Pós-Graduação em Ciência da Computação

The Influence of Job Rotation on Motivation and Satisfaction of Software Engineers

By

Ronnie Edson de Souza Santos

Master Thesis

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

Recife
2015

Ronnie Edson de Souza Santos

The Influence of Job Rotation on Motivation and Satisfaction of Software Engineers

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

ADVISER: ***PROF. FABIO QUEDA BUENO DA SILVA***

Recife
2015

Catalogação na fonte
Bibliotecária Joana D'Arc Leão Salvador CRB4-532

S237i Santos, Ronnie Edson de Souza.
The influence of job rotation on motivation and satisfaction of software engineers / Ronnie Edson de Souza Santos. – Recife: O Autor, 2015.
83 f.: fig., tab.

Orientador: Fabio Queda Bueno da Silva.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIN, Ciência da Computação, 2015.
Inclui referências e apêndices.

1. Engenharia de software. 2. Motivação no trabalho. 3. Satisfação no trabalho. I. Silva, Fabio Queda Bueno da (Orientador). II. Titulo.

005.1 CDD (22. ed.) UFPE-MEI 2015-112

Dissertação de Mestrado apresentada por **Ronnie Edson de Souza Santos** à Pós Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**The Influence of Job Rotation on Motivation and Satisfaction of Software Engineers**" orientada pelo **Prof. Fabio Queda Bueno da Silva** e aprovada pela Banca Examinadora formada pelos professores:

_____

Prof. Sergio Castelo Branco Soares
Centro de Informática/UFPE

_____

Prof. Alberto Cesar Cavalcanti França
Centro de Estudos e Sistemas Avançados do Recife

_____

Prof. Fabio Queda Bueno da Silva
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 22 de janeiro de 2015.

_____

Profa. Edna Natividade da Silva Barros
Coordenadora da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

I dedicate this work to my grandfather, the bravest man I've ever known.

ACKNOWLEDGEMENTS

First of all, I would like to remember my parents, who for seventeen years put my education in first place, and did whatever it took to give the basis that was so important to reach this dream. Now, almost ten years have gone, but somehow, I know that somewhere, they are watching over me and I'm really thankful for that. Quoting the Kooks now, "*if you could see me now, if you could see my smile, see your little boy. Would you be proud?*". I hope you are!

I thank my family, the most important people in the world to me, who always believed in me and followed me into all the projects I was willing to face. My grandfather Eloi, my sister Kecia, and my brother Robson, thank you for always be there for me. I am also grateful to the people who held my hand in the most difficult moment in my life: Aunt Diva, who for ten years is taking care of me as her own son and Uncle Neno who always gave me the support I needed. When I think about family, when I think about you all, I think about Matthew 6:21 *"For where your treasure is, there your heart will be also."*

This master thesis is a result of great work and dedication and if there is someone who encouraged me to make this possible is my advisor Dr Fabio Queda, who conducted me not just in this work, but in a set of other research along these two years, and helped me to give one more step in my way to become a scientist. This is also the moment when I remember other great mentors that I had in my life as a student. My professors from UFRPE-UAST, especially João Ferreira, who taught me the first algorithms and brought me into Computer Science, and also Jorge Correia and Ellen Souza, who accompanied me on my first steps as a researcher, who were fundamental in my five years as a undergraduate student, and who gave me all the support to get here. I am also thankful to my teachers from Escola Monsenhor Luiz Sampaio, where I finished high school and the teachers from Escola Higino Bezerra, where I studied from kindergarten to fourth grade, especially my first teacher Tia Nilzinha, who taught me, not just the first letters, the first numbers, but also the first english words. All of you inspire me to seek the career I want for me: to become a great researcher and to be an excellent professor.

I also owe a special thanks to my friends. First, Ted, I mean, Ross, no wait, Vanut, my eldest brother and research buddy; and also Marshal, well, I'd say Joey, actually, Ivo. You are my best friends, dudes, so thank you not just for being there for me in all the good and bad moments but also for all the beer we drank in this period. It was, LEGEN -wait for it- DARY, legendary! This acknowledgement would not be completed without mention the rest of the gang: Vanessa (my stepsister), Paulo, Gabriela, Emanuelle, and Bel, thank you guys for all the parties and good memories. Also, thanks for the people who was there for me in the first steps of this journey, Celina, Andresa, Clébia, and off course all the friends from Triunfo and Serra Talhada who believed in this. That Muhammad Ali guy was right "*Friendship is the hardest thing in the world to explain. It's not something you learn in school. But if you haven't learned the meaning of friendship, you really haven't learned anything*."

Finally, I would like to thank people from C.E.S.A.R, who were so supportive of this research. Thanks to the people from HASE, my research group, for the openness, the experiences shared and the help, especially those I had the opportunity to be directly working, at least once, Isabela, Marcão, Cleviton, Anderson, Rodrigo, Roberta, and Cesar. And talking about work, a thanks to the colleagues from CIn, who, as me, have enjoyed all the amusements and the afflictions present in the life of a master candidate in computer science, especially Mouglas, Leandro, Sidartha, Carla, and Aline.

This research work was partially supported by the Fundação de Amparo à Ciência e Tecnologia de Pernambuco (FACEPE), through a master degree grant process number IBPG-0651-1.03/12.

"Wit beyond measure is man's greatest treasure."

— Rowena Ravenclaw, Harry Potter and the Deathly Hallows

ABSTRACT

Context. During the last decades, human factors have become of great interest for many software engineering researchers, considering there are a wide variety of human and social aspects that might affect the way software engineers perform their work. As an example of this, recent research revealed a need for the proper management of two elements, the motivation and the satisfaction of software engineers, in order to achieve higher levels of performance at work. In this context, the Theory of Motivation and Satisfaction of Software Engineers (TMS-SE), recently established, confirms this need and recognizes the difference between these two factors, demonstrating that motivated software engineers are engaged and concentrated, while satisfaction is perceived in terms of happiness at work. **Goal**. Although having observed a wide diversity of aspects present at the software development environment, the TMS-SE did not specifically address the practice known as Job Rotation, whereby people are constantly switching jobs or projects at the same organization, and the effects of this practice on the motivation and satisfaction of software engineers. Thus, the main goal of this research is to investigate and discuss how the practice of job rotation can influence the motivation and the satisfaction of these individuals. **Method**. To achieve this goal, a qualitative case study was conducted in a software organization where the practice of job rotation amongst software projects is common. A group of software engineers were interviewed in order to collect data about their experience with this practice. **Results**. The findings suggest that, in a context in which the rotation of software engineers is frequent, it is necessary to find the balance between the positive and negative factors affecting the engagement and the concentration of these individuals, otherwise, their motivation will be impaired by the increase in the cognitive overload at work. In addition, the lack of feedback, resultant from constant movement among projects and teams, has a direct and negative impact on job satisfaction.

Keywords: Software engineer. Work Motivation. Job Satisfaction. Job rotation

RESUMO

Contexto. Durante as últimas décadas, os fatores humanos tem recebido grande atenção de muitos pesquisadores de área de Engenharia de Software, pois estes fatores afetam a forma como os engenheiros de software executam o seu trabalho. A exemplo disto, pesquisas recentes revelaram a necessidade de uma gestão adequada da motivação e satisfação de engenheiros de software para que se possa atingir altos níveis de desempenho. Neste contexto, a Teoria da Motivação e Satisfação dos Engenheiros de Software (TMS-SE), estabelecida recentemente, confirma esta necessidade e estabelece a diferença entre estes dois fatores, demonstrando que engenheiros de software motivados são empenhados e concentrados, enquanto a satisfação é percebida em termos de contentamento com o trabalho. **Objetivo**. Apesar de ter discutido uma grande variedade de aspectos relacionados com o trabalho de equipes de desenvolvimento de software, a TMS-SE não explica diretamente o impacto da prática de *Job Rotation* (rotações de trabalho), através da qual estes indivíduos são periodicamente mudados de equipe e de projeto de software, na motivação e satisfação dos engenheiros de software. Assim, o objetivo desta pesquisa foi investigar e discutir como as rotações de trabalho podem influenciar a motivação e a satisfação de engenheiros de software. **Método**. Para atingir este objetivo, um estudo de caso qualitativo foi realizado em uma organização de software que utiliza a prática de rotação de engenheiros de software, os quais foram entrevistados sobre a sua experiência neste processo. **Resultados**. Os resultados sugerem que, em um contexto em que a rotação de engenheiros de software é freqüente, faz-se necessário buscar o equilíbrio entre os fatores positivos e negativos que afetam o engajamento e a concentração destes indivíduos, caso contrário, sua motivação será prejudicada pelo aumento da carga cognitiva do trabalho. Além disso, a falta de *feedback* sobre o trabalho, tem impacto negativo direto sobre a satisfação no trabalho.

Palavras-chave: Engenheiro de software. Motivação. Satisfação. Rotação de trabalho.

LIST OF ILLUSTRATIONS

LIST OF TABLES

CONTENTS

Chapter 1 Introduction

During the last decades, human factors have become of great interest for many software engineering researchers. The explanation to this phenomenon is the fact that software development process depends on human-centred activities, in which aspects such as motivation, satisfaction, personality, behavior, and several others have direct impact on the effectiveness of the software development process (PIRZADEH, 2010).

There is a wide variety of human and social aspects that might affect the way software engineers perform their work, which makes important to understand the concepts and perceptions around this topic. As an example, for a long time, the term motivation was used as a synonymous to job satisfaction and to several others distinct behaviors of software engineers (FRANÇA, SHARP, da SILVA, 2014). This disagreement among concepts represents a problem both to academic research and industrial practice.

The interest for motivation and job satisfaction has increased, in software engineering context, due to the findings of recent research that revealed the need for the proper management of these two factors in software organizations, to achieve higher levels of productivity at work (FRANÇA, GOUVEIA, et al., 2011). Nevertheless, as observed by França (2014) the existing theories about these concepts, developed in other fields, are not entirely applicable to software development environments because regarding individual needs, computer personnel may be a distinctive group from the average population. Further, the studies on this topic in software engineering research, in general, rarely use well-established and adequate theories to support their findings.

The Theory of Motivation and Satisfaction of Software Engineers (TMS-SE) (FRANÇA, 2014) states that motivation and job satisfaction are not the same concept, contradicting common beliefs among many researchers in the software engineering field. This theory argues that motivated software engineers are engaged and concentrated, while the satisfaction is perceived in terms of happiness at work. Moreover, this theory improves the knowledge on this specific topic in software engineering and also proposes a framework that can be applied to guide the management of software engineers and the aspects around work motivation and job satisfaction in the development environment.

The TMS-SE was constructed based on data collected from software engineers from four distinct software organizations, with distinct goals, different business nature (public and private), different sizes and maturity, but within a similar context. Data was collected by applying semi-structured interviews, diary studies, and document analysis, to observe issues related to experience, behavior, opinion, values, feelings, knowledge, and background around motivation and satisfaction of software engineers (FRANÇA, 2014).

Although having observed a wide diversity of aspects present in the context of the software teams and the software development environment to construct the TMS-SE, França (2014) did not investigate the practice known as Job Rotation, whereby people are constantly switching from a job (or project) to another at the same organization. Woods (1995) properly defines Job Rotation as “the systematic movement of employees from job to job or project to project within an organization” during the development of a task, as an approach to achieve many different human resources objectives.

Previous studies performed in other areas have discussed the effect of this organizational practice and its negative and positive influences on employees’ knowledge, satisfaction, motivation, turnover, and other aspects (CAMPION;

CHERASKIN; STEVENS, 1994; RICHARDSON et al., 2003). Under these circumstances, it is possible that the practice of job rotation in software organizations might influence the motivation and satisfaction of software engineers, who are frequently moved from one software project to another during the software development.

Drawing a parallel between the TMS-SE concepts and the outcomes of studies about Job Rotation, some particularities can be observed. For example, TMS-SE proposes task identity and task variety as two elements that could positively influence the motivation of software engineers at work, while cognitive effort could negatively influence the concentration of software engineers and consequently, their motivation (FRANÇA, 2014). Similar to this, Ollo-Lopez, Bayo-Moriones, and Larraza-Kintana (2010) observed that job rotation increases task identity and task variety, while Eriksson and Ortega (2006) perceived that job rotation forces employees to perform a greater number of different activities and to apply more skills in their jobs, which demands a high level of cognitive effort.

Thus, there is a need for understanding whether job rotation can actually affect motivation and satisfaction of software engineers that are moved among different software projects, while the process is running, and in the light of the TMS-SE, which aspects of motivation and/or satisfaction are impacted by this practice. Besides, this research is also motivated by the existence of a considerable number of studies exploring the impact of job rotation in several fields, but wispy empirical knowledge about this aspect in the context of software engineering. These objectives are summarized in Table 1.

Table 1. Study Goals

To Investigate	The practice of job rotation in which software engineers are frequently rotated among software projects.
Aiming to	Understand wheter this practice can influence the motivation and satisfaction of software engineers.
From the point of view of	Software teams members.
In the context of	A private software organization from Brazil.

To achieve our goal, a qualitative case study was conducted in a private software organization in Brazil, in which a group of software engineers, working in two different projects, were interviewed in order to collect data about their experience with the practice of job rotation used in this organization. In this case study, qualitative research was the most suitable approach considering the need to collect, analyze, and interpret feelings, opinions, and perceptions of software engineers about the frequent rotation of team members occurring in the software development environment.

The outcomes suggest that in a context where rotation is frequent, it is extremely necessary to seek balance between the positive and the negative factors that affect the engagement and the concentration of software engineers, otherwise, motivation of the individuals may be impaired, especially by the increase of cognitive workload. Further, the lack of feedback that may result from rotation, has direct negative impact on job satisfaction. This study contributes to software engineering research in two perspectives. First, the results contribute to the advance of the knowledge about motivation and satisfaction of software engineers.

