Information and Software Technology

Benefits and limitations of project-to-project job rotation in software organizations: A synthesis of evidence

Ronnie E.S. Santos[a], Fabio Q.B. da Silva[a,*], Maria Teresa Baldassarre[b], Cleyton V.C. de Magalhães[a]

[a] Centro de Informática – Universidade Federal de Pernambuco, Cidade Universitária – 50.540-740 Recife – PE – Brasil
[b] Dipartimento di Informatica – Università degli Studi di Bari, Via E.Orabona 4, Bari, Italy

ARTICLE INFO

Article history:
Received 27 September 2016
Revised 14 April 2017
Accepted 19 April 2017
Available online 20 April 2017

Keywords:
Job rotation
Work design
Software engineering
Systematic literature review
Case study
Replication

ABSTRACT

Context: Job rotation has been proposed as a managerial practice to be applied in the organizational environment to reduce job monotony, boredom, and exhaustion resulting from job simplification, specialization, and repetition. The scientific literature distinguishes between *job-to-job* and *project-to-project* rotations. Despite the potential benefits and its actual use on behalf of software companies, we do not have an accumulated body of scientific knowledge about benefits and limitations of job rotation in software engineering practice. In particular, we have no concrete empirical evidence about the use of project-to-project rotations in practice.
Goal: We aim to identify and discuss evidence about project-to-project (P2P) job rotation, in order to understand the potential benefits and limitations of this practice in software organizations.
Method: We deployed a mix-method research strategy to collect and analyze empirical evidence from the scientific literature, performing a systematic literature review, on one hand and from industrial practice, performing qualitative case studies on the other. We synthesized the evidence using techniques from meta-ethnography.
Results: We found eight benefits, nine limitations, and two factors classified as both benefits and limitations of P2P rotations in software engineering. Different research methods yielded confirmatory and complementary evidence, emphasizing the importance of conducting mix-method research. We found no contradictory evidence and five factors were identified in more than one study using different research methods, contributing to the strength of the evidence.
Conclusion: We synthesized evidence from multiple sources and used different research methods concerning the benefits and limitations of P2P rotation in software engineering practice. Our findings show that rotation tends to benefit important job outcomes, such as motivation, and to decrease job monotony. The main limitations were associated with the potential increase in intra-group social conflicts, individual cognitive effort, and workload, and a temporary decrease in productivity.

1. Introduction

Job rotation has been proposed as a managerial practice to be applied in the organizational environment to reduce job monotony, boredom, and exhaustion resulting from job simplification, specialization, and repetition [41]. Researchers have studied job rotation in diverse types of organizations and different jobs, such as nursing, business, and manufacturing industries, finding negative and positive effects on factors such as knowledge exchange, job satisfaction, motivation, and job burnout [4,9,29,32]. In an attempt to tap the above-mentioned benefits, software companies have also been using job rotation in practice. However, so far there has been no consistent and comprehensive body of evidence about its benefits and limitations in software engineering practice. Our goal is to contribute to reducing this knowledge gap.

Woods defines job rotation as "the systematic movement of employees from job to job, or project to project, within an organization, as a way to achieve various human resources objectives" [42]. In *job-to-job* (J2J) rotation, individuals are rotated between different jobs in the same organization, to perform activities with distinct natures. In *project-to-project* (P2P) rotation, individuals are moved between projects of similar nature (e.g. two software development projects), often keeping the same technical role. In a recent systematic review [35], we identified that both types of ro-

[*] Corresponding author.
E-mail addresses: fabio@cin.ufpe.br (F.Q.B. da Silva), mariateresa.baldassarre@uniba.it (M.T. Baldassarre).

http://dx.doi.org/10.1016/j.infsof.2017.04.006
0950-5849/© 2017 Elsevier B.V. All rights reserved.

tation had been used by software organizations. For instance, Fægri et al. [11] investigated job-to-job rotation in which software developers were rotated to customer support to increase knowledge redundancy at the organizational level. On the other hand, Santos et al. [34] investigated an organization in which project-to-project rotations were used to supply the need for specialized skills in a given project and to increase the variety of the tasks performed by the employees.

In our systematic review [35], we also identified that these two types of rotation serve different organizational and individual purposes, thus having distinct benefits and limitations. According to our findings, P2P rotations seem to enhance team flexibility and reduce job monotony through its increase in task variety. On the other hand, J2J rotations focus on organizational goals, in particular on organizational understanding and its effects on innovation and the establishment of multicultural teams. Further, we also found that J2J and P2P rotations seem to differ in how they address organizational, team, and individual needs, with J2J rotations focusing on organization and managerial needs, and P2P affecting workgroup and individual ones.

However, among the 17 primary studies analyzed in the abovecited review, only one investigated job rotation in the context of software engineering as its main goal. The remaining studies gathered evidence about this practice while researching other topics, in a non-intentional way, resulting in a low strength of evidence. Furthermore, the only study that focused on job rotation investigated J2J rotation in a context where software engineers were rotated to other areas in the company. No primary study, therefore, explicitly investigated the benefits and limitations of P2P rotations in software engineering practice. This is an important knowledge gap for the proper application of this managerial practice.

In this article, our goal is to contribute to reduce this knowledge gap by answering the following research question:

> *RQ: What are the benefits and limitations of the application of project-to-project (P2P) job rotation in software development industrial practice?*

To achieve this goal, we synthesized evidence from multiple primary and secondary studies using techniques from meta-ethnography [28]. We used four sources of evidence in this synthesis: (i) a systematic literature review, published by Santos et al. [35] (hereafter called SLR) covering studies published between 1997 and July 2015; (ii) an industrial case study, published by Santos et al. [34], which focused on the motivational aspects of P2P rotations (hereafter called Case I), investigating the potential effects of P2P rotations on the motivation and satisfaction of software engineers in industry. These two studies created a preliminary understanding about potential benefits and limitations of job rotation in software engineering practice. They found five distinct benefits (two in common) and six distinct limitations (no intersection) of the use of P2P in software engineering.

We added two further sets of evidence to this initial body of knowledge: (iii) the extension of Case I (unpublished) covering a broader set of factors besides the motivational ones (hereafter called Case I – Extension); (iv) a second industrial case study (unpublished), conducted using the same protocol of Case I and a multiple case replication logic [45] (hereafter called Case II – Replication). We then compared the findings from the software engineering contexts with results from other areas.

Our meta-ethnographic synthesis resulted in 19 distinct factors that are potentially affected by the practice of P2P job rotations in software development organizations. Among them, eight were considered benefits or positive outcomes of the application of P2P rotations, nine were identified with potentially negative outcomes and, thus, considered as limitations of this practice, and two were considered as both benefits and limitations, depending on contextual factors. This synthesis produced the most comprehensive set of potential benefits and limitations of the use of P2P rotations in software engineering so far. Therefore, it constitutes a solid contribution to research and can also be used to inform practice, as discussed below.

