 ACM SIGMIS CPR Conference on Computer Personnel Research: Forty Four Years of Computer Personnel Research: Achievements, Challenges & the Future, ACM New York, NY, USA, 2006.

[34] S.B. Merriam, Qualitative Research: A Guide to Design and Implementation, Jossey-Bass, San Francisco, 2009.

[35] N.K. Denzin, Y.S. Lincoln, The Sage Handbook of Qualitative Research, third ed., Sage, Thousand Oaks, CA.

[36] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical Software Engineering 14 (2) (2008) 131–164.

[37] R.K. Yin, Case Study Research Design and Methods, third ed., Sage, London, 2003.

[38] B.G. Glaser, A. Strauss, The Discovery of Grounded Theory, Aldine, Chicago, 1967.

[39] J. Rieman, The diary study: a workplace-oriented research tool to guide laboratory efforts, in: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, ACM New York, NY, USA, 1993.

[40] D. Dearman, M. Kellar, K.N. Truong, An examination of daily information needs and sharing opportunities, in: Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW'08), ACM New York, NY, USA, 2008.

[41] A.C. Strauss, J.M. Corbin, Basics of Qualitative Research: Second Edition: Techniques and Procedures for Developing Grounded Theory, third ed., Sage Publications Inc., 2007.

[42] D.S. Cruzes, T. Dybå, Synthesizing evidence in software engineering research, in: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement – ESEM '10, New York, New York, USA, ACM Press, 2010, p. 1. doi:10.1145/1852786.1852788.

[43] M. Sandelowski, J. Barroso, Handbook of Synthesizing Qualitative Research, Springer Publishing Company, New York, 2007.

[44] G.W. Noblit, R.D. Hare, Meta-Ethnography: Synthesizing Qualitative Studies (Qualitative Research Methods), Sage Publications Inc., 1988.

[45] N. Britten, R. Campbell, C. Pope, H. Donovan, M. Morgan, R. Pill, Using meta ethnography to synthesise qualitative research: a worked example, Journal of Health Services Research & Policy 7 (4) (2002) 209–215.

[46] D.I.K. Sjøberg, T. Dybå, B.C.D. Anda, J.E. Hannay, Building theories in software engineering, in: F. Shull, J. Singer, D.I.K. Sjøberg (Eds.), Guide to Advanced Empirical Software Engineering, Springer, 2008, pp. 312–336. Chapter 12.

[47] G.W. Dalton, P.H. Thompson, R.L. Price, The four stages of professional career – a new look at performance by professionals, Organizational Dynamics 6 (1) (1977) 19–42.

[48] D.E. Yeatts, C. Hyten, High-Performing Self-Managed Work Teams, Sage Publications Inc., 1998.

[49] S.G. Cohen, Designing Effective Self-Managing Work Teams, in: M. Beyerlein (Ed.), Advances in Interdisciplinary Studies of Work Teams, Series of Self-managed Work Teams, vol. 1, JAI Press, Greeenwich, Connecticut, 1993.

[50] J. Lung, et al., On the difficulty of replicating human subjects studies in software engineering, in: ICSE'2008: Proceedings of the 13th International Conference on Software engineering, New York, 2008.

[51] Fabio Q.B. Da Silva, Marcos Suassuna, A.C.C. França, Alicia M. Grubb, T.B. Gouveia, Cleviton V.F. Monteiro, Igor I. Dos Santos, Replication of Empirical Studies in Software Engineering Research: A Systematic Mapping Study, Empir Software Eng, Springer, 2012. doi:10.1007/s10664-012-9227-7.

[52] R.M. Davison, M.G. Martinsons, N. Kock, Principles of canonical action research, Information Systems Journal 14 (1) (2004) 65–86.

[53] Maxwell K. Hsu, H.G. Chen, James J. Jiang, Gary Klein, Career satisfaction for managerial and technical anchored is personnel in later career stages, The DATA BASE for Advances in Information Systems 34 (4) (2003) 64–72.