Second, it provides theoretical and empirical evidence about the impacts of job rotation in a software development environment.

From this introduction, the remainder of this work is organized as follows. Chapter 2 presents the theoretical background about the main topics around this study: the TMS-SE proposed by França (2014), description and concepts of Job Rotation and previous studies performed in the software engineering context about Job Rotation. Chapter 3 describes the methodological approach used to collect and analyse data required to achieve the study goals. Chapter 4 presents the results obtained and Chapter 5 discusses these results. Finally, the conclusions of this study are presented in Chapter 6 together with the limitations encountered during the research and the suggestions of future work.

Chapter 2 Theoretical Background

This chapter presents the theoretical background that supports this study, as well as related works in similar context of this research, following three main topics: Motivation and Satisfaction of software engineers and the TMS-SE proposed by França (2014), Job Rotation, and research about Job Rotation performed in the Software Engineering.

The study of work motivation and job satisfaction is dated from the early 1900s. However, since the beginning there was certain confusion about the distinction between these two concepts. França (2014) argued that the explicit differentiation between them only became recurrent after the 1970's and, nowadays, these concepts remain frequently confounded. Additionally, Beecham et al. (2008) observed that the research covering motivation in software engineering started in early 1980's and has increased since then, based on the premise that what motivates software engineers is different from what motivates the population in general.

Even in more recent studies, researchers are still adopting outdated theories, as theoretical source to their research procedures, and because of this, there are frequently failure on the identification and operationalization of the phenomenon they are actually investigating (FRANÇA, 2014). The TMS-SE, presented in Section 2.1, distinguishes work motivation and job satisfaction as different phenomena in software engineering context, with distinct antecedents and outcomes.

The TMS-SE (Section 2.1) was constructed based on the definitions of two initial theories, the Job Satisfaction Theory (LOCKE, 1969) and the Job

Characteristics Theory (HACKMAN et al., 1975). Both theories were deeply analyzed to be improved and adapted for the software development context, following the findings achieved from a multiple case study that comprised data obtained from four different software organizations (FRANÇA, 2014).

However, none of the case studies conducted to establish the TMS-SE observed whether the scenario in which software engineers are frequently rotating among software projects (job rotation) influenced their motivation and satisfaction. This is the main goal of this research. Thus, Section 2.2 presents definitions and discussions related to the practice of job rotation. Finally, Section 2.3 presents previous studies that have observed and discussed issues related to this phenomenon in the context of Software Engineering.

2.1 The Theory of Motivation and Satisfaction of Software Engineers

França (2014) used data collected from case studies performed in four distinct software organizations to construct the TMS-SE. These organizations belonged to the same business environment, they had similar contexts (social and educational), but different sizes, goals, maturity, and business nature. Two of them are public organizations and the other two are private organizations.

According to the participants interviewed during the research, there are four main representative adjectives that characterize the motivated behaviour: engagement, concentration, collaboration and happiness (FRANÇA, 2014). Table 2 presents the data collected in the four case studies performed to reach this conclusion. Following these findings, França (2014) observed that the behaviour of software engineers is affected by some factors that can be present in the workplace and that represent the antecedents of motivated behaviour (Table 2).

Table 2. Behavioural descriptors for motivated and unmotivated behaviours (França, 2014)

Categories	Positive Adj. (motivated)	Case I	Case II	Case III	Case IV	Negative Adj. (unmotivated)	Case I	Case II	Case III	Case IV
Concentration	Careful			✓*	✓	Careless		✓*	✓*	
	Concentrated		✓	✓	✓	Distracted	✓*		✓*	✓*
	Focused	✓*		✓	✓	Unfocused	✓		✓	✓
Collaboration	Communicative	✓*	✓	✓	✓*	Reserved	✓*	✓	✓	✓*
	Helpful		✓	✓	✓	Unhelpful				✓
Engagement	Involved	✓*	✓*	✓*	✓	Uninvolved	✓	✓*	✓*	
	Hard-working	✓	✓		✓*	Lazy	✓	✓*	✓	✓*
	Interested	✓*			✓*	Indifferent	✓*	✓	✓	✓
	Proactive	✓	✓	✓*	✓	Passive		✓	✓	✓
Happiness	Excited	✓			✓	Bored			✓	✓*
	Good mood	✓	✓	✓	✓	Bad mood	✓	✓	✓	✓
	-					Resented	✓			✓
	Upbeat				✓	-				

✓*- representative in the case
✓ - mentioned in the case, but not representative

- Engagement represents a person that is involved, interested, proactive and hard-working. This characteristic is mainly influenced by the acquisition of useful knowledge at work and the possibility of performing a creative work. Nevertheless, software engineers also observed that engagement is affected by the social impact of their activities, the variety of tasks they perform at work, the engagement with co-workers, the technical confidence to accomplish their deeds, and the monetary rewards and incentives.

- As it happens with the Engagement, the Concentration gathers attitudes observed before and during the execution of a task. Concentration is influenced by the need that software engineers have

to recognize what exactly is their role into a team, allowing them to perform a well-defined work at the organization. Concentration can be affected by the cognitive workload resulting from the overload of tasks from different knowledge domains, and it receives some influence from the physical elements, such as work environment, artefacts, and tools.

- Collaboration represents helping others not as an obligation, but as a free will. The factors that influence the collaboration are communication and participation, which stimulate a sociable and friendly environment; it is also affected by the exchange of knowledge among team members, the interdependence (a necessary condition to cooperation), and also the team competence, which promotes technical balance among the group.

- Happiness is influenced by performance and feedback. When software engineers successfully perform their work, they tend to achieve happiness in the workplace. In addition, the feedback about their results and about the impacts of their work directly influences their happiness. However, it is important to state that, following initial theories (LOCKE, 1969; HACKMAN et al., 1975), motivation happens before the action, while satisfaction happens afterwards. Thus, França (2014) concluded that the happiness commented by the interviewees was a sign of satisfaction to work and not a sign of motivation.

Table 3. Summary of the antecedents of motivation (França, 2014)

Motivated behaviour trait	Antecedents	Case I	Case II	Case III	Case IV
Engagement	Monetary Rewards	✓	✓	✓	✓
	Acquisition of useful knowledge	✓*	✓*	✓*	✓*
	Social Impact	✓*	✓	✓	✓*
	Work variety	✓	✓*	✓	✓
	Creative work	✓*	✓*	✓*	✓*
	Engagement of co-workers	✓*	✓*	✓	✓
	Technical confidence	✓	✓	✓*	✓*
Concentration	Well defined work	✓*	✓	✓	✓*
	Workload and pressure	✓*	✓	✓	✓
	Artefacts and tools	✓			✓
	Work environment		✓	✓	✓
Collaboration	Communication and participation		✓*	✓*	✓
	Knowledge exchange		✓	✓*	✓
	Interdependence			✓	✓
	Team competence	✓	✓		✓
Happiness	Performance	✓*	✓*	✓*	✓*
	Feedback	✓	✓*	✓*	✓

✓ - mentioned, but not representative
✓* - representative for the case study

The final model proposed by França (2014) integrates the aspects and relationships observed in the cross-case analysis and in the initial theories. These relationships are moderated by individual characteristics of software engineers. The TMS-SE suggests that work motivation and job satisfaction are strongly connected. However, there are fundamental characteristics that differentiate the concepts. Motivation refers to the desire of software engineers to work. On the other hand, job satisfaction refers to pleasurable emotions of software engineers in reaction to their job. Motivation is future oriented, while satisfaction is past oriented. It makes motivation an antecedent of performance, while satisfaction is a consequence of work events. Finally, motivation impacts the productivity of software engineers, while job satisfaction impacts attendance, turnover, and health (FRANÇA; SHARP; DA SILVA, 2014).

TMS-SE provided a theoretical framework with observable traits of motivated, not motivated, demotivated, and satisfied software engineers and pointed out how important it is, to practitioners, the correct understanding of work motivation and job satisfaction as two distinct phenomena, with different antecedents, behavioural signs, and outcomes (FRANÇA, 2014). **Erro! Fonte de referência não encontrada.** Figure 1 summarizes TMS-SE and the topics below were directly transcribed from França's work.

Figure 1. A Model of Work Motivation and Job Satisfaction of software engineers (França, 2014)

(a) Software engineers value co-workers' engagement, so their perception that co-workers are engaged positively influences their engagement;
(b) Software engineers value technical confidence, so their perception that they are technically confidend positively influences their motivation to work;
(c) Software engineers value social impact of the work, so their perception that the work have social impact positively influences their motivation to work;
(d) Software engineers value the acquisition of useful knowledge, so their perception that they will acquire useful knowledge positively influences their own motivation to work;
(e) Software engineers value work variety, so their perception that a work is variable positively influences their motivation to work;
(f) Software engineers value creativity, so their perception that a work demand creativity influences their motivation to work;
(g) Software engineers value well defined work, so their perception of how well the work is defined positively influences their concentration;

(h) Software engineers value cognitive workload balance, so their perception of cognitive overload negatively influences their concentration;
(i) Motivated engineers are engaged while demotivated engineers are distracted.
(j) An individual's level of work motivation positively influences his/her own individual performance at work;
(k) An individual's level of work motivation positively influence his/her collaborative performance at work, proportionally to the level of communication and participation;
(l) Software engineers' discrepancy-value perceptions of the organizational context account for their happiness;
(m) Software engineers' discrepancy-value perceptions of the agents account for their happiness;
(n) Software engineers' discrepancy-value perceptions of their informed level of performance, based on the available feedback, influence the software engineers' appraisal of the workplace factors;
(o) Job satisfaction influences attendance, continuance intention and health;
(p) (q) (r). Individual characteristics influences the software engineers appraisal of the workplace factors before and after the actions.

2.2 Job Rotation

Since the early 1990's, research on Human Resource Management have investigated practices and approaches to improve performance of individuals at work (OLLO-LOPEZ; BAYO-MORIONES; LARRAZA-KINTANA, 2010). These studies, focused on employees rather than in the organization as a whole, and regard the experience of workers and the individual work practices as the main

elements to achieve high-performance and to increase the interest and attractiveness of a job.

Job rotation arises as one of these practices to be applied at the organizational environment in order to increase task variety, and designated to reduce the monotony, boredom and fatigue, resultant from job simplification, specialization and repetition (HSIEH; CHAO, 2004). Nowadays, the practice of job rotation has being explored in several areas and for this reason the literature provides a wide variety of definitions (COYNE, 2011). The concept of job rotation proposed by Woods (1995) is the main definition used to guide this study because the dynamics of a software team frequently concerns the rotation of members among different software projects within the organization and less frequently among different types of jobs.

Following Woods (1995), job rotation is the systematic movement of employees from job to job or project to project, within an organization, as a way to achieve various different human resources objectives. Complementary, other definitions to job rotation can be mentioned for completeness sake:

- Coyne (2011) describes job rotation as the purposeful and organised movement of staff within and across organizations to enhance both the success of the organization and the employability of staff.
- Diego-Mas, Asensio-Cuesta et al. (2009) state that the rotation of people means that workers are suitably trained to carry out different tasks, giving the company greater flexibility in assigning employees to different jobs.
- Kuijer, Vries et al. (2004) suggest that job rotation is a regular alternation between different jobs within an organization, based on a scheme or spontaneously on the basis of the workers' personal needs.

- Richardson, Douglas et al. (2003) define job rotation as a reciprocal exchange of staff between two or more areas for a predetermined period of time.

Besides the fact that job rotation has several definitions, typically depending on the knowledge area in which the phenomenon is being investigated, the literature presents different and sometimes conflicting results about the impacts of this practice both to the employees and to the organization.

Campion, Cheraskin, and Stevens (1994) studied the process of job rotation and identified potential costs and negative impacts, such as the increase of workload and decrease of productivity, the increase of costs with learning process and the decreased of satisfaction and motivation, resulting in job burnout. On the other hand, Richardson, Douglas et al. (2003) identified a set of benefits obtained by the use of job rotation, e.g., improvement of knowledge and skills, promotion and development of opportunities, networking, ability to recruit and retain staff, improvement of interdepartmental relationships, and increase of motivation.

Based on these two studies, it is possible to infer that there are competing views about the way job rotation affects aspects such as motivation, satisfaction or job burnout and for this reason several research were conducted in different study areas to investigate the real impact of this practice. For example:

- Kaymaz (2010) presented how the motivational effect resulting from the practice of job rotation can improve individual performance at work.
- Ho et al. (2009) investigated the effects of rotation in stimulating satisfaction of nurses at work.
- McGillicuddy (2007) suggested that job rotation amongst Information Technology professionals and companies could be highly useful, especially to bring staff from business and to promote business alignment.

- Weerd-Nederhof et al. (2002) studied the use of this practice as a tool to promote innovation.

Many other examples can be found in the literature in several sciences, however, in the specific context of software engineering, few studies have discussed the effects and impacts of job rotation on software teams, as will be discussed in the following section.

2.3 Job Rotation in the Software Engineering Field

Considering the specific context of software engineering, Faegri, Dyba and Dingsøyr (2010) described job rotation as a broadly known approach to increase knowledge redundancy, but emphasized the lack of empirical evidence about the introduction and adoption of this practice in software development. Their study aimed to explore benefits and challenges to improve knowledge redundancy among developers using job rotation.

By applying an action research approach, during a period of eighteen weeks, they observed and collected data from nine developers being rotated from their job to customer support activities. The results suggests that although there were strong indications of increasing knowledge redundancy, the benefits obtained were not sufficient to justify the practice of rotation of developers, is this case, regarding learning about different products in the customer support department (FAEGRI; DYBA; DINGSØYR, 2010).