Summarizing, this article synthesizes and extends the findings from previous publications. Firstly, we extended our first case study and then replicated it. Secondly, we used meta-ethnographic techniques to synthesize and consolidate the evidence from the different data sources.

The rest of this article is organized as follows. In Section 2, we present the conceptual background that characterizes job rotation in general and in software engineering projects. In Section 3, we describe the research methods used in each individual study and also the steps used in the meta-ethnographic synthesis. In Section 4, we present the results of our study. In Section 5, we discuss the implications of our findings for research and practice, together with the limitations of this study. Finally, in Section 6, we present our conclusions and directions for future research.

2. Background and related work

We start by presenting definitions and types of job rotation discussed in the literature. Then, we present a review of studies about job rotation performed in other areas. We also characterize P2P job rotation in software projects.

2.1. Definitions and types of job rotation

Since the 1950′s, job rotation has been proposed as a practice to be applied in the organizational environment to reduce job monotony, boredom, and exhaustion, resulting from job simplification, specialization, and repetition [41]. The literature presents many definitions to describe this practice that focuses on distinct approaches to achieve the desired organizational goals.

A group of authors focuses their definition on job-to-job (J2J) rotations. Coyne [5] described job rotation as the purposeful and organized movement of staff within and across organizational areas to enhance both the success of the company and the employability of staff. Kuijer et al. [24] stated that job rotation is a regular alternation between different jobs within an organization, based on a scheme or spontaneously based on the workers' personal needs. Richardson et al. [32] defined job rotation as a reciprocal exchange of staff between two or more areas for a predetermined period.

Other authors made explicit reference to project-to-project (P2P) rotations, in which individuals move among projects or places but keep the type of job or role they were performing before the rotation. In this group, Soderquist and Prastacos [39], Alei et al. [1] and Brady et al. [2] presented job rotation as a practice that allows individuals or group of individuals to be moved from team to team and from project to project within the same organizational area. In software engineering, this would be equivalent to moving engineers from one software development team to another team in the same organization.

Encompassing both types of rotations, Woods [42] 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, such as staffing jobs, orienting new employees, preventing job boredom or burnout, rewarding employees, enhancing career development, and exposing employees to diverse environments". In this study, we use Wood's characterization as the conceptual definition to guide our research.

Table 1
Job rotations in other areas.

Factors	Correlation with factor		Impact of factor on work	Benefit/Limitation
	J2J	P2P		
Organizational factors				
Organizational commitment	+[4] +[29]	+[18]	+	Benefit
Organizational understanding	+[32] +[44]		+	Benefit
Innovation	+[44]		+	Benefit
Learning costs	+[4]		–	Limitation
Communication	+[4]+ [32]	+[39]	+	Benefit
Time consuming	+[32]	+[39]	–	Limitation
Team factors				
Knowledge exchange	+[44]	+[39]	+	Benefit
Knowledge transfer		+[39]	+	Benefit
Work process and workflow	–[4]		+	Limitation
Work characteristics				
Task characteristics				
Task variety	+[4] +[32]– [19]	+[21]	+	Benefit/Limitation (J2J) | Benefit (P2P)
Task autonomy	–[19]		+	Limitation
Knowledge characteristics				
Acquisition of knowledge	+[32]	+[21] +[2]	+	Benefit
Specialization	–[19]		+/–	Benefit/Limitation
Social characteristics				
Social interaction	+[24]	+[21]	+	Benefit
Outcomes				
Individual outcomes				
Motivation	–[4]+ [32]	+[21]	+	Benefit/Limitation (J2J) | Benefit (P2P)
Job satisfaction	–[4]	+[18]	+	Limitation (J2J) | Benefit (P2P)
Career development	+[32]	+[1]	+	Benefit
Job outcomes and correlates				
Exhaustion	–[19]		–	Benefit
Professional efficacy	–[19]		+	Limitation
Productivity	–[4]		+	Limitation
Cognitive effort	+[29]+[9]		–	Limitation
Workload	+[4] +[9]		–	Limitation

2.2. Related work from other fields

We performed a review looking for articles addressing the impact or effect of job rotation on several work-related factors. We found 12 studies in distinct fields such as automotive industry and nursing. None of the studies addressed software engineering or software development organizations. Five studies addressed P2P rotations and seven studied J2J rotations. Table 1 summarizes our findings, which we used to build the initial conceptual framework of our research and guide our empirical studies. We also compared our results with the literature from other fields to raise the theoretical level and sharpen construct definitions, as recommended by Eisenhardt [8].

The studies identified work-related factors that were correlated with the use of job rotation (first column of Table 1). The studies identified direct and inverse correlations. Direct correlations (shown as a + sign in the second column of Table 1) indicate that the use of job rotation was related to the increase of the factor. For instance, study [21] found that the use of P2P rotations was directly related to an increase in task variety and individual motivation. Inverse correlations (shown as a – sign in the second column of Table 1) indicate that the use of rotations was related to a decrease on the factor. For instance, study [19] showed that the use of J2J rotation was correlated with a decrease in job specialization.

The studies also analyzed the impact of the factor on the work of individuals or organizational effectiveness. For instance, study [39] found that P2P rotations were time-consuming, which was considered as negatively affecting the work of individuals and, ultimately, organizational effectiveness. Other factors, such as motivation, job satisfaction, and innovation were considered as positively affecting the work. The third column in Table 1 indicates when the factor positively (+) or negatively (–) impacted the work.

In our research, we defined benefits and limitations of P2P rotations (fourth column of Table 1) combining the correlation of job rotation with the factor (second column of Table 1) and the impact of the factor on the work (third column of Table 1), as follows:

- + correlation and + impact is a Benefit: job rotation potentially increases a factor that has a positive impact on the work;
- – correlation and – impact is a Benefit: job rotation potentially decreases a factor that has a negative impact on the work;
- + correlation and – impact is a Limitation: job rotation potentially increases a factor that has a negative impact on the work;
- – correlation and + impact is a Limitation: job rotation potentially decreases a factor that has a positive impact on the work;

In two situations, the identification of benefits or limitations needed some care. First, studies did not always agree on the potential influence of job rotations on certain factors. For instance, job satisfaction was negatively correlated in one study of J2J rotations [4] and positively correlated in one study of P2P rotations [18]. In such cases, job rotation was found to be a benefit in some studies and limitation in others. We acknowledge these contradictory findings by indicating this in the fourth column of Table 1 as Benefit/Limitation.

A second situation resulted from the study of Hsieh and Chao [19], where the researchers found that the impact of the factor job specialization on individuals was dependent on the individual's attitude toward specialization, i.e., certain individuals prefer to become specialists in a narrow set of tasks or skills (high job specialization) whereas other individuals prefer to work on a broad range of tasks and use broad set of skills (low job specialization). Therefore, the impact of this factor on the work of individuals can be both positive and negative depending on individual characteristics. Therefore, this factor has a +/- impact sign in the third column of Table 1 and a Benefit/Limitation value in the fourth column.