A set of studies relating job rotation practices with software engineering were also identified. However, differently from the research performed by Faegri, Dyba and Dingsøyr (2010), the main goal of these studies were not to investigate aspects around the practise of job rotation. The researchers only commented

informations about the influence of this practice on members of software development teams observed during their studies:

- The study of Passos, Cruzes and Mendonça (2013) observed that job rotation might be used to increase team flexibility to deal with software quality issues.
- Larsson, Wall et al. (2006) perceived the potential use of job rotation in technology transfer process. However, they only mentioned the possible use of the practice in this context. No specific discussion was done.
- Cohen, Birkin, et al. (2004) found that job rotation can be successfully applied in several areas to improve employees' understanding of various aspects within an organization. But, this practice might not be appropriate among software testing and development teams.
- Birk et al. (2003) argued that, typically, organizations can develop communication processes using job rotation and it can be applied to software development companies. The researchers encourage the establishment of mechanisms for rotation to improve communication between team members.
- Rus and Lindvall (2002) point out that while some software development practices, such as pair programming, facilitates knowledge sharing among pairs of developers, job rotation can be useful to spread knowledge throughout the project or organization.

This literature review highlighted the problems with these studies reporting aspects of job rotation in the context of software engineering. First, the study performed by Faegri, Dyba and Dingsøyr (2010), the only study in which job rotation is the main focus, showed strong aversion of the developers to this practice, who did not see any significant benefits of being rotated from they job to another, even for a short period of time.

In this case, this aversion happened due the fact that job rotation was investigated in the scenario described by Woods (1995) as job to job. In other words, the developers were rotated to a different department at the organization to perform a different job. Our research aims to investigate this practice by observing the rotation of software engineers not to a different department at the organization, but to a different software project (project to project).

Furthermore, the remainder of collected studies did not actually investigate the practice of job rotation. Their observations about this theme occasionally appeared from the data analysed and from the results obtained. Thus, the empirical validity about job rotation in these reports are weak. In our research, an empirical study was planned and conducted to investigate aspects directly related to this practice in the context of software engineering.

2.4 Summary

Chapter 2 presented the theories and concepts that underlie this research, and also, other studies related to the theme. First, Section 2.1 presented details about the Theory of Motivation and Satisfaction of Software Engineers, introducing the elements observed to construct this theory and the model derived from the findings. The TMS-SE reinforces the importance of the distinction between work motivation and job satisfaction, establishes that engagement and concentration are the key elements of work motivation and pointed out that this motivation improves

the satisfaction, that is moderated by feedback information provided about the individual's performance.

Posteriorly, Section 2.2 introduced the concept of Job Rotation, a practice that emerged in 1990s and suggests the rotation of work and tasks. Through this practice, individuals are systematically moved among jobs or projects, within the organization, in order to achieve various goals, especially those related to task variety and spread of knowledge. This section discussed several findings of researchers about the advantages and the disadvantages of job rotation in diverse areas and scenarios over the years, including studies related to motivation and satisfaction of individuals. However, in the specific context of software engineering, only few studies have discussed the effects and impacts of job rotation on software engineering.

Finally, Section 2.3 discussed the research of Faegri, Dyba and Dingsøyr (2010) that focused specifically on job rotation in the context of software engineering and that characterized the practice as a broadly known approach to increase knowledge redundancy in the software development environment. A set of studies relating job rotation practices with software engineering were also identified, however, this practice was not the central goal of research and researchers only superficially commented results about the influence of this practice in software development environments.

Chapter 3 Methods

This chapter describes the methodological steps followed to achieve the research objectives of this work: to investigate how job rotation can affect motivation and satisfaction of software engineers, and in the light of the TMS-SE analyse which aspects of the motivation and/or satisfaction are affected, considering the frequent movement of software engineers among different software projects within an organization.

The nature of the phenomenon under study and the type of variables involved in this inquiry required a qualitative research method to explore and interpret the problem. Merriam (2009) argues that qualitative researchers are interested in the understanding that people have constructed along their experiences about the phenomenon under study. According to Hazzan (2006), the qualitative research is usually used to investigate social phenomena, that is, situations in which people are involved and different types of experiences are being observed. Further, Merriam (2009) defines four main characteristics to describe a qualitative research:

- Qualitative research seeks to understand the meaning attributed to individuals' experiences. This research is mainly used to study individuals' understanding about some phenomenon.
- Qualitative research takes the researcher as an important part of the process. The researcher cannot completely separate himself/herself from the topic or people under study, once the knowledge is created from the interaction between them. Thus, inevitable biases and their impact on data collection and analysis must be monitored.

- Qualitative research regards an inductive process to gather evidence in order to establish theories and hypotheses.
- Qualitative research provides essentially descriptive data in the form of words and pictures, rather than the numbers commonly produced by other types of research.

Due to the variety of human and social aspects around the software engineering context, research in this area frequently addresses questions and problems involving human elements that are complex to interpret (SEAMAN, 2013). Qualitative methods can be helpful in handling this complexity, that is also existent in the present work, in which better understanding about the impacts of job rotation on the motivation and satisfaction of software engineers can be investigated by the truthful reporting of experiences and perceptions of a group of individuals about this practice.

A qualitative case study was performed to reach the goals of this study. A case study is “an empirical inquiry that investigates a contemporary phenomenon within a real-life context” (YIN, 2003), and this approach uses multiple sources of evidence to explore the relation between the phenomenon and a context. Case studies have been used in diverse research areas (YIN, 2003), in software engineering, for example, this method may be suitable to explore the complex interactions among people and technologies (RUNESON; HÖST, 2009). Moreover, the social and human centred activities present in the practical software engineering issues, should be investigated in their natural settings to achieve better understandings (EASTERBROOK, SINGER, et al., 2008) and the case study is a method that allows this investigation.

According to Yin (2003), a case study can be performed in different ways depending on the context being studied and the number of cases (analysis units). The guideline for conducting and reporting case studies in software engineering

proposed by Runeson and Höst (2009) suggests five major process steps to be implemented:

- Firstly, the study design is defined, the objectives are shaped and the case study is planned.
- The second step is the preparation for data collection. This is the phase in which procedures and protocols for data collection are defined.
- Following this, the step of collecting evidence means to collect data from the case under study.
- Posteriorly, the collected data are analyzed.
- Finally, researchers prepare the reporting of their findings.

Section 3.1 presents the detailed steps implemented to perform this research work.

3.1 Case Study Design

This case study was performed by following five main steps: Research Planning, Data Collection, Data Analysis, Report, and Member Checking (including the review of the final version of the report). The whole process is summarized in Figure 2. These steps were inspired by the five phases to conduct case studies in software engineering proposed by Runeson and Höst (2009) and also by the process provided by Eisenhardt (1989) to build theories from case study research. These steps are summarized as follows:

- Research Planning, step in which the scope of the study was defined.
- Data Collection, step in which instruments of data collection were applied to gather information from the participants of the study;

- Data Analysis, step in which data collected were analysed in order to understand the knowledge included in the participants' narratives.
- Research Report, step in which the understanding and considerations about the phenomenon observed were described in a written report.
- Member Checking, step in which the participants were consulted about whether the interpretations of the data collected reflects their actual experience. Results from this step were used to refine the Research Report.

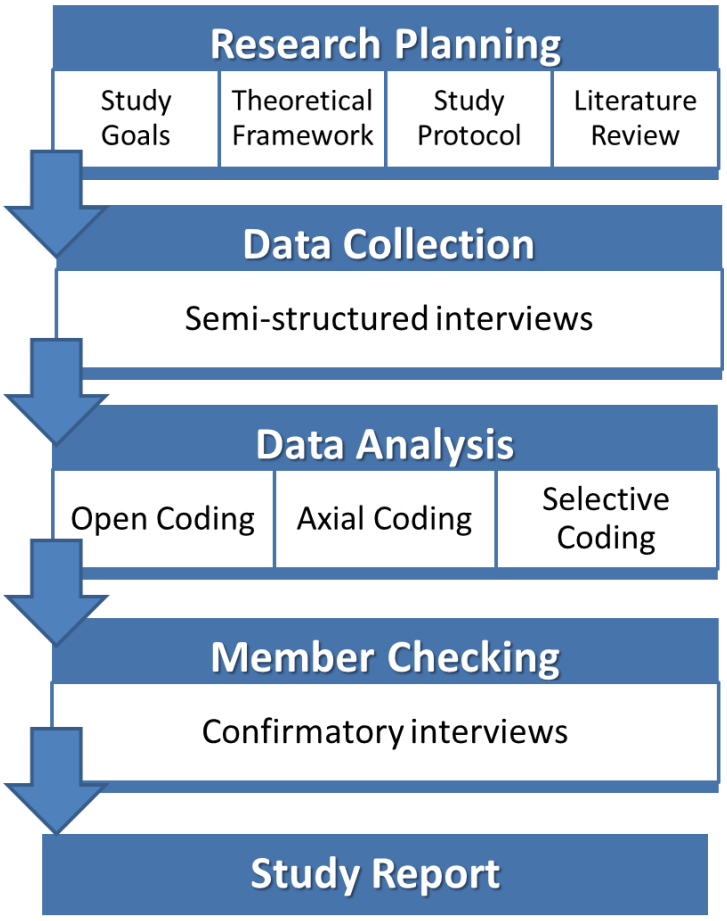

Figure 2. Case study design

Research Planning

During the research planning, the goal of the study and the theoretical framework were defined. Semi-structured interviews were chosen as the main instrument for data collection and a research protocol was designed to register and guide the process during the next phases. The Theory of Work Motivation and Job Satisfaction of Software Engineers proposed by França (2014) was chosen as the initial framework of this research to provide guidance for designing this study and for collecting relevant data, once this theory emphasizes the distinction between work motivation and job satisfaction and provides a well defined framework around these phenomena in the context of software engineering.

Further, a literature review on job rotation was performed in order to identify three types of information: (i) theories and definitions about job rotation to complement the design of this study, considering that the initial framework only encompasses issues around motivation and satisfaction; (ii) studies regarding the impacts of job rotation in several areas; (iii) previous studies investigating job rotation in the context of software engineering. Further, according to Merriam (2009) the literature review is important because data analysis may involve comparisons to significant literature to emphasize the relevance of the findings.

The case selected to be studied was a private software organization located in Recife, Brazil, in which members of software teams are frequently rotated amongst software projects, depending on the business needs. This software development company is specialized on system development, in which the main business is related to the development of innovative software solutions. This organization was founded in 1996 and is based in Recife (Brazil), but it has four more offices in four other cities in the country where about 500 employees are working. The projects developed at this organization are spread over several

domains, including finance, telecommunication, government, industry, services, and energy.

Monteiro (2014) observed the same organization in his study about innovate behaviour and leadership, and described the organization in terms of a tipical vertical structure, as presented in Figure 3, in which the software development unit is called *Advanced Engineering*. This Advanced Engineering area of the organization is composed by designers, system administrators, system analysts, software engineers, software testers, database administrators, project managers, and the Chief Operation Officer (COO). Besides, there are four organizational levels on the top of this area and eight areas on the same organizational level, not directly related to the goal of this study.

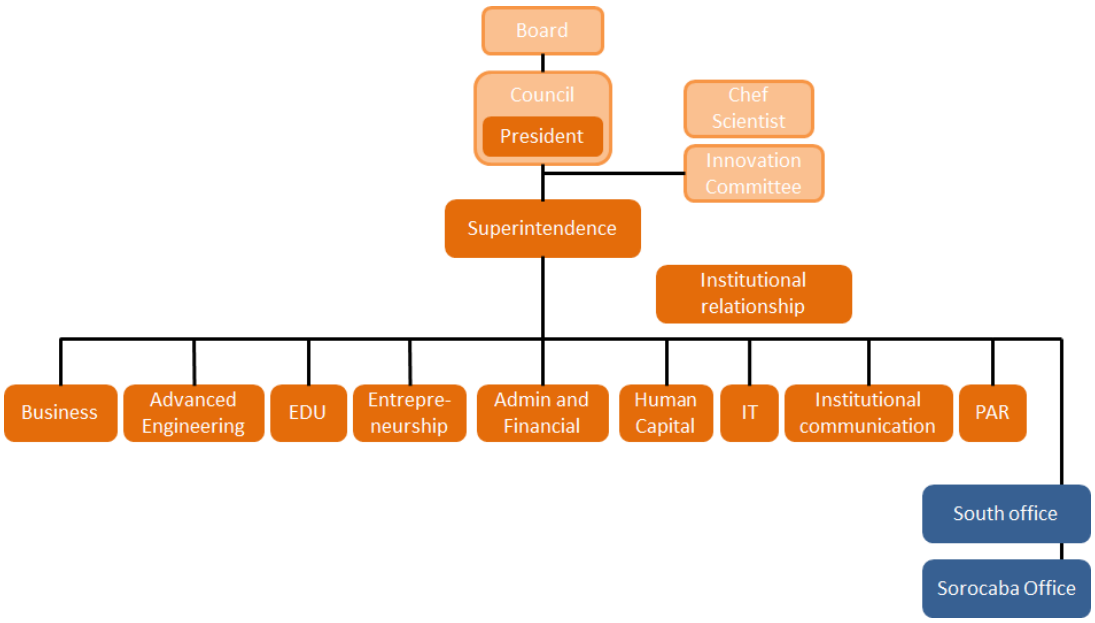

Figure 3. Organizational structure of the company (Monteiro, 2014)

To better understand the type of job rotation applied at this organization, a preliminary interview was conducted with the high level management of the company about the organizational context. Later, a group of software engineers were selected to comment about their experience with these rotations.