We used the above rationale throughout the rest of this study, in particular in the construction of the summary tables throughout Section 4.

Table 1, adapted from Santos et al. [35], summarizes the findings of our literature review of other fields. To make the presentation of these findings consistent with the results of our synthesis, we grouped the work related factors into four categories:

- ***Organizational Factors***: those related to organizational wide and managerial aspects not directly related to a project team or an individual.
- ***Team Factors***: factors that are related to team characteristics and team level processes.
- ***Work Characteristics***: these characteristics broadly refer to the different ways in which a given work can be structured, assigned to individuals and/or teams, and performed. To organize work characteristics, we used the factor structure presented in the WDQ model [27]. It defines three categories of characteristics: (1) *Task Characteristics,* "concerned with how the work itself is accomplished and the range and nature of tasks associated with a particular job" [27]; (2) *Knowledge Characteristics:* "reflect the kinds of knowledge, skill, and ability demands that are placed on an individual as a function of what is done on the job" [27]; (3) *Social Characteristics*: group together the social and interactional aspects of the work reflecting "the fact that work is performed within a broader social environment" [27].
- ***Outcomes***: as it is common in work characteristics models [17], the term (work) ***Outcomes*** refers to factors related to what turns out from performing some work either as tangible factors, such as productivity or subjective, or less tangible ones such as motivation and burnout.

Although the grouping of the factors in Table 1 differs from the grouping used by Santos et al. [35], the factors and the corresponding findings are the same.

Regarding factors at the organizational and team level, it seems that job rotation was capable of achieving several organizational and team goals, in particular in the context of J2J rotations, although some limitations were found concerning learning costs, being time consuming, and disrupting workflow. Further, no contradictory evidence was found between P2P and J2J rotations.

Concerning work characteristics and outcomes, we have a less uniform scenario. In the J2J context, studies found contradictory evidence about the correlation of job rotation and task variety and motivation. For these two factors, there is a divergence among studies regarding whether J2J rotation offers benefits or limitations in practice. We did not find similar contradictions among the studies of P2P rotations. However, J2J and P2P studies do not agree with respect to the correlations of rotations with job satisfaction and individual motivation. P2P rotations were positively correlated with these factors whereas J2J rotations correlated negatively. These discrepancies could be explained by the potential effects of J2J on other factors, such as increase of cognitive effort and workload, and a decrease of task variety and task autonomy, which are likely to affect motivation and satisfaction.

We briefly discussed J2J and P2P rotations in this section to provide a broader view of the theme and to present the rationale we used to characterize benefits and limitations of job rotation. We have compared our findings for the software engineering context with the evidence from the literature in other fields, related to P2P rotation, in Section 5.1.

2.3. Characterizing project-to-project job rotation in software projects

Although both types of rotation can be used in software organizations, as identified by Santos et al. [35], the focus of this research is on P2P rotations in the software development context. In the early stages of our study, we built a characterization of P2P rotations in software projects using two sources of information. We interviewed project and human resources managers that have deployed P2P rotations in software companies to understand how rotations worked in practice in software development projects. We then compared and contrasted our findings with the definitions from the literature (Sections 2.1 and 2.2) to create a characterization of P2P rotations to guide our research. This characterization was built considering five elements, described below and illustrated in Fig. 1.

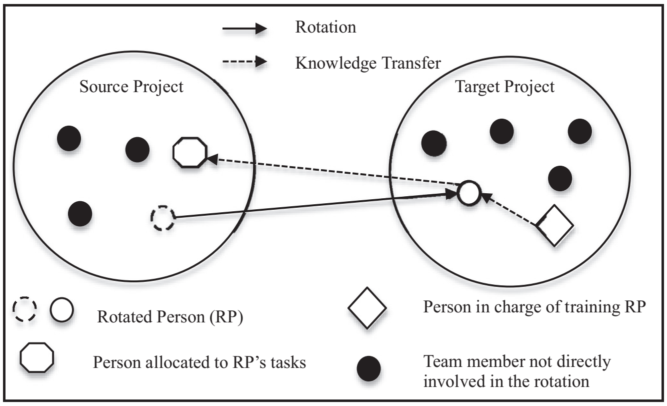

Rotation
Knowledge Transfer
Source Project
Target Project
Rotated Person (RP)
Person in charge of training RP
Person allocated to RP's tasks
Team member not directly involved in the rotation

Fig. 1. Illustration of a P2P rotation.

Definition – P2P job rotation is defined as the practice of moving one professional from one software project (the "source project") to another project (the "target project) within the organization. In most situations, the role (software engineer, test engineer, team leader, software architect, etc.) performed by the rotated person remains the same. However, this position could change under various circumstances, e.g., when a test engineer changes to the role of a software engineer to fulfill a resource need in the target project.

Agents – individuals in the organization who participate directly in one rotation: (MCR) manager in charge of the rotation; refers to the senior or project manager that implements the rotation; (RP) the rotated person that moves from source to target project in a rotation; (RPH) the rotated person host is the individual (or group of individuals) in charge of training RP in target project; (RPR) the rotated person replacement is the individual (or group of individuals) assuming RP's tasks in source project.

Triggers – a P2P rotation is triggered by three reasons: (1) target project needs: when the target project requires more manpower (quantity) or different set of skills (diversity); (2) source project needs: when the performance or skills of the rotated person is not compatible with the requirements of the source project and the person is rotated to a target project with more compatible performance or skill requirements (in such cases, the source project would potentially become a target project due to trigger 1); (3) individual request: an individual manifests the desire to change projects. Triggers 1 and 2 are related to organizational needs whereas trigger 3 addresses individual motivation and satisfaction needs.

Tasks – several tasks are directly related to the rotation. They are temporary and not directly related to the end tasks of the project. In fact, they support the rotation and create new workload on project managers and some team members at target and source project. These tasks are mostly planned and supervised by MCR: identifying the trigger, agents, source, and target project involved in the rotation; communicating the rotation to agents, source, and target project, potentially including projects' clients; actually moving the RP to target project; assigning tasks to RPH to train RP in