Data Collection

There is a set of instruments that can be used to collect qualitative data to case studies. Merriam (2009) describes interviews, observations, and archival sources as instruments that are most often used. This work used semi-structured interviews to collect data, a type of interview more flexible in which questions are planned, but they are not necessarily asked in the same order as they are listed and the improvisation allows better investigation of relevant topics during the course of the interview (RUNESON; HÖST, 2009).

According to Merriam (2009) interviews are an effective approach to discover information about elements difficult to observe and measure, such as feelings, thoughts, and intentions. The complete interview script used in this research (Appendix A) was composed of questions designed to explore experiences, opinions, values, feelings, knowledge, and background of the participants. Questions were defined following the arguments of Patton (1989) about the six types of interviews questions, also commented by Merriam (2009):

- Experience and behaviour contemplates questions that regard interviewees' experiences, actions, or behaviours.
- Opinion and values questions, which concern the interviewees' opinions or beliefs regarding a certain topic or event.
- Feeling questions, which require from interviewees a reflection on their feelings about the phenomenon.

- Knowledge questions, the ones that ask interviewees to recall specific factual information.
- Sensory questions, similar with experience questions, but used to elicit information regarding what the interviewee has seen, heard, touched, etc.
- Demographic questions are used to elicit demographic information, such as the interviewee's age, education, background, etc:

Before the actual interviews start, pilot interviews were conducted with a group of five people who had experience with being rotated amongst software projects. This strategy is important to refine the questions defined previously and helps to prepare the interviewer for the real data collecting (MERRIAM, 2009).

The actual interviews happened individually, at the organizational environment, with audio recorder, performed by an interviewer and a scribbler. Each interview lasted approximately 40 minutes. The interviewees were team members of two different projects, one with a great incidence of rotations and other one with a more static team. This distinction between the two projects allowed the researcher to collect information from distinct types of experiences with job rotation, both those participants who were rotated before and those who just observed the rotations indirectly. The profile of participants is present below:

- **PA01NR**: Software engineer, Male, 22 years old, undergraduate student, 3 years working at the organization and never rotated.
- **PA02NR**: Software engineer (trainee), Male, 23 years old, undergraduate student, 1 year and 3 months working at the organization and never rotated.
- **PA03RO**: Test engineer, Male, 30 years old, technical degree in software development, 3 years working at the organization and rotated once before.
- **PA04NR**: Software engineer, Male, 24 years old, undergraduate student, 3 years working at the organization and never rotated.

- **PA05RO**: Software engineer, Male, 26 years old, B.Sc. in Computer Science, 3 years working at the organization and rotated once before.
- **PA06RO**: Software team leader, Male, 30 years old, M.Sc. in Computer Science, 6 years working at the organization and rotated three times (mostly at ramp downs).
- **PA07RO**: Software project manager, Female, 43 years old, B.Sc. in Computer Science, 6 years working at the organization and rotated three times.
- **PB01NR**: Software engineer, Male, 28 years old, B.Sc. in Computer Science, 3 years working at the organization and never rotated.
- **PB02RO**: Test engineer, Male, 34 years old, undergraduate student, 1 year and 9 months working at the organization and rotated once before.
- **PB03RO**: Software engineer, Male, 23 years old, B.Sc. in Computer Science, 3 years working at the organization and rotated once before (just for two weeks).
- **PB04RO**: Software engineer, Male, 32 years old, M.Sc. in Computer Science, 3 years working at the organization and rotated once before (just for one week).
- **PB05RO**: Software engineer, Male, 26 years old, B.Sc. in Computer Science, 2 years working at the organization and rotated twice.
- **PB06RO**: Software team leader, Male, 36 years old, M.Sc. in Computer Science, 12 years working at the organization and rotated several times over the years.
- **PB07RO**: Software project manager, Female, 39 years old, M.Sc. in Computer Science, 16 years working at the organization and rotated several times over the years.

During the data collection, as Runeson and Höst (2009) recommend, first the objectives of the interview and the case study were presented to the participants. Then, a set of introductory questions were asked about their background and the organization as whole. After this, the main interview questions focused on the phenomenon under study assumed the longest part of the meeting. Finally, as Merriam (2009) suggested, the initial data analysis began right after the first interview was finished, to guarantee a continuous and progressive qualitative process.

Seven main questions guided the interviews aiming to obtain data about the experience of participants with the practice of job rotation among software projects at the organization. These questions are presented below, translated from the original questions from the interviews script (Appendix A) which was constructed and applied in Portuguese language:

- What do you think of the practice of rotation, applied at the organization, that normally moves people from a software project to another?
- How do you feel about the possibility of being rotated to another team while a software project is running?
- Have you ever been rotated to another project that was already in progress? Tell me about this experience.
- Was there any rotation of people in the teams that you are or were member? Tell me about this experience.
- What would you do if the organization asks you to change to another project right now? What would you consider to make this decision?
- Imagine if you were rotated to another project right now, how this would impact your current team?
- What would be the impact of your rotation to another team? Tell me about the impact not directly related to the project itself but to the team.

Data Analysis

The data analysis process in qualitative research involves labelling and coding all of the data in order to identify similarities and differences to describe the phenomenon under observation. The objective of qualitative analysis is to consolidate, reduce, and interpret data obtained from various sources, and make sense of them (FRANÇA, 2014). Strauss and Corbin (1990) define qualitative analysis as “a non-mathematical process of interpretation, carried out for the purpose of discovering concepts and relationships in raw data and then organizing these into a theoretical explanatory scheme”.

Processing qualitative data often begins during the early stages of data collection. According to Merriam (2009), it is recommended performing data collection, data analysis, and reporting simultaneously, because ongoing findings can affect the data being collected, and how they are collected, in order to obtain better performance from the researcher and richer results at the end. Usually, qualitative data analysis includes: (i) coding and categorization of the information obtained at the data collection step (interviews, observation, field notes, etc.); (ii) definition of concepts to characterize the defined categories; (iii) linking and combination of these concepts; (iv) elaboration and report of a scheme from the emerging understanding obtained (STRAUSS; CORBIN, 2008).

Frequently, the data analysis process is conducted by applying the coding approach defined by Strauss and Corbin (2008) to construct grounded theory, in which the collected data are submitted to three phases of coding:

- Open Coding: According to Strauss and Corbin (2008) this is the moment when “the data are broken down into discrete parts, closely examined, compared for similarities and differences, and questions are asked about the phenomenon reflected in the data”. In other words, each line, sentence and

paragraph is analyzed in order to identify the concepts, their properties and dimensions to define codes, that is, to label chunks of data that posteriorly are grouped into representative categories.

- Axial Coding: This phase consists in intense analysis around the defined categories in order to find relationships by making connections between a category and its subcategories or other categories (STRAUSS; CORBIN, 2008). Axial coding is an inductive and deductive process, focused in emphasizing causal relationships amongst data.
- Selective Coding: The final stage of data analysis is to set the main story underlined in the investigation through the identification of the core category that gathers all the data analyzed. Strauss and Corbin (2008) proper define selective coding as "the process of selecting the central or core category, systematically relating it to other categories". Thus, through selective coding, the categories are integrated and developed into a theory.

This research used *in vivo* codes in the phase of open coding, which is a colloquial interpretation of the phenomena under observation, obtained directly from the interviews data (BANDEIRA-DE-MELLO; CUNHA, 2003). In this specific case, it means that a chunk of text, coming from the transcribed interviews was underlined. This chunk represented one of the antecedents of motivation and satisfaction (Table 4), among those defined by França (2014), or a new aspect discovered from further analysis. Posteriorly, the axial coding identified relationships among the categories and the selective coding built the story of the case, that is, a report describing the meaning behind the connection of the categories, producing a discursive set of propositions.

Table 4. Examples of open coding in vivo

What do you think of the practice of rotation, applied at the organization, that normally moves people from a software project to another?
Work Variety { PA02NR, PA04NR, PA05RO, PA06RO, PA07RO} *"Good, especially when this rotation allows you to work with a diferente technology." (PA04NR)*
Acquisition of useful knowledge {PA02NR, PA03RO, PA04NR, PA06RO} *"(...) You have the opportunity to learn several things." (PA06RO)*
Feedback {PA04NR} *"(...) There are some guys I know that have to wait too long to get a good feedback because they are always switching projects." (PA04NR)*
Performance {PA01NR} *"(...) So the Project progress gets a little slow until we can reorganize our activities again" (PA01NR)*

The data analysis resulted in an initial report containing all the interpretations of the data collected during the interviews, which posteriorly was used to construct the set of observations and the conclusions of this research. This initial report was the main component used in the next research stage: the member checking.

Member Checking

The Member checking is a quality control process, largely associated with qualitative research, applied to improve accuracy, credibility, and validity of what was interpreted form the qualitative data collected in the interviews, diaries or observations (HARPER; COLE, 2012). The effort for this step is to measure how well the researcher understood the participants' viewpoints about the phenomenon under study and whether the conclusions are representative and complete (KREFTING, 1991), commonly using two approaches: (i) a second interview with individuals who participated of the research; (ii) an interview with a small group of

individuals similar to those who participated before. Both cases require that researchers to conduct an interactive presentation of their findings.

This research applied the first approach, in which a group of individuals formally interviewed before participated of two confirmatory interviews, seeking to verify: (i) the accuracy of the findings by confirming or disconfirming the interpretations; (ii) the representativeness and completeness of conclusions by indicating the frequency in which the aspects identified happened. For both confirmatory interviews, a qualitative scale was constructed to measure the impact of the responses, as presented in Table 5.

A set of individuals with distinct point-of-views about the phenomenon under observation was selected to the member checking phase, based on their previous answers, in order to confirm (or disconfirm) the data interpreted. In this sense, the confirmatory interviews were performed with participants PA04NR, PA06RO, PB02RO, PB06RO and PB07RO, who commented (individually) about the observations derived from the interpreted data. These interviewees were chosen to participate of member checking because they had different experiences with rotations and so, they could comment about the elements observed from distinct viewpoints. The participation of interviewees at this phase of research was limited for the availability of the organization to release the participants to the second interview. Besides, there were a group of interviewees who were no longer working for the organization during the member checking.

During the member checking, the interviews also occurred at the organization’s office, lasting no more than 15 minutes. The complete interview script used in this phase is presented in Appendix B.

Table 5. Examples of Member Checking interview (presented in Portuguese)

Work Variety	
Movimentações internas permitem que os engenheiros de software possam trabalhar com diferentes projetos e tecnologias, ocasionando variedade de trabalho.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
As movimentações internas trazem oportunidade de trabalhar com um novo projeto ou tecnologia.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Feedback	
Movimentações internas afetam o feedback do trabalho, uma vez que uma pessoa pode não passar tempo suficiente em uma equipe para serem avaliado.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
As movimentações internas prejudicam o recebimento de feedback sobre o trabalho.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca

Research Report

At this stage of research, the findings of the case study need to be carried to closure (YIN, 2003). According to Robson (2002), reporting case studies is a process that: (i) summarizes what the study was about; (ii) communicates a clear sense of the case studied; (iii) provides a “history of the inquiry”, so the reader can understand what was done, by whom and how; (iv) affords basic data in focused form, so the reader can make sure that the conclusions are reasonable; (v) articulates the researcher’s conclusions and set them into the context they affect.

The report of this research included all the relevant snapshots obtained from the data to support the conclusions, including citations, narratives, anonymous quotations of participants, research instruments and details of the procedure, as Runeson and Höst (2009) suggests, in order to lead the reader to understand the found evidence and also allow researchers to perform replications and further investigations in similar contexts.

3.2 Summary

A qualitative case study was performed to reach the goals of this study: to investigate how job rotation can affect motivation and satisfaction of software engineers. This case study was performed by following five main steps: (i) Research Planning: the scope of the study was defined. (ii) Data Collection: the instruments of data collection were applied to gather information from the participants; (iii) Data Analysis: the participants' narratives were analysed and the data were interpreted; (iv) Research Report: the understanding and considerations about the phenomenon observed were reported; (v) Member Checking: the participants were consulted about whether the interpretations of the data collected reflected their actual experience.

The case study took place at a private software organization located in Recife, Brazil, in which members of software teams are regularly rotated amongst software projects, depending on the business needs. A semi-structured questionnaire guided the interviews and participants commented their personal experience with rotations and/or their point-of-view based on their perception of the effects when people of the team were rotated.

The data analysis involved the process of labelling and coding all of the data obtained in the interviews in order to identify similarities and differences to describe the phenomenon under observation. The final report included all the relevant snapshots obtained from the data to support the conclusions, including citations and narratives.

At the end, a set of individuals with distinct point-of-views about the phenomenon under observation was selected to the process of member checking and participated of two confirmatory interviews, seeking to verify the accuracy of the interpretations and the representativeness and completeness of conclusions.

Chapter 4 Case Study Results

This chapter presents the findings of the case study performed with members of two projects, among those in progress at the organization described in Section 3.1. Projects were chosen according to the organization's availability to release the members to participate of this study. The first software project (Project A) was working on the development of a new web system for a logistics company, and the second software project (Project B) was focused on the development of a 3D visualizer for a company that produces printers.

The group of individuals that participated of this study was composed by software engineers, software engineering trainees, programmers, test engineers, software team leaders, and software project managers. All professionals are referenced here as software engineers, as it is a practice of the organization. Table 6 presents the summarized information about the group of participants of this case study, which was already detailed in Section 3.1.