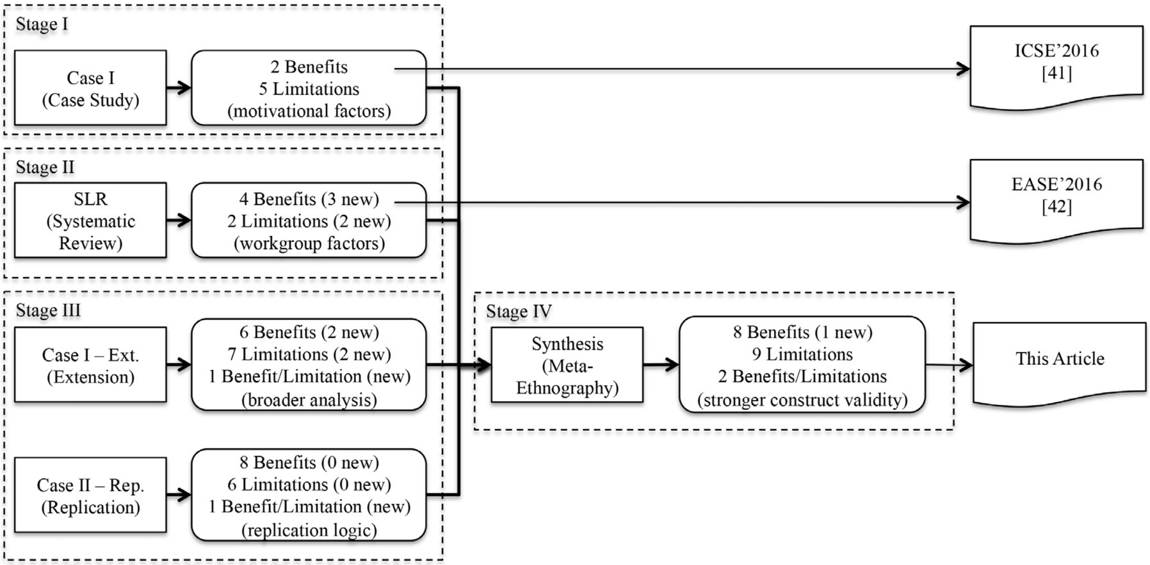


Fig. 2. Research stages.

the new project, if needed; assigning tasks to RPR to assume RP's tasks in source project, if needed.

Moment of rotation – a rotation can occur at any moment while the project is under execution.

Using this characterization helps to distinguish P2P rotations from other types of resource allocation. For instance, when a project finishes it is natural that the members of the project are allocated to new projects. We do not consider this re-allocation as a P2P rotation because several potential impacts (negative and positive) of rotation do not occur in this type of re-allocation. Further, this characterization also identifies agents that might have different perceptions of benefits and limitations of rotations, which becomes important when analyzing research findings.

3. Methods

We used a mix-method research strategy combining evidence from two types of studies: systematic literature review and qualitative industrial case studies. In this section, we start describing the stages of our research and then summarize the research method for each type of study.

3.1. Research stages

The results presented in this article were produced in several studies initiated in 2014. The four stages of this research are illustrated in Fig. 2, and described below.

Stage I – we started our investigation performing an industrial case study in 2014 (Case I). Our goal was to investigate the potential effects of P2P rotations on motivation and satisfaction of software engineers that worked in an organization that used rotations systematically. We, therefore, focused our analysis using pre-formed codes from a theory of motivation and job satisfaction of software engineers [12,13], even though we collected data that potentially covered other factors. This analysis identified two benefits and five limitations.

Stage II – before publishing the results from Stage I, we decided to understand how job rotation was being investigated in empirical software engineering research. In 2015, we conducted a systematic literature review (SLR) that covered studies published until July 2015. We analyzed 17 unique studies published in 18 articles, addressing both J2J and P2P rotations. The set of 11 studies that addressed P2P rotations presented four benefits (3 new ones with respect to Stage I) and two limitations (new).

We then consolidated the results of Stages I and II, and published them in separate venues. Santos et al. [34] published the full analysis of Case I, presenting a preliminary theory about the interacting effects of benefits and limitations of P2P rotations. Santos et al. [35] published the results of the SLR. The intersection of the results from Case I and the SLR were very small, with only one benefit in common and no common limitations. We attributed this small intersection to the focus of the analysis of Case I on motivational factors and also to the fact that no study in the SLR investigated P2P as its primary goal. This prompted us to perform an extended analysis of Case I data, looking for a broader set of factors, and to extend the findings of the SLR by synthesizing evidence from other studies.

Stage III – in the first half of 2016, we performed two studies that produced complementary results. We extended the analysis of the data collected in Case I (Case I – Extension), looking for benefits and limitations not directly related to motivation or satisfaction. This extended analysis found six benefits (2 new ones concerning stages I and II) and seven limitations (2 new ones). In parallel, we performed a replication of Case I (Case II – Replication) where we selected two other projects in the same software company and interviewed 14 participants that were not involved in Case I. We used the same data analysis technique looking for a broader set of benefits and limitations. Case II – Replication identified eight benefits (no new ones), six limitations (no new ones), and a factor considered as both a benefit and a limitation (new), as explained in Section 2.2.

Stage IV – in this stage, we applied techniques from meta-ethnography [28] to synthesize the findings of the previous three stages. The synthesis produced a consolidated set of 19 factors divided into eight benefits (1 new for the previous studies), nine limitations, and two factors considered as both benefits and limitations. During the final steps of the meta-ethnographic synthesis, we looked at the scientific literature on work design and organizational psychology to refine the meanings of these factors and raise construct validity, as recommended by Eisenhardt [8].

In the rest of the section, we describe the research methods used in each stage. We finish the section describing ethical issues considered in our study.

3.2. Stage I – qualitative case study

In this study, we followed the method proposed by Eisenhardt [8] to build theories from case study research. In this section, we present a summary of the research protocol to make this article complete. The detailed description of the case study protocol can be consulted in [34].

3.2.1. Getting started

We started with the definition of the research question and the case study design. We performed a broad literature review in several research fields to better understand and to precisely define job rotation (Sections 2.1 and 2.2), which together with the interviews with managers supported the characterization presented in Section 2.3. Further, this review aimed at identifying potentially important variables to be observed in the field, and to increase construct and internal validity during data analysis and synthesis. Based on this information, we selected a case, a well-established, mature software company, in which the job rotation practice was performed throughout all projects and teams, and all employees knew about the application of this practice.

As recommended in the literature [8] [36] [45], we used multiple data collection methods: interviews, document analysis, and questionnaires. The semi-structured interviews were performed with two groups of participants (using different interview scripts): a) the senior managers of the company, to collect data about the organizational context; b) project managers and software team members, to obtain information about their experience with, perceptions about, and attitudes toward job rotations. We validated the interview scripts by conducting pilot interviews with a group of five professionals of different companies, who had prior experience with P2P rotations.

3.2.2. The study context

We performed Case I in a Brazilian software company, founded in 1996. At the time of this case study, the company had just over 500 employees, about 70% of which worked directly in software development. The company executed an average of 50 projects concurrently in various business areas. The company had a typical hierarchical organizational structure. The software development division was headed by the Chief Operation Officer (COO) and supported by a Project Management Officer (PMO). Both COO and PMO were in charge of resource allocation for all projects and also job rotation (they had the role of the MCR agent, as described in Section 2.3), with support from the Human Resources department.

3.2.3. Entering the field

Field research was performed in three steps. We first selected the target projects and the individual participants within each project, then we interviewed the project managers and members of the project teams, and finally, we accessed documents to complement and triangulate the data collected in the interviews.

We purposively sampled two projects in the organization portfolio seeking for a maximum variation of information. We were seeking projects with different levels of incidence of rotations. We, thus, asked the Project Management Officer (PMO) to select a project with a great incidence of rotations (Project A) and another one with a more stable team (Project B):

- Project A – this team was developing a web-based system for a multinational logistics company. At the time of data collection, the project had been running for 2.5 years. The project team was composed of 13 professionals. One project manager, four software engineers, one test engineer, and one technical team leader were interviewed in this research. Project A used an Agile process based on SCRUM.
- Project B – this team was developing a 3D visualizer for a multinational printer company. This was an innovation project with the objective of creating and implementing new products. One project manager, four software engineers, one test engineer, and a technical team leader were interviewed in this research. Project B also used an Agile process based on SCRUM.

Table 2
Profile of participants – case I.

Projects	PA	PB
Team roles	4: Software engineers 1: Test engineer 1: Team leader 1: Project manager	4: Software engineers 1: Test engineer 1: Team leader 1: Project manager
Genders	6: Males 1: Female	6: Males 1: Female
Ages	3: <25 3: 26 – 35 1: >36	1: <25 4: 26 – 35 2: >36
Education	3: Undergraduates 1: Technician 2: B.*Sc.* 1: M.*Sc.*	1: Undergraduate 3: B.*Sc.* 3: M.*Sc.*
Time in the job	1: <3 years 4: 3 – 5 years 2: >6 years	2: <3 years 3: 3 – 5 years 2: >6 years
Experience with rotation	3: Never rotated 2: Rotated 1 or 2 times 2: Rotated >3 times	1: Never rotated 4: Rotated 1 or 2 times 2: Rotated >3 times

This diversity was important for understanding the different perceptions concerning benefits and limitations of job rotation. For project B, to achieve the necessary diversity we selected all team members. For project A, just under half of the team provided the required diversity.

We then selected participants from each project using purposeful sampling methods to achieve large variation in data collection, as recommended in the literature on qualitative research [26]. Therefore, we selected participants with different team roles, gender, age, education level, and time in the job, and distinct experiences in participating in rotations. In Table 2, we only present aggregated profile of participants to comply with ethical norms regarding anonymity of the participation.

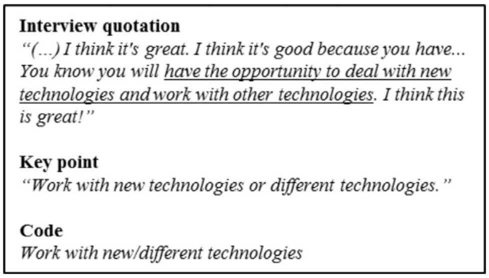
All interviews occurred in the organization's facilities and were performed by an interviewer and supported by a second researcher (that took notes to enhance data analysis). All interviews were recorded with permission from participants.

3.2.4. Data analysis

The objective of the qualitative analysis is to consolidate, reduce, and interpret data obtained from various sources, and make sense of them [26,36]. It involves labeling and coding all data to identify similarities and differences to describe the phenomenon under study. We used coding techniques [36] to code, categorize, and synthesize the data collected from the field.

Initially, all audio from the interviews was verbatim transcribed. Data analysis began with open coding of these transcripts, in which codes were constructed and attached to particular direct quotes of participants (Fig. 3). Then, the codes arising from each interview were constantly compared to codes in the same interview and from other interviews and grouped into categories that represent benefits and limitations of job rotation (Fig. 4).

In this article, we only used the results from open coding from Case I to identify benefits and limitations of P2P rotations. Santos



Interview quotation

"(...) I think it's great. I think it's good because you have... You know you will have the opportunity to deal with new technologies and work with other technologies. I think this is great!"

Key point

"Work with new technologies or different technologies."

Code

Work with new/different technologies

Fig. 3. Open coding - example.

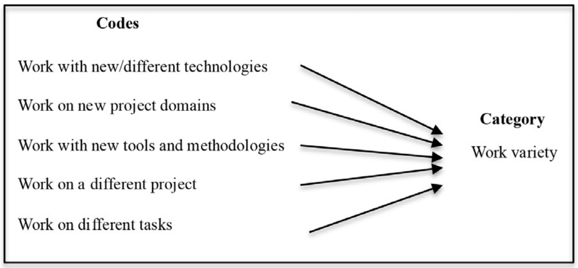

Codes
Work with new/different technologies
Work on new project domains
Work with new tools and methodologies
Work on a different project
Work on di fferent tasks
Category
Work variety

Fig. 4. Open coding - building categories.

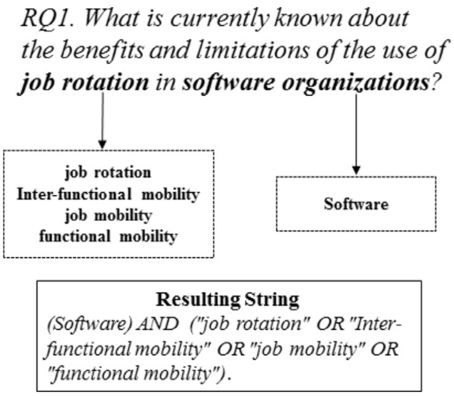

RQ1. What is currently known about the benefits and limitations of the use of job rotation in software organizations?
job rotation
Inter-functional mobility
job mobility
functional mobility
Software
Resulting String
(Software) AND ("job rotation" OR "Inter-functional mobility" OR "job mobility" OR "functional mobility").

Fig. 5. String construction.

et al. [34] presented the entire data analysis process, including axial and selective coding, together with propositions and a central narrative that explains the interacting effects of benefits and limitations in the studied context.

3.3. Stage II – systematic literature review

We followed the conceptual work on systematic literature review [31] and the guidelines for performing this type of study in software engineering [22]. In this section, we present a summary of the review protocol, which can be seen in full in Santos et al. [35].

We used the following research question to guide our review processes:

> *RQ1. What is currently known about the benefits and limitations of the use of job rotation in software organizations?*

3.3.1. Search process

We performed an automatic search in five search engines and indexing systems (IEEE Xplore, ACM, Scopus, Science Direct and Springer) using a search string based on the general terms extracted from our research question (Fig. 5). The automated search process, performed until July 2015, retrieved over 4000 papers. The set of synonyms for job rotation added to the search string, together with the use of just one term to specify the research field (software), increased the sensitivity of the search (thus increasing coverage) but also decreased its precision, i.e., a number of non-relevant studies found in the automatic search.

3.3.2. Selection process

The pre-selection of papers was based on the analysis of the full text of all the papers retrieved by the automatic search. Two researchers, working independently, excluded those papers that either addressed other areas besides computer science or only referenced papers about job rotation without actually addressing job rotation in their findings. At this point, sixty-three potentially relevant studies were pre-selected.