Table 6. Participants of the case study

Project	Participants	Code	Member Check
Project A	Software Engineer	PA01NR	N
		PA04NR	Y
		PA05RO	N
	Software Engineering Trainee	PA002	N
	Test Engineer	PA03RO	N
	Software Team Leader	PA06RO	Y
	Software Project Manager	PA07RO	N
Project B	Software Engineer	PB01NR	N
		PB03RO	N
		PB04RO	N
		PB05RO	N
	Test Engineer	PB02RO	Y
	Software Team Leader	PB06RO	Y
	Software Project Manager	PB07RO	Y

4.1 Antecedents of Motivation and Job Satisfaction Influenced by Job Rotation

The analysis of collected data shows that the practice of job rotation can influence six factors that impacts motivation and job satisfaction of software engineers among those described on the TMS-SE:

(i) antecedents of motivation affected by job rotation: work variety, acquisition of useful knowledge, cognitive overload, and well defined work.

(ii) antecedents of job satisfaction affected by job rotation: performance and feedback.

Furthermore, we identified a new factor that can indirectly influence both motivation and satisfaction: Knowledge Transfer. This factor is similar to the knowledge exchange observed in the TMS-SE. The main difference is between this two factors is that in TMS-SE a knowledge exchange is characterized as a bi-directional exchange process. On the other hand, in our study, the knowledge transfer happens in a unilateral way, in which an individual (that was rotated to a new project) gives knowledge (about his/her prior project) and another individual (in the prior project) only receives this information. The rotated individual does not receive any new knowledge in the process.

Figure 4 presents the scale of representativity of each factor that influences motivation identified through the analysis of the interviews and Figure 5 presents this scale considering the satisfaction of software engineers. This representativity is based on the frequency and intensity that these elements appear in the interviewees dialog. The representativity of these factors was checked and reviewed during the member checking process.

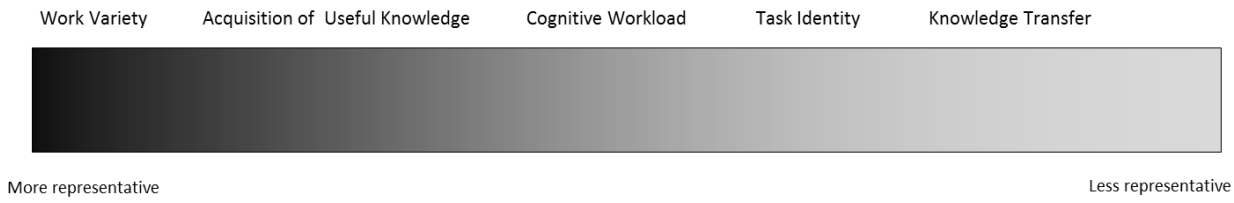

Figure 4. Identified factors that influence motivation of software engineers

Figure 5. Identified factors that influence satisfaction of software engineers

The practice of job rotation in software development organizations can produce both positive and negative effects. Regarding motivation, it can be positive due to the increase of software engineers' engagement, but negative because it might also impair the concentration of these individuals. Besides, analysis demonstrated that job rotation can decrease job satisfaction, unless proper strategies of planning and management are adopted to overcome problems related to performance and feedback of rotated individuals.

The effects of the practice of job rotation associated to the motivation and satisfaction of software engineers observed through the analysis of inteviews are detailed below.

Influence of Job Rotation in the Motivation of Software Engineers

Observation #1: Job rotation enhances the variety of work. Job Rotation gives software engineers the possibility to work with a wide variety of

projects and technologies, as demonstrated through the quotations of interviewees in Table 7. This aspect produces a positive impact on their engagement and so on their motivation to work. However, this variety depends on how many software projects the organization are executing in a given time.

Table 7. Interviewees' quotes about work variety

Work Variety	
"(...) lidar com novas tecnologias ou com outras tecnologias. Eu acho ótimo!" *English: "(…) to handle with new technologies or different technologies. I think it is great!"* (PA02NR)	***"(...) Quando você tem vontade de trabalhar com determinada tecnologia e você sai de um projeto e vai pra outro que tem aquela tecnologia então você com certeza acha bom."*** *English: "(…) When you want to work with some technology and you come out of a project and go to another that uses that other technology, then you think it is good, for sure." (PA05RO)*
"(...) É muito bom porque o cara vai trabalhar com várias coisas diferentes." *English: " (...) It is very good because one is going to work with several different stuff" (PB04RO)*	***"Faz com que tragam coisas novas (...) Eu não consigo parar num único projeto. "*** *English: "It brings new stuff (...) I can't stay in the same project for long." (PB07RO)*
"Bicho, não tem outro projeto pra mim não? Eu já tô fazendo isso há tanto tempo! Fazer de novo?" *English: "Man, there is no other project for me? I am doing this for a long time. Do I have to do it again?" (PB01NR)*	***Independente se os problemas são complexos, mas eu ainda estou na zona, então quando eu saí eu gostei.*** *English: "Regardless the problems are complex, I am still in the zone, so when I got out I liked it." (PB03RO)*

Observation #2: Job rotation enables acquisition of useful knowledge. When software engineers are rotated amongst different software projects they frequently obtain a relevant amount of knowledge, both technical and practical, increasing their professional experience in several aspects, e.g., technologies, process and tools. Table 8 illustrates the Acquisition of useful knowledge that positively influences the motivation.

Table 8. Interviewees' quotes about Acquisition of useful knowledge

Acquisition of useful knowledge	
"(...) Eu acho bom porque você vai ter, vai saber que vai ter a oportunidade de aprender." *English: "(...) I think it is good because you know you will have the opportunity to learn." (PA02NR)*	***"(...) É bom porque você vai aprender coisas novas."*** *English "(…) It is good because you will learn new stuff" (PA04NR)*
"(...) Se for uma tecnologia que eu tenha interesse em aprender vai ser gratificante, agora se for uma tecnologia que eu não tenho interesse, ai eu vou achar ruim." *English "(...) If it is a technology which I am interested in learning it, it is going to be rewarding, but if it is a technology in which I am not interested in, I am not going to like it." (PA05RO).*	***"Pra mim é legal, eu tô sempre aprendendo coisa nova."*** *"It is nice to me, I am always learning new stuff" (PA06RO)*
"Ao mudar de projeto você muda muitas vezes de tecnologia então é coisa nova que você vai aprendendo." *English: "By switching projects the technologies also change, so you learn new stuff". (PB05RO)*	***(...) Se for uma coisa diferente que eu queira aprender, poderia ser que sim.*** *English: "If I want to learn different stuff, then it could be." (PB02RO)*

Observation #3: Job rotation increases cognitive overload. The movement among software projects demands higher levels of mental effort from individuals, because too much information needs to be learned and understood in little time from the moment the individual is allocated in a new project. Thus, job rotations increase the possibility of a cognitive overload of software engineers and consequently their concentration, which negatively influences their motivation as demonstrated in Table 9.

Table 9. Interviewees' quotes about Cognitive Overload

Cognitive Overload	
"Veja, bom não é, mas faz parte. (...) No momento em que ele saiu (do projeto) eu conhecia muito pouco do projeto né, então ai eu tive que correr atrás." *English: "Look, it is not good, but it is part of the process. (…) At the moment he left (the project), I knew just a little about the project, so I had to learn."* (PA07RO)	***"(...) É necessário certo tempo trabalhando no projeto pra poder entender como funciona os processos."*** *English: "It is necessary a certain time working in the project to understand how processes work." (PA04NR)*
"Você pegar um projeto andando eu acho complicado (...) se você precisa de um entendimento maior da regra de negócio do outro projeto que você esta lidando." *English: "It is complicated if you enter in a project that is already running (…) If you need an understanding of the business rule of another project you are handling". (PB05RO)*	***"O cara que entra no projeto vai ralar no mínimo, no mínimo, se o cara for desenrolado, um mês, dois meses para poder começar a entender como é que resolver o problema lá dentro."*** *English: The guy who enters in the project will have a lot of work. At least, if the person is smart, it takes a month, or two months to start to understand how you can solve the problems there". (PB04RO)*
"(...) como o projeto é de complexidade media para alta, apenas duas semanas para o cara 'desenrolar' não ia funcionar." *English: As the project is of a medium or high complexity, just a couple of weeks to understand it are not enough". (PB02RO)*	

Observation #4: Job rotation impairs task identity. The concentration of software engineers is commonly related to the process of performing tasks with clear goals, well defined requirements, and predictable results. This aspect is negatively affected by job rotation, in which individuals can be moved to execute a different work in other software project, and not necessarily finishes his/her prior tasks. Table 10 demonstrated how software engineers value well defined work that is a component of task identity and how job rotation can be adverse to this.

Table 10. Interviewees' quotes about Task Identity

Task Identity and Well Defined Work	
"No meio do desenvolvimento eu não sinto muito a vontade, porque eu fico preocupado com o projeto que eu tô atualmente." *English: In the middle of the development I don't feel like doing it, because I stay worried about the project I am working at the moment".* (PA06RO)	***"(...) eu não gosto desta questão de sair e deixar uma coisa inacabada."*** *English: "I don't like this situation of leaving something unfinished".(PA06RO)*
"Não vou dizer que seria bom, porque é bom quando você faz algo e termina." *English "I would not say it would be good. Because it is good when you do something and finish it". (PB02RO)*	***"Porque se eu sair, eu vou ficar tipo: eu vou para outro problema, mas eu ficaria preocupado com aquele outro problema."*** *English: "Because if I leave, I am going to be like: I'm going to other problem, but I would be worried with the previous problem. (PB03RO)*
"(...) no meio do processo sou removido, ai eu ficaria chateado, pois você fica com a sensação de que o projeto ficou pela metade." *English: "At the middle of the process I'm removed, and then I would be mad, because you get that feeling that the project is unfinished". (PB02RO)*	***"Então a única sensação que eu tenho de sair, de fazer essa movimentação agora é saber que não vai ter ninguém pra preencher."*** *English: "So the only feeling I get by being rotated now is that there is no one to fill the gap". (PB04RO)*

Observation #5: Job rotation requires Knowledge Transfer. Often, individuals that are rotated to other software project have to keep in contact with their previous team to transfer information about their prior tasks. This is a situation in which few or none knowledge is exchanged, but only transferred from the individual that has been moved to the new individual responsible for his/her work in the previous software project.

As represented in Table 11, due to this unilateral characteristic, the knowledge transfer, resulting from job rotations, has negative influence on the concentration of software engineers, because it increases the cognitive effort and therefore decreases their motivation. Moreover, knowledge transfer can also decrease individual performance and posteriorly reduce the levels of job satisfaction.

Table 11. Interviewees' quotes about Knowledge Transfer

Knowledge Transfer	
"Você vai tá tutoreando o projeto anterior. Então você vai ter de ajudar aquela pessoa no que você tava fazendo." *English: "You will keep supporting the previous project. So you will have to help that person on what you were doing".* (PB01NR)	***"(...) Qualquer coisa, pode chegar e perguntar 'isso aqui que tu fizesse, como é que foi'?"*** *English: "If it is necessary, you can ask: 'This thing, how did you do it?" (PA02NR)*

Influence of Job Rotation in the Satisfaction of Software Engineers

Observation #6: Job rotation has negative impact on performance. Each job rotation implemented commonly affects the individual performance of at least three individuals: (i) the one that is being rotated; (ii) the member of the new team who will train the newcomer (the individual rotated) in his/her new tasks; (iii) the member of the previous team who will be responsible for the tasks associated to the individual rotated. Table 12 presents quotes of interviewees describing the problems with performance after a job rotation.

This observation can be graphically observed in Figure 6, which demonstrates the three individuals who will have their performance affected by the rotation of a member among the teams. However, it is important to highlight that this decrease of performance is temporary and depends on issues such as type of job, tasks, and the domain of knowledge of the software projects affected by the rotation.

Table 12. Performance

Performance	
"As vezes você pega um projeto que vai te dar mais trabalho, que vai dar mais hora extra, um fim de semana que você vai vir aqui e tal." *English: "Sometimes you enter in a project that is going to be harder, that will bring overtime or a weekend you will have to come here".* (PB05RO)	***"(...) Ai tem de trabalhar mais, mais focado, render mais pra suprir a necessidade daquela pessoa (que saiu)."*** *English: "Then you have to work more, more focused, to supply the lack of that person (who left). (PA01NR)*
"A gente vai ter de preparar a pessoa nova que chegou. Tem de gastar um tempo com isso." *English: "We will have to prepare the new person who just joined us. And it takes some time".(PA06RO)*	***"É sempre traumatizante. Você acaba tendo perda de performance."*** *English: It is always traumatic. You end up having loss of performance". (PB01NR)*
"Vai demorar uns seis meses para ensinar outro cara para alguém assumir o papel que ele tá fazendo." *English: "It's going to take about six months to teach other guy, and then someone can do that task he is doing". (PB06RO)*	

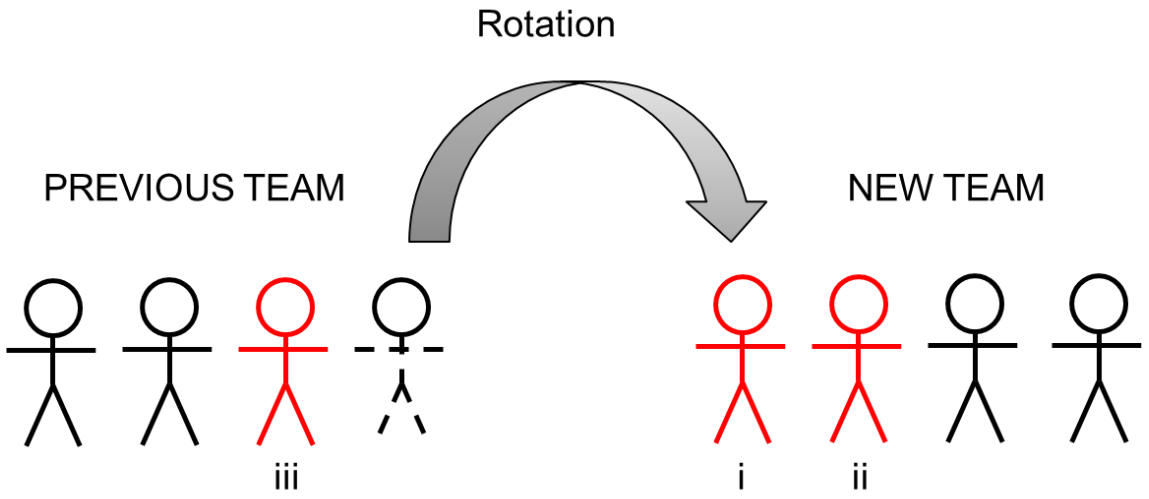

Figure 6. Job rotation affects the individual performance

Observation #7: Job rotation is related to the loss of feedback. Job rotations can cause loss of feedback data about the performance and the work of software engineers. As the individuals are commonly being moved from a software project to another, there may not be time enough to managers and leaders to construct a complete opinion about the individual and his/her work, as stated in Table 13.

Table 13. Interviewees' quotes about Feedback

Knowledge Transfer	
"(...) tem caras que eu conheço que demorou muito para terem um feedback bom porque ficava indo de projeto em projeto" *English: (...) there are some guys who had to wait too much time to get a good feedback because they were switching among projects".*(PA04NR)	***"(...) outro gerente que não conhece nada de você, ele vai ter de começar uma avaliação sua do zero."*** *English: "(…) Other manager that knows nothing about you, he will have to start an evaluation from the start.* *(PA04NR)*

Cross-analysis of Factors that Influence Motivation and Satisfaction of Software Engineers Manifested d*uring Job Rotations*

The analysis pointed out a set of connections among the factors that can inlfuence motivation and satisfaction of software engineers expressed during job rotations. These observations are the first step to construct a list of recommendations to inform managers and leaders about which factors they should be aware when job rotations are being planned and executed.

Observation #8: The work variety resultant from job rotations enhances the acquisition of useful knowledge. The movement of software engineers to a different type of project, working with different types of technologies, allows these professionals to acquire more knowledge than if they were working always in the same project and with the same team. This knowledge will be valuable both to the

current project and to future projects, which contributes to spread and share information within the teams.

Observation #9: The acquisiton of new knowledge necessary to support a job rotation increases the cognitive overload of software engineers. The rotation to a different type of project requires the acquisition of knowledge and the assimilation of a vast set of information in a relatively short time. Consequently this rotation might also demand a great level of mental effort, and thereby, result in cognitive overload of individuals.

Observation #10: Knowledge transfer commonly required after rotations intensify the cognitive effort and impair individual performance. After job rotations, the cognitive overload of software engineers can be increased due to the need for a technical support on his/her former tasks at the previous team, especially considering the process of transferring knowledge to the new individuals responsible for the tasks. Besides, this process can also reduce the performance of the rotated individual who will switch between his/her tasks and the transfer of information.

Observation #11: The work variety resulting from job rotation can negatively impact task identity. Even thought they are enthusiastic with work variety and the possibility of acquire new knowledge, software engineers prefer a well defined work rather then rotation to a different project before finishing their responsibilities.

Observation #12: Management of feedback can reduce the dissatisfaction with temporary low performance. The dissatisfaction of software engineers recently affected by a rotation, resultant from the temporary decrease of performance, can be mitigated by providing regular feedback about their work at the new project.

Other Observations

Data analysis demonstrated divergences among the point of view of participants interviewed, mainly based on two factors. First, whether the participants had been involved in a rotation process or not. Second, what was the individual's role in the team (management, leadership or operation).

Observation #13: Rotated versus not rotated. Software engineers who never participated of job rotations tended to highlight the positive effects of this practice, as work variety and acquisition of useful knowledge, and ignore the cognitive effort required by this process and its possible overload, along with the lack of feedback and the problems with performance. On the other hand, individuals that had participated in at least one rotation tended to be more critical about the negative effects.

Observation #14: The role and the position also matter. Project managers and leaders are not aware of the negative impact of the lack of feedback, which can be increased by job rotations. Further, during the planning of job rotations, they also tend to overlook the negative impact of knowledge transfer required from individuals that were recently rotated and the decisive value of well defined work (Task Identity) to software engineers.

4.2 Job Rotation of Software Engineers and the Need for Balance

Based on these observations it is possible to infer that the practice of job rotation in software organizations demands a balance among the main positive and negative factors that affect motivation and that are manifested during the rotations, in order to avoid a decrease on the levels of performance, and consequently the

satisfaction of software engineers. In this context, work variety, acquisition of useful knowledge, and cognitive overload tend to be proportional. Therewith, the more distinct is the project in which the software engineer will be rotated, the more knowledge they will need to acquire, and thereby, more cognitive effort will be required to learn about this new project and its dynamics in a short period of time.

Thus, considering that the temporary low performance of software engineers after job rotations is inevitable, this balance is the key aspect that managers should be aware of, in order to guarantee that the work motivation and the job satisfaction of software engineers will not be strongly affected. Besides, it is important to be concerned of the following situations.

First, the lack of feedback increases the dissatisfaction that might result from the temporary low performance after job rotations. Therefore, to apply this practice, the software organization needs to keep data about the work of each software engineer constantly updated and these data should be passed on to each project to which the individual is rotated.

Second, the level of commitment of software engineers with their tasks has to be considered when planning job rotations, because in some situations the individuals value well defined work more than work variety. In this case, the motivational effects of work variety tend to be reduced if the software engineer was rotated to another project before completely finishing his/her previous activities.

And finally, the level of cognitive overload can be even higher if the rotated software engineer has to transfer knowledge back to the previous team at the first days after the job rotation and it also impacts the indiviaudual performance.

Figure 7, summarizes the balance need in job rotations of software engineers in order to preserve their motivation and satisfaction.

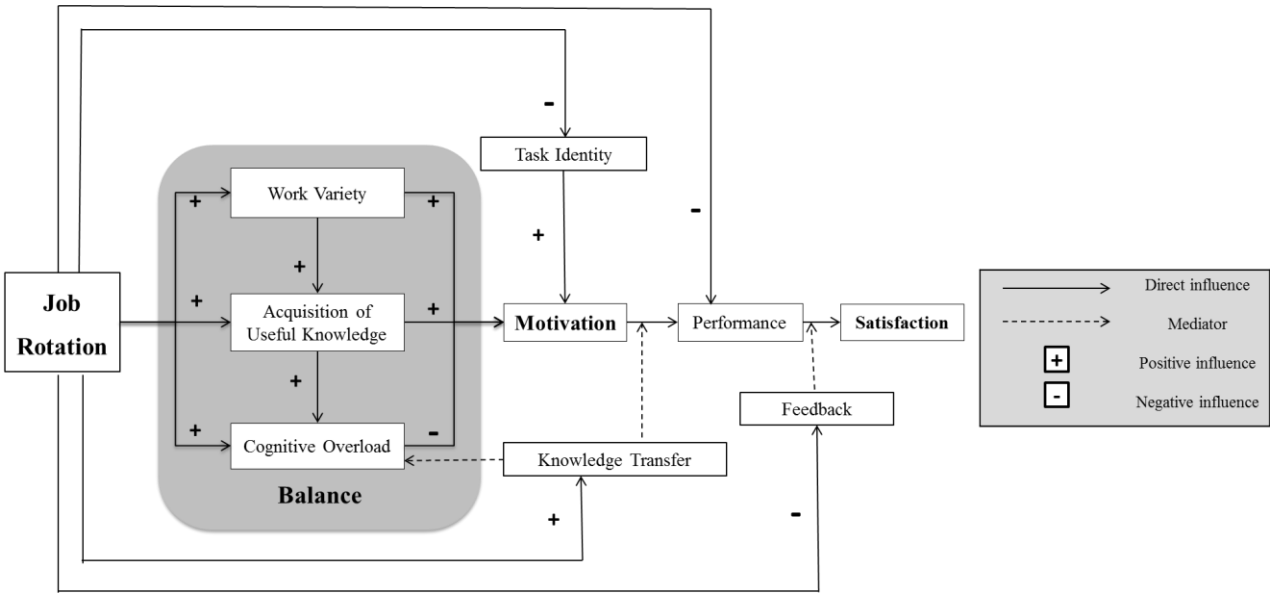

Figure 7. Balance required to reduce the negative impact of job rotations on motivation and satisfaction of software engineers

4.3 Member Checking

The member checking represented an important component to validate this research for two reasons. First, to assess the accuracy of the interpreted data. Second, to refine the level of representativity of each factor that is manifested through job rotation, in this context, and that can impact positively or negatively the motivation and satisfaction of software engineers.

Figure 8 represents the level of agreement of interviewees in relation to the interpretations and observations extracted from the data collected. The participants strongly agreed with perceptions about how job rotation is related to acquisition of useful knowledge (AUK), work variety (WV), and cognitive overload (CO).

There was a small divergence among those who completely agreed with the observation about performance and feedback, and those who partially agreed with this, declaring that these two factors depend on the type of project involved in the process of rotation. This divergence can be explained by the different point-of views of leaders and managers in relation to the opinion of the rest of the team, especially about feedback. Participants in leading position usually do not see lack of feedback as a major problem in this process, while software engineers usually see this as a problem.

Participants also made some considerations about the observations about knowledge transfer (KT) and task identity (TI), and partially agreed with what was interpreted around this two factors, based on the premise that both the type of project and the professional skills might increase or decrease the effects of these two factors during a rotation. Moreover, there is a link between these two elements, because the more a software engineer is rotated without finishing his/her work, the more knowledge transfer will be required to the new individual responsible for the tasks.

None of the participants of member checking completely desagreed with the interpretation of data collected and the explanations constructed. Nevertheless, there is a clear divergence on the opinion of participants who had participated in a rotation and those who were only indirectly involved with this practice.

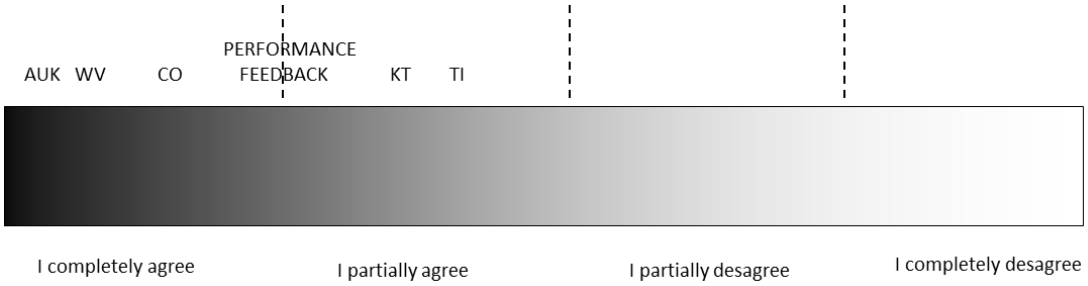

Figure 8. Level of agreement of participants to the interpretations of the study

Finally, the participants commented about the frequency in which each factor was manifested during the rotations (Figure 9). This is an important outcome to increase the representativeness of what was discussed in the interviews (Figure 4 and 5), considering that some factors were strongly mentioned and others hardly cited by the participants. For instance, feedback and task identity were mentioned by few participants but all participants of member checking agreed that this two factors are commonly mannifested during a rotation. Further, it clearly demonstrates that the practice of job rotation frequently influences one or more antecedents of motivation and satisfaction of software engineers.

Cognitive overload, acquisition of useful knowledge, and work variety are the most common factors affected by job rotations, followed by the problems with performance and feedback. However, the problems related to the lack of well defined work (task identity), exist, but don't happen as often as the others.

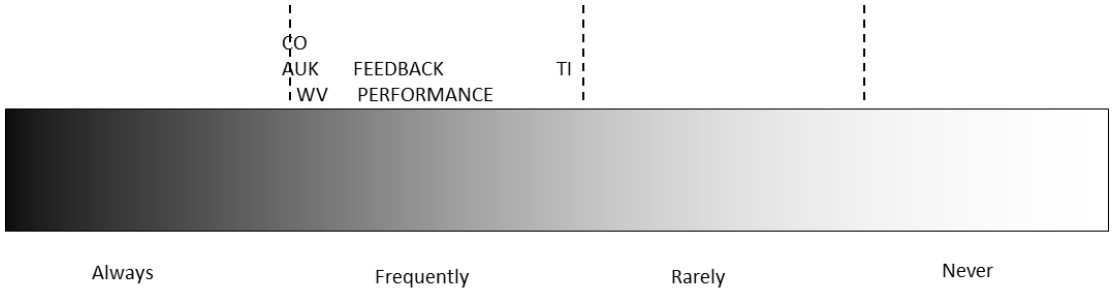

Figure 9. Frequency in which each factor is manifested through job rotations

4.4 Summary

The findings demonstrate that the practice of job rotation can influence motivation and job satisfaction of software engineers both positively and negatively.

Job rotation can positively influence the motivation of software engineers, once this practice enhances the variety of work and enables acquisition of useful knowledge. These rotations also influence the motivation in a negative way, because it can increase cognitive overload and impair task identity.

The rotations have negative impact on satisfaction not just because it can negativelly impact motivation, but also because it can decrease individual performance and cause loss of feedback.

Further, job rotations commonly requires transfer of knowledge increasing cognitive overload and decreasing performance, once individuals that are rotated to other software project usually have to keep in contact with their previous team to transfer information about their prior tasks.