Next, we excluded studies addressing topics of computer science that were not related to software engineering, or papers that did not present discussions concerning the practice of job rotation in the context of software engineering. Duplicates were also excluded at this stage. When a study had been published in more than one journal or conference, all versions were reviewed for the purpose of data extraction. The selection process concluded with 18 papers, reporting 17 unique studies.

3.3.3. Data extraction and data synthesis

An extraction form implemented in MS Excel^{TM} guided the data extraction process. Two researchers, working independently, analyzed each paper in order to extract the following data: Paper title, year of publication, type of publication (journal or conference), country where the authors were located, main objective of the research, study method (according to [7]), instruments to collect data used in the study, type of participants of the study, definition used by authors to define job rotation, main application of the practice of job rotation reported in the study, benefits and limitations of the use of job rotation.

Two researchers working in the data extraction process were implied to improve the accuracy of the information extracted and, therefore, the reliability of the results. Conflicts of extraction were discussed and solved in a consensus meeting, which involved a third researcher.

Finally, we used the following steps to synthesize data:

(1) *Identifying Factors*: we used qualitative coding techniques [26,36] to identify work-related factors that were potentially impacted by job rotation in each study. We compared the findings between studies to make sure they were addressing the same construct. We performed an analysis similar to open coding as described in Section 3.2.4.
(2) *Grouping Factors*: similar to the use of axial coding in qualitative analysis, we compared and grouped the factors in three levels related to the: organization, workgroup or team, and individual.
(3) *Creating Propositions*: finally, using selective coding techniques, we performed an analysis of factors and categories to find relationships among them, and to present these relationships as propositions.

In this article, we use results from steps (1) and (2) to create a list of benefits and limitations of job rotation. Santos et al. [35] present the complete results of all three steps.

3.4. Stage III – extension and replication

This stage consisted of two parts: Case I – Extension and Case II – Replication. We started by extending the analysis of the data collected in Case I. We performed open coding on the transcriptions of all interviews. In Stage I, our analysis concentrated on identifying factors related to motivation and satisfaction. In this stage, we extended this analysis to look for all factors related to P2P rotation.

The extended analysis used the same coding techniques described in Section 3.2.4.

The second case study (Case II – Replication) was performed in the first half of 2016. We used the replication logic described by Yin [45] to develop Case II based on the same protocol used in Case I (Section 3.2). Consistently with the first case study protocol, we purposively sampled two other projects, one with a great incidence of rotations (Project C) and another one with a more stable team (Project D):

- Project C – this team was developing a web-based system for a multinational company. Details about the type of system were not provided by interviewees, due to confidentiality issues. At the time of data collection, the project has been running for 3 years, and the team was working on the development of new features, fixing bugs, and working on the performance improvements. This team was composed of 11 professionals: One project manager, one technical team leader, one designer, two test engineer and six software engineers. Project C used an Agile process based on SCRUM.
- Project D – this team was working on a research project, investigating algorithms to perform an image-based search. The team worked in identifying these algorithms in the literature and implementing them to validate the results with the client. At the time of data collection, the project has been running for three years and the team was composed of one project manager, one technical team leader, and five software engineers. Project D also used an Agile process based on SCRUM.

We then selected participants from each project using purposeful sampling methods, as in Case I. In Table 3, we only present aggregated profile of participants to comply with ethical norms regarding anonymity of the participation.

3.5. Stage IV – data synthesis

To synthesize the findings, we applied phases 2 to 5 of meta-ethnography as described by Da Silva et al. [6], summarized as follows:

3.5.1. Phase 2: deciding what is relevant for the synthesis

In this phase of the meta-ethnography, we decided the studies that would be included in the synthesis. We used the results of the previous Stages I to III as data for the meta-ethnographic synthesis.

Table 3
Profile of participants – case II - replication.

Projects	PC	PD
Team roles	3: Software engineers 2: Test engineer 1: Designer 1: Project manager	5: Software engineers 1: Team leader 1: Project manager
Genders	6: Males 1: Female	5: Males 2: Female
Ages	3: < 26 2: 26 – 35 2: > 36	1: < 26 4: 26 – 35 2: > 36
Education	2: Technician 4: B.*Sc*. 1: M.*Sc*.	1: Technician 5: B.*Sc*. 1: PhD
Time in the job	0: < 3 years 4: 3 – 5 years 3: > 6 years	1: < 3 years 1: 3 – 5 years 5: > 6 years
Experience with rotation	2: Never rotated 2: Rotated 1 or 2 times 3: Rotated > 3 times	0: Never rotated 6: Rotated 1 or 2 times 1: Rotated > 3 times

3.5.2. Phase 3: reading the studies

This phase involved carefully going through the findings from each study to identify the key concepts addressed in the studies through repeated reading and noting of the main concepts. The results from the three studies were already presented in tabular format, which simplified the work in this phase and also reduced the likelihood of misinterpretations.

3.5.3. Phase 4: determining how the studies are related

In this phase, we considered the relationships between the different studies. It was the first phase in which we put the studies together and started comparing them. Noblit and Hare [28] suggest juxtaposing the lists of key metaphors, ideas, and concepts occurring in each study as the initial step in relating the studies. We identified two different types of factors to guide our work in this phase:

- *Factors addressed by two or more studies:* these were used as input to the translations performed in Phase 5.
- *Factors addressed in only one study:* for these factors, we could not perform translations. Therefore, these factors were used in Phase 5 and compared with existing concepts from the literature (Sections 2.1 and 2.2) to sharpen construct definition.

3.5.4. Translating the studies into one another and raising theoretical level

During this phase, we started translating the concepts from one study into the concepts of the other ones for those concepts that were addressed in at least two studies. As prescribed in the literature [6,28], we started by considering studies as analogies, i.e. considering that findings in one study are like findings in the other studies, but also taking into account their non-similarities. An adequate translation preserves the meanings of concepts in each study. It also compares the meanings of concepts from one study with those from the other accounts. In general, concepts from the studies can compare with each other in three ways: (i) they are directly comparable as *reciprocal* translations; (ii) they may contradict or stand in opposition to one another as *refutational* translations; (iii) taken together they may represent a *line-of-argument*. In our synthesis, we only found cases of reciprocal translations.

Table 4 shows an example of how the translation of the concepts **Task Variety** and **Skill Variety** was built. From left to right, columns 1–4 present the concepts extracted from each study used in the synthesis (in this case, work or task variety was identified in all four studies). The fifth column presents the concept definition from the literature. In the example, the concepts of Task Variety and Skill Variety were extracted from the work of Morgeson and Humphrey [27]. The sixth column shows the synthesis of the concept that aggregates the results of all studies, checked for consistency with the concept definitions from the literature (column 5).

In Table 4, the reader can see how the use of meta-ethnographic translations allowed the researchers to refine the meanings of the concepts when comparing the studies. Case I and SLR identified concepts related to Work or Task Variety. Comparing these two studies showed that the concepts were similar and, thus, supported synthesis through reciprocal translation.

As we added results from Case I – Extension and Case II – Replication, which provided richer qualitative information, we noticed that participants were describing two different types of variety. First, some participants referred to variety related to performing different software engineering tasks, such as requirement analysis and coding, which is consistent with the interpretations of Task/Work Variety from Case I and SLR (labeled Work Varity (T) where T stands for Task). Second, participants also referred to variety related to use of different skills and technologies and to work

Table 4
Translation of concepts.

Case I	SLR - SwEng (P2P)	Case I (Extension)	Case II (Replication)	Concept from literature	Synthesis of the concept
Work variety "(…) It's really good (job rotation) because one can **work with several different things**." [PB-IN11] "(…) I think this is great (job rotation)! I think it's good because you know you'll have the opportunity to **work with new and different technologies**." [PA-IN02]	**Task variety** "Job rotation may help to increase the **variety** and challenge of IS development **work**." [JOB11] "Job rotation strategies could accommodate the different individual aspirations related to task variety." [JOB08]	**Work variety (T)** "I would be like, 'man, **there is no other thing (task) for me to do**? I'm doing this for such a long time'…" [PB-IN08]	**Work variety (T)** (…) **I can't work for too long doing the same thing**. So, I need to be moving (job rotation). [PD-IN26]	**Task variety** "Task variety refers to the degree to which a job requires employees to perform a wide range of tasks on the job. As such, it is similar to notions of task enlargement discussed in the literature" [27].	**Task variety** In software engineering, Task Variety refers to the scenario in which software engineers can perform a wide range of tasks. Thus, Task Variety is related to the role performed in the project. For instance, a developer that is assigned to perform tests activities or an analyst that also team leader.
"…, **there is no other thing (task) for me to do**? I'm doing this for such a long time'…" [PB-IN08]		**Work variety (S)** "(…) Good (job rotation). Especially if this rotation allows you to **work with a different technology**. [PA-IN04] "**I don't want** to spend the rest of my life **working with computer graphics** (same technology always)." [PB-IN12]	**Work variety (S)** "(…) (job rotation) allows one to **know different types of projects, different types of technologies** … [PD-IN27]	**Skill variety** "Skill variety reflects the extent to which a job requires an individual to use a variety of different skills to complete the work [17]" [27].	**Skill variety** In software engineering, skill variety refers to the diversity of different skills that the software development process requires from professionals. Thus, Skill Variety is related to the variety of technology or process related issues. For instance, the ability to work with backend and with interfaces, or the ability to fit in different domains using new technologies.

on different application domains (labeled Work Variety (S) where S stands for Skill).

At this point of the synthesis, we searched the literature about work design looking for definitions of work characteristics that would explain these two types of variety. The study of Morgeson and Humphrey [27] consolidated over 50 years of research on work design and their model distinguishes Task Variety (a task characteristic of the work) from Skill Variety (a knowledge characteristic of the work). The definitions of these two concepts, presented in the fifth column, are consistent with the two types of variety described by the participants. We then added Skill Variety to the list of factors related to job rotation.[1]

Finally, we provided definitions for each factor synthesized from the five studies. We used the definitions from the literature (column 5) and compared with the coding of the concepts resulting from the studies (columns 1–4) and created a definition of the concept that synthesizes our findings in column 6. In Appendix A, we present the complete glossary with the definitions of factors.

3.6. Ethics

In the development of Case I and Case II – Replication we followed the norms of Resolution 466/12 – CNS-MS of the Brazilian National Health Council that regulates research with human subjects. The software organizations signed a Term of Authorization, and the researchers signed a Non-disclosure Agreement (covering access to sensitive information). Both documents granted the researchers access to facilities, to the participants, and to necessary documentation. They also authorized the participants to use work hours for the interviews. We believe that this formalization reduced the possibility of participants concealing information that they would consider sensitive. Before the interviews, each participant signed an Informed Consent Form that explained the overall objective and relevance of the research, guaranteed data confidentiality, the anonymity of the participation, the non-obligatory nature of the participation, and the right to withdraw from the research at any moment. All invited individuals freely agreed to participate and no participant withdrew from the research.

4. Results

We start by describing the results of each study. We summarize the findings of Case I (Section 4.1) and the SLR (Section 4.2). We then describe the novel results related to Case I – Extension (Section 4.3), Case II – Replication (Section 4.4), in more detail. Finally, we present the results of the meta-ethnographic synthesis, which consolidates the benefits and limitations of P2P job rotation in software engineering practice, in Section 4.5.

4.1. Case I

Participants in Case I identified work-related factors that were directly affected by job rotation according to their perceptions. We analyzed the potential effects of job rotation on each factor and how participants perceived the importance or impact of each factor on their work (Table 5). Using the rationale explained in Section 2.2, we built column 4 indicating benefits or limitations of job rotation.

Job rotation provided software engineers the opportunity to work with a wide diversity of projects and technologies. Therefore, job rotation helped to create a working environment that was rich

[1] It is clear that Task and Skill Variety are related because to perform a task one requires certain skills, but these are different work characteristics. The study of their relationships is not in the scope of this study, but is an important theme for further research.

Table 5
Benefits and limitations – case I.

Factor	Correlations with factor	Impact of factor on work	Benefit/ Limitation
Team factors			
Knowledge transfer	+	–	Limitation
Work characteristics			
Task characteristics			
Work variety	+	+	Benefit
Well-defined work	–	+	Limitation
Knowledge characteristics			
Knowledge acquisition	+	+	Benefit
Social characteristics			
Feedback	–	+	Limitation
Outcomes			
Job outcomes and correlates			
Workload	+	–	Limitation
Performance	–	+	Limitation

in **Work Variety**,[2] which was positively valued by software engineers:

> *"(...) It is very good because one is going to work with several different stuff."*

Participants also perceived the lack of variety at work as being negative, reinforcing the positive value of Work Variety:

> *"... I can't stay in the same project for long."*

Job rotation also created an opportunity for the rotated person (RP) to acquire new knowledge (**Knowledge Acquisition)**, both technical and business related, increasing professional experience and skills regarding technologies, process, tools, business domains, etc. Knowledge acquisition was also perceived as positive for the job, emphasizing the constantly changing nature of the technologies and practices in software engineering:

> *"(...) I think it is good because you know you will have the opportunity to learn."*

Job rotation could happen at any point in the life cycle of a project. Therefore, individuals could be transferred to a new project before completing their assignments in the source project. Two effects of this characteristic of the job rotation practice were identified. First, participants emphasized that when they were rotated and left the assignment unfinished, they still had to perform some activities in their previous project, mostly to transfer knowledge about their prior tasks:

> *"You will keep supporting the previous project. So you will have to help that person on what you were doing"*