Thus, the practice of job rotation in software organizations demands a balance among the main positive and negative factors that affect motivation and that are influenced by rotations, in order to avoid a decrease on the levels of performance, and consequently the satisfaction of software engineers. Further, it is important to be aware of the importance of feedback to software engineers work, because the lack of feedback increases the dissatisfaction resulting from rotations.

Chapter 5 Discussion

This Chapter, discusses additional remarks about the practice of job rotation in software projects, observations made in light of the TMS-SE and the theories constructed in previous studies, in order to enhance the findings and the theoretical framework.

Initially, this research demonstrated that the TMS-SE proposed by França (2014) can be apllied to different contexts in software engineering, once the practice of job rotation was not observed in the conception of this theory. However, considering especially job rotations, a new element was attached to the initial theory, the Knowledge Transfer, which can increase the Cognitive Overload of individuals recently rotated, due to its one-way characteristic, and posteriorly have a negative effect on motivation and performance.

Individual characteristics continues representing an important role in this scenario, once this element mediates the intensity of how each factor is perceived by software engineers after job rotations. Thereby, individual characteristics such as education, work experience, and technical role influenced the tendency to mention more positive or negative descriptons to the factors observed. However, we have not assessed participants' personalities, which may be influential in this aspect.

Figure 10 demonstrates the specific traits of the TMS-SE influenced by the practice of job rotation at software organizations, in which the boxes highlighted in green represent the positive influence of job rotation and those boxes in red represent the negative influence of this practice. Boxes in gray demonstrate the elements of TMS-SE with none identified influence of job rotations.

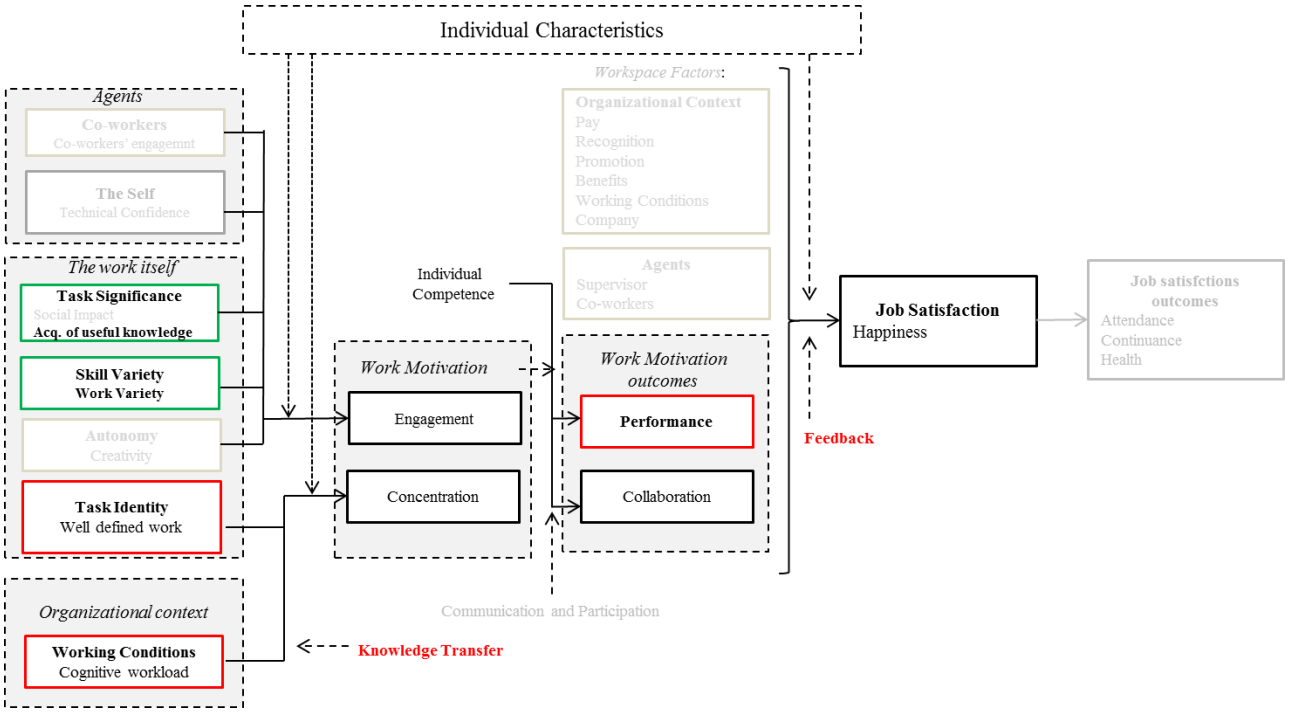

Figure 10. Elements of TMS-SE manifested in face to Job rotations

Moreover, other considerations can be inferred based on the findings of previous research about job rotation in software engineerig and in other areas. As discussed in the observations of Birk et al. (2003), software organizations can develop communication processes using job rotation, and an important issue around this process should be the communication between project managers to guarantee the manutenability of feedbacks to software engineers.

In addition to communication, job rotations can actually be applied to facilitate the knowledge sharing throughout software projects, as pointed out by Rus and Lindvall (2002). Because different professionals with different skills are constantly participating of different teams, the sharing process is inevitable. However, practitioners should be aware to the problems with unilateral transfer of

knowledge and how it can adversely affect the motivation of individuals by the increase of cognitive workload.

Similar to the observations of Ollo-Lopez, Bayo-Moriones, and Larraza-Kintana (2010) in the area of Business Administration, job rotation increases task variety in the context of software engineering. However, differently from their findings, job rotation can decrease and impair task identity of software engineers, which represents a challenge to software organizations willing to apply this practice. Besides, our findings do not support the conclusions of Cohen, Birkin, et al. (2004) that job rotation practice might not be appropriate among software development teams, because this practice can indeed produce satisfactory results with software engineers and the improvement of technical and practical skills, if the balance discussed before is taken into account.

Further investigations will be needed to construct a complete framework to guide practitioners in the adoption of job rotations, especially because this research regarded only data about the motivation and satisfaction of software engineers. Nevertheless, considering these two aspects, the following recommendations arise:

- Planning the balance among the positive and the negative effects of a rotation should be the first concern of project managers.
- Team members who are about to finish their tasks are the best candidates to a possible rotation, considering the fact that software engineers strongly value well defined work and task identity.
- Candidates to job rotations should be aware about how different is the new project that they are about to go and the possibility of a performance decrease at the beginning.
- The acquisition of the useful knowledge needed in the new project might be an intermediary process before the job rotation happens. Thus, the software

engineer can avoid the cognitive overload resulting from the great number of information to be assimilated.

- This intermediary phase might also guarantee that the process of knowledge transfer will not completely overload the software engineer, once he/she will not perform new tasks in this period, only learn about the new project and transfer his/her knowledge about the previous project to the new individual responsible for his/her previous tasks.
- Managers and leaders should be aware to the importance of feedbacks in face of job rotations and its importance to the satisfaction of software engineers, especially in this context.

Finally, job rotations are recommended to software organizations in order to stimulate the acquisition of knowledge and development of professionals abilities, enhance the perceptions about the business, expand networks and relationships, improve the process and the software projects, and, of course, enhance the motivation and the satisfaction of software engineers by providing a range of job options.

Chapter 6 Conclusions and Future Work

This research presented findings of a case study which aimed to investigate whether the practice of job rotation can influence the motivation and satisfaction of software engineers. The methodological framework of this research was elaborated with the objective to enable a deep understanding of the phenomenon in order to provide qualitative explanations.

The results pointed out seven factors that are commonly manifested during job rotations of software engineers amongst different projects within a organization, and that may affect their motivation and satisfaction: work variety, acquisition of useful knowledge, cognitive overload, well defined work, knowledge transfer, performance, and feedback. Further, the findings alert for a planning that concerns a balance between the positive and the negative effects of job rotations on motivation and satisfaction of these individuals in order to achieve satisfactory results when applying this practice in software organizations.

Thus, in summary, this research provided contributions both for academy and practitioners in software engineering field:

- The literature review about job rotation in software organizations pointed out the lack for theoretical and empirical studies regarding this topic.
- The findings achieved in this study demonstrated that the propositions of the TMS-SE developed by França (2014) are applicable in a context different from those observed to construct the theory.
- Through the case study, a systematic analysis on how the job rotations can impact the motivation and the satisfaction of software engineers was performed.

- From a methodological point of view, this research used a complete framework to develop qualitative research in the context of software engineering, including the process of member checking.
- The results revealed observable aspects of motivation and satisfaction of software engineers, which have not been addressed on TMS-SE.
- For the practitioners, initial recommendations were provided for those who apply or want to apply the practice of job rotation in software development teams, to maintain high levels of motivation and satisfaction.
- The discussions emphasized how job rotation can be useful to software organizations and to the professional growth of software engineers.

The main limitation of this research was the lack of theoretical and empirical studies addressing the practice of job rotation in the context of software engineering, especially in software organizations and software development teams. This fact turned more difficult the process of enfolding the literature, in order to confront these results with the point of views of other researchers. Furthermore, as a single case study, the findings of this research are related to the context of the participants and the projects analysed in this research. Nevertheless, analytical generalization of these findings can be done by applying and checking the findings in other settings.

Regarding the publication of this research, a paper comprising all the results and conclusions of this work will be submitted for publication in an international journal. Besides, in order to spread the knowledge constructed in this study and to interact and collect feedback from the software engineering community, the obtained results will be compiled in two more papers of conference: (i) the first paper will focus on the lack of research about job rotation in software teams and the results of evidence found through the literature review; (ii) the second paper will present the main observations on the effects of job rotation in

motivation and satisfaction of software engineers and the initial recommendations proposed to managers and leaders of software projects.

Our findings recommend the use of job rotation in organizations, once the efficient management of rotations can reduce the negative effects of this practice, enabling a better appreciation of the positive effects. Nevertheless, regarding the improvement of the current knowledge about this topic, future research may be implemented towards a better understanding of job rotations in software organizations and also to understand how this practice can affect software development teams.

First, the execution of a longitudinal study combining both quantitative and qualitative analysis may provide further insights about the effect of job rotations onn motivation and satisfaction of software engineers. Thus, it is expected that the results of such research can be generalized to a higher number of cases and differents types of software organizations.

Secondly, a systematic mapping study can be performed in order to understand the effects of job rotations in other sciences and industries. Thus, the findings about this topic can be used to improve the theoretical background and build a set of hypotheses in order to understand which other aspects of software development process and software teams are affected by job rotation.

And finally, by applying multiple case studies it might be possible to observe the real impact of job rotations in others aspects around software engineers and software teams, not exclusively motivation and satisfaction of individuals. Thereby, a more complete set of solutions, recommendations, and guidelines regarding the adoption of job rotation by software organizations can be proposed to assist project managers and leaders in this specific context.

References

BANDEIRA-DE-MELLO, R., CUNHA, C. Operacionalizando o método da Grounded Theory nas Pesquisas em Estratégia: técnicas e procedimentos de análise com apoio do software ATLAS/TI. In: Encontro de Estudos em Eestratégia da ANPAD, 1., 2003, Curitiba. Anais... Curitiba: Anpad, 2003.

BEECHAM, S.; BADDOO, N.; HALL, T.; ROBINSON, H.; SHARP, H. Motivation in Software Engineering: A systematic literature review. Information and Software Technology, v. 50, 2007.

BIRK, B.; HELLER, G.; JOHN, I.; SCHMID, K.; VON DER MAßEN, T.; MULLER, K. Product line engineering, the state of the practice. IEEE Software, v. 20, n. 6, p. 52-60. 2003.

CAMPION, M.; CHERASKIN, L.; STEVENS, M. Career-related antecedents and outcomes of job rotation. Academy of Management Journal, v. 37, n. 6, p. 1518-1542. 1994.

COHEN, C. F.; BIRKIN, S. J.; GARFIELD, M. J.; WEBB, H. W. Managing conflict in software testing. Communications of the ACM , v. 47, n. 1, p. 76-81. 2004.

COYNE, P. F. An evidence-based model of job rotation. PhD Dissertation. Middlesex University. London, UK, 2011.

DIEGO-MAS, J. A.; ASENSIO-CUESTA, S.; SANCHEZ-ROMERO, M. A.; ARTACHO-RAMIREZ, M. A. A multi-criteria genetic algorithm for the generation of job rotation schedules. International Journal of Industrial Ergonomics, v. 39, n. 1, p. 23-33. 2009.

EASTERBROOK S.; SINGER J.; STOREY M-A.; DAMIAN D. Selecting Empirical Methods for Software Engineering Research. In: Shull F. et al. (Edt). Guide to advanced empirical software engineering, Chapter 11. London: Springer-Verlag. 2008.

EISENHARDT, K. M. Building Theories From Case Study Research. The Academy of Management Review, v. 14, n. 4, (Oct., 1989), p. 532-550. 1989.

ERIKSSON, T.; ORTEGA J. The adoption of Job Rotation: testing the theories. Aarhus School of Business, Department of Economics, 2004.

FRANÇA, A. C. C. A Theory of Motivation and Satisfaction of Software Engineers PhD Dissertation. Center for Informatics, Federal University of Pernambuco, Recife, 2014.

FRANÇA, A. C. C.; SHARP, H.; DA SILVA, F. Q. B. Motivated software engineers are engaged and focused, while satisfied ones are happy. Proceedings… Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM). Torino, Italy, 2014.

FRANÇA, A. C. C., GOUVEIA, T. B., SANTOS, P. C. F., SANTANA, C.,; DA SILVA, F. Q. B. Motivation in software engineering: a systematic review update Proceedings…. Proceedings of the 15th International Conference on Evaluation & Assessment in Software Engineering (EASE). IET, 2011. doi:10.1049/ic.2011.0019.

FÆGRI, T. E.; DYBÅ, T.; DINGSØYR, T. Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. Information and Software Technology, v. 52, n. 10, p. 1118-1132. 2010.

HACKMAN, R. J.; OLDHAM, G.; JANSON, R..; PURDY, K. A New Strategy for Job Enrichment. California Management Review, v. 17, n. 4, p. 57-71, 1975.

HARPER, M.; COLE, P. Member Checking: Can Benefits Be Gained Similar to Group Therapy? The Qualitative Report, n. 17, n. 2, p. 510-517. 2012.

HO, W. H.; CHANG, C. S.; SHIH, Y. L.; LIANG, R. D. Effects of job rotation and role stress among nurses on job satisfaction and organizational commitment. BMC Health Services Research, v. 9, n. 8. 2009.

HSIEH, A.; CHAO, H. A Reassessment of the Relationship between Job Specialization, Job Rotation and Job Burnout: Example of Taiwan's High-technology Industry'. International Journal of Human Resource Management, v. 15, n. 6, p. 1108-1123. 2004.

KAYMAZA, K. The Effects of Job Rotation Practices on Motivation: A Research on Managers in the Automotive Organizations. Business and Economics Research Journal, v. 1, n. 3. 2010.

KREFTING, L. Rigor in qualitative research: The assessment of trustworthiness. The American Journal of Occupational Therapy, v. 45, n. 3, p. 214-222. 1991.

KUIJER, P. P.; VRIES, W.H.; VAN DER BEEK, A. J.; VAN DIEEN, J. H.; VISSER, B. FRINGSDRESEN, M. H. Effect of job rotation on work demands, workload, and recovery of refuse truck drivers and collectors. The Journal of Human Factors and Ergonomics Society, v. 46, n. 3, p. 437-48. 2004.

LARSSON, M.; WALL, A.; NORSTRÖM, C.; CRNKOVIC, I. Technology Transfer: Why some Succeed and some don't. Proceedings… Proceedings of the International Workshop on Software Technology Transfer in Software Engineering. New York, USA, 2006.

LOCKE, E. A. What is job satisfaction? Organizational Behavior and Human Performance, v. 4, n. 4, p. 309-414. 1969.

MERRIAM, S. B. Qualitative Research: a Guide to Design and Implementation. San Francisco, CA: Jossey-Bass, 2009.

MCGILLICUDDY, S. IT job rotation rare, but critical for business alignment. Disponível em: http://searchcio.techtarget.com/news/1244573/IT-job-rotation-rare-but-critical-for-business-alignment. Acesso em: 30 out. 2014.

MONTEIRO, C. V. F. Innovative Behavior Model of Sofware Professionals. PhD Dissertation. Center for Informatics, Federal University of Pernambuco, Recife, 2014.

OLLO-LOPEZ, A.; BAYO-MORIONES, A.; LARRAZA-KINTANA, M. The relationship between new work practices and employee effort. Journal of industrial relations. v. 52, n. 2, p. 219-235. 2010.

PASSOS, C.; CRUZES, D. S.; MENDONÇA, M. Applying Theory of Reasoned Action in the Context of Software Development Practices. Proceedings… Proceedings of International Conference on Evaluation and Assessment in Software Engineering. Porto de Galinhas, Brasil, 2013.

PIRZADEH, L. Human Factors in Software Development: A Systematic Literature Review. Master Thesis. Chalmers University of Technology. Göteborg, Sweden, 2010.

RICHARDSON, A.; DOUGLAS, M.; SHUTTLER, R.; HAGLAND, M. R. Critical care staff rotation: outcomes of a survey and pilot study. Nursing in Critical Care, v. 8, n. 2. 2003.

ROBSON, C. Real World Research. Oxford: Blackwell. 2. ed., 2002.

RUNESON P.; HÖST, M. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, v. 14, n. 2, p. 131-164. 2009.

RUS, I.; LINDIVALL M. Guest Editors' Introduction: Knowledge Management in Software Engineering. IEEE Software, v. 19, n. 3, p. 26-38. 2002.

SEAMAN, C. Using Qualitative Methods in Empirical Studies of Software Engineering. Disponível em: http://ccsl.ime.usp.br/files/QualMethods%20minicourse%20USP-1.pdf. Acesso em: 05 dez. 2014.

STRAUSS, A.; CORBIN, J. Pesquisa qualitativa: técnicas e procedimentos para o desenvolvimento de teoria fundamentada. 2. ed. Porto Alegre: Artmed, 2008.

WEERD-NEDERHOF, P. C.; PACITTI, P. J.; GOMES, J. F. S.; PEARSON, A. W. Tools for the improvement of organizational learning processes in innovation. Journal of Workplace Learning, v. 14, n. 8, p. 320-331. 2002.

WOODS, R. Human Resources Management. Michigan: AHMA, 1995.

YIN, R. K. Case study research: design and methods, 3. ed. Thousand Oaks, CA: Sage Publications, 2003.

Appendices

APPENDIX A: INTERVIEW SCRIPT (PT_BR)

GUIA DE ENTREVISTA COM ENGENHEIROS DE SOFTWARE SOBRE JOB ROTATION

APRESENTAÇÃO

- Apresentação do pesquisador e cumprimentos.
- Agredecimento ao participante
- Solicitação de permissão para gravar (caso o participante não autorize a gravação, todos os passos seguintes devem ser registrados por escrito.)

INTRODUÇÃO

O objetivo desta pesquisa é entender aspectos relacionados a movimentaçcao de pessoas dentro de equipes de desenvolvimento de software. Todas as informações fornecidas nesta entrevista serão tratadas como confidencial. Apenas a equipe de pesquisa, relacionada ao assunto, terá acesso às informações fornecidas. Em particular, nenhuma pessoa direta ou indiretamente ligada a empresa terá acesso às informações fornecidas nesta entrevista e em nenhuma outra fase da pesquisa. A equipe de pesquisa empregará todos os meios possíveis para evitar que informações individuais possam ser associadas diretamente aos participantes.

Sua participação nesta pesquisa é voluntária e você pode decidir não participar ou se retirar da pesquisa a qualquer momento. Caso você decida não participar, não receberá nenhuma sanção ou penalidade. Você concorda em participar desta pesquisa?

IDENTIFICAÇÃO DO ENTREVISTADO

- As informações a seguir serão utilizadas caso a equipe de pesquisa precise entrar em contato com você no futuro para esclarecimentos sobre a entrevista.
- Por favor, diga seu nome.
- Por favor, diga seu endereço de e-mail.

SOBRE AS RESPOSTAS

Não existem respostas certas ou erradas nesta entrevista. Nosso objetivo é coletar suas impressões, opiniões e sentimentos sobre os vários assuntos abordados. Leve o tempo que for necessário, tudo o que for importante para você me interessa. Reforçando que suas respostas não serão disponibilizadas para a empresa e, portanto, não terão nenhuma influência em avaliações realizadas pela empresa. Por favor, responda da forma mais sincera possível.

PERGUNTAS

AQUECIMENTO

Caracterização do entrevistado

- Idade
- Estado civil
- Qual e quando foi a última Titulação
- Tempo de atuação profissional

1. *Vamos iniciar com algumas informações gerais sobre sua história dentro desta organização. Por favor, conte-me em detalhes sua história, da seleção até o dia de hoje.*

2. *De que forma as suas opiniões e sentimentos a respeito da empresa modificaram-se desde a sua entrada? Você atribui essa mudança a que?*

3. *Que mudanças importantes você identifica na história da empresar? Como isso afetou suas equipes? Como isso lhe afetou?*

4. *Hoje em dia, o que é determinante para você continuar trabalhando nesta empresa?*

JOB ROTATION

Nesta empresa existe a cultura de movimentação de colaboradores entre equipes durante e ao final dos projetos (Job Rotation). As próximas perguntas estão relacionadas as movimentações (entradas e saídas de pessoas) durante o desenvolvimento de um projeto.

Para todos os entrevistados

5. *O que você acha sobre essas movimentações internas?*
   - ✓ *Como você acha que os funcionários se sentem?*
6. *Como você se sente com a possibilidade de ser "movimentado" para outra equipe no meio do desenvolvimento de um projeto? Por quê?*
7. *Como você se sente com a possibilidade de ocorrer uma movimentação (saída ou entrada de novos membros) dentro da sua equipe atual? Por quê??*

Apenas para os entrevistados que já foram movimentados

8. *Você já foi mudado de equipe no meio de um projeto? Quantas vezes, aproximadamente? Fale me sobre essa experiência? Como você avalia a experiência?*
   - ✓ *Em que fase do projeto ocorreu essa movimentação?*
   - ✓ *Como e porque ocorreu essa movimentação?*
   - ✓ *Como você se sentiu? Teve problemas de adaptação?*

- ✓ *Você acha que essa movimentação afetou o seu desempenho?*

Apenas para os entrevistados que já tiveram pessoas movimentadas na equipe

9. *Nas equipes que você já participou, alguma vez ocorreu alguma movimentação de outras pessoas da equipe? Fale me sobre essa experiência.*
   - ✓ *Como você avalia a experiência?*
   - ✓ *Como a equipe se sentiu? Como você se sentiu?*
   - ✓ *Quais foram os impactos positivos e negativos causados por essa movimentação?*
   - ✓ *Afetou estimativas, tempo, custo, escopo, etc.? Como?*
   - ✓ *Afetou a cooperação entre os membros?*

Para todos os entrevistados

10. *O que você diria se te oferecessem a oportunidade agora de mudar para outro projeto? O que você pesaria?*
    - ✓ *Se você tivesse a chance de fazer o mesmo tipo de trabalho em outra equipe como você se sentiria sobre a mudança?*
11. *Se você fosse movimentado agora para outra equipe, como isso impactaria o andamento do seu projeto atual?*
    - ✓ *Como você avaliaria a importância das suas atividades para o projeto.*
12. *E qual seria o impacto da sua saída em outros aspectos da sua equipe atual não diretamente associado ao desenvolvimento do projeto?*
    - ✓ *Como você avaliaria a sua influência dentro da sua equipe?*

APPENDIX B: MEMBER CHECKING INTERVIEW SCRIPT (PT_BR)

GUIA DE ENTREVISTAS CONFIRMATÓRIA SOBRE CONTEÚDO

N	*Afirmação*	Concordância
1	**Work variety (Variedade do Trabalho)** Movimentações internas permitem que os engenheiros de software possam trabalhar com diferentes projetos e tecnologias, trazendo variedade de trabalho.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		
2	**Acquisition of useful knowledge (Aquisição de Conhecimento)** Movimentações internas possibilitam que os engenheiros de software obtenham conhecimento técnico e prático sobre diversos aspectos profissionais.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		
3	**Cognitive workload (Sobrecarga Cognitiva)** Movimentações internas afetam o indivíduo quando as suas novas responsabilidades em outro projeto requerem um alto grau esforço mental e de concentração.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		
4	**Feedback (Feedback)** Movimentações internas afetam o feedback do trabalho, uma vez que uma pessoa pode não passar tempo suficiente em uma equipe para serem avaliado.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		
5	**Well Defined Work (Compromisso com a Tarefa)** A movimentação interna faz com que os indivíduos tenham de sair do projeto sem finalizar completamente o trabalho que estava desenvolvendo anteriormente.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		
6	**Knowledge Transfer (Troca de conhecimento)** Geralmente, após uma movimentação, a pessoa movimentada precisa ficar em constante contato com a equipe anterior para repassar informações sobre o trabalho que estava desenvolvendo anteriormente.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		

7	**Performance (Performance)** A movimentação interna dos membros pode afetar a performance individual. Ao ser movimentado, o indivíduo terá baixo desempenho até que consiga assimilar corretamente as suas atividades na nova equipe.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
	Performance (Performance) A movimentação interna dos membros pode afetar a performance individual, pois requer tempo para treinar um novo membro recém chegado na equipe.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
	Performance (Performance) A movimentação interna dos membros pode afetar a performance individual, pois essa movimentação pode ocasionar aumento na carga de trabalho de alguns membros da equipe anterior que terão de suprir a falta deixada pelo membro que foi movimentado para outro projeto.	( ) Concordo Completamente ( ) Concordo Parcialmente ( ) Discordo Parcialmente ( ) Discordo Totalmente
Comentários		

GUIA DE ENTREVISTAS CONFIRMATÓRIAS SOBRE FREQUÊNCIA

	Afirmação	Concordância
	Work variety (Variedade do Trabalho) As movimentações internas trazem oportunidade de trabalhar com um novo projeto ou tecnologia.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		
	Acquisition of useful knowledge (Aquisição de Conhecimento) As movimentações internas possibilitam a obtenção de novos conhecimentos.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		
	Cognitive workload (Sobrecarga Cognitiva) As movimentações internas requerem um alto grau de concentração para desenvolver as novas atividades no novo projeto.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		
	Feedback (Feedback) As movimentações internas prejudicam o recebimento de feedback sobre o trabalho.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		
	Well Defined Work (Compromisso com a Tarefa) As movimentações internas fazem com que não seja possível desenvolver um trabalho ou tarefa em um projeto do início ao fim.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		
	Knowledge Transfer (Troca de conhecimento) Após uma movimentação, a pessoa movimentada precisa manter contato com a equipe anterior para repassar informações sobre o trabalho que estava desenvolvendo anteriormente.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		

	Performance (Performance) A movimentação causa queda temporária da performance individual.	( ) Sempre ( ) Frequentemente ( ) Raramente ( ) Nunca
Comentários		