Job rotation, thus, creates the need for **Knowledge Transfer** and the RP perceived no positive effect on her work. A second effect of rotating a person before the end of the project or the person's assignments was the perception that they were not developing a **Well-defined Work**. Participants emphasized that they liked to perform complete tasks, i.e., from start to end, before moving to another project:

> *"I would not say it [job rotation] would be good. Because it is good when you do something and finish it".*

> *"During the on-going project, if I'm moved, then I would be mad, because I'll get that feeling that my work remains unfinished."*

Participants felt that job rotations could negatively impact the **Feedback** provided by managers and team leaders about their performance. They felt that managers and team leaders did not have enough information about their performance or capabilities at work because they were frequently moving between projects, as emphasized by this participant:

> *"... There are some guys who had to wait too much time to get a good feedback because they were always switching between projects".*

Finally, participants pointed out that job rotation had a potential negative impact on the **Performance** of all three individuals directly involved in the rotation (RP, RPH, and RPR described in Section 2.3,). Therefore, job rotation had a potentially negative effect on performance, as emerges from the following quotes:

> *"It is always traumatic [job rotation]. You end up having loss of performance"* [point of view of the RP].

It is possible to contend that there are several complex interacting effects among the factors potentially influenced by P2P rotations. In fact, Santos et al. [34] studied some of these interactions and presented relationships among these factors. Discussing these interactions is not in the scope of this paper, but constitutes an interesting topic for future research.

4.2. SLR

The SLR searched for evidence about how job rotation affected the work of software engineers in practice. In this section, we summarize the results directly related to the benefits and limitations of P2P rotations, which is the focus of this paper. Table 6 summarizes the factors potentially affected by the rotations reported in the primary studies. The references [JOBnn] are presented at the end of the list of references in this paper.

The SLR identified six factors related to the use of P2P rotations. All correlations were direct, meaning that the use of rotations was related to the increase of the factor. Four factors were identified as having a positive impact on the job and two as having negative ones. Using the rationale described in Section 2.2, we identified four benefits and two limitations of the use of P2P rotations in the primary studies analyzed in the SLR.

Case I and the SLR have only one benefit in common: *Task/Work Variety*. The two studies agreed on the direction of the relationship, considering Task Variety as a potential benefit of P2P rotations. Further, Task Variety was the only work characteristic found in the SLR, indicating that software engineering studies have not focused on individual and social work characteristics, contrasting with Case I and also with the literature from other areas discussed in Section 2.2.

[2] Notice that at this point we had not yet produced the meta-ethnographic translations that separated Work Variety into Task and Skill Variety.

Table 6
Benefits and limitations – SLR.

Factor	Correlation with factor	Impact of factor on work	Benefit/ Limitation
Organizational factors			
Communication (*new*)	+[JOB01]+[JOB03] +[JOB12]+[JOB17]	+	Benefit
Difficult to plan (*new*)	+[JOB15]	–	Limitation
Time consuming (*new*)	+[JOB09]	–	Limitation
Team factors			
Team flexibility (*new*)	+[JOB14] +[JOB15]	+	Benefit
Knowledge exchange (*new*)	+[JOB09]+[JOB12]+ [JOB13]	+	Benefit
Work characteristics			
Task characteristics			
Task variety (Case I)	+[JOB02]+[JOB08]+[JOB11]	+	Benefit

Table 7
Benefits and limitations – case I – extension.

Factor	Correlation with factor	Impact of factor on work	Benefit/ Limitation
Organizational factors			
Communication (SLR)	+	+	Benefit
Team factors			
Knowledge exchange (SLR)	+	+	Benefit
Knowledge transfer (Case I)	+	-	Limitation
Social conflicts (*new*)	+	–	Limitation
Work characteristics			
Task characteristics			
Work variety (Case I/SLR)	+	+	Benefit
Knowledge acquisition (Case I)	+	+	Benefit
Well-defined work (Case I)	–	+	Limitation
Social characteristics			
Feedback (Case I)	–	+	Limitation
Social interaction (*new*)	+/-	+	Benefit/Limitation
Outcomes			
Individual outcomes			
Motivation (*new*)	+	+	Benefit
Job outcomes and correlates			
Job monotony (*new*)	–	–	Benefit
Cognitive effort (*new*)	+	–	Limitation
Workload (Case I)	+	–	Limitation
Performance (Case I)	–	+	Limitation

Comparing the results of Case I and the SLR, we devised new directions for our research. First, to perform a broader analysis of the data collected in Case I trying to identify factors that were found in the SLR but were missed in Case I due to the focus on motivational aspects. Second, to produce further primary evidence from new empirical studies in software engineering to increase the strength of the evidence.

4.3. Case I – Extension

We performed an extended analysis of the data collected in Case I, in order to identify further factors related to P2P rotations in addition to those directly related to motivation. This process resulted in seven new factors not presented in Case I. Among these seven factors, two were found in the SLR and five are new findings from this extended analysis: two new benefits, two new limitations, and one factor classified as both benefit and limitation. We grouped the factors into the same categories defined in Section 2.2. Table 7 shows all factors; we marked (Case I) for those found in Case I, (SLR) for those found in the SLR and (new) for the new finding from Case I – Extension.

The extended analysis shows that the participants see job rotations positively related to *Motivation*, because the movement among projects can give them new opportunities for personal development, dynamism, and new challenges. Further, some of the factors identified so far, e.g., work variety and acquisition of knowledge, are related to motivation according to existing motivation theories [12,17].

> *"(…) The dynamism (of rotations) is positive to the motivation, because it brings new stuff."*

Job rotation is also perceived as a practice negatively related to *Job Monotony* at work, which might help to reduce boredom and other factors that can demotivate the individual concerning the job.

> *"(…) I liked it (job rotation). Because it changes my everyday work, it changed what I normally do."*

> *"(…) Normally, people like change. In my opinion, most of the people like when things change."*

On the other hand, there are new factors associated with the practice of job rotation identified in the extended analysis that resulted in limitations. One of these factors is the *Cognitive Effort* required to equalize the need for new knowledge required for the new assignments.

> *"(…) My knowledge about the project was too little, so I had to work hard on that (to fulfill that lack of knowledge after the rotation)"*

> *"The rotated person will have to study really hard for like… a month, or two months, to start to understand the project."*

This and other factors potentially affected by the use of job rotation can cause internal conflicts in the project. Interviewees reported that the frequent movement of individuals could result in

Table 8
Benefits and limitations – case II - replication.

Factor	Correlation with factor	Impact of factor on work	Benefit/Limitation
Organizational factors			
Communication (SLR/Extension)	+	+	Benefit
Difficult to plan (SLR)	+	–	Limitation
Team factors			
Knowledge exchange (SLR/Extension)	+	+	Benefit
Team flexibility (SLR)	+	+	Benefit
Work characteristics			
Task characteristics			
Work variety (SLR/Case I/Extension)	+	+	Benefit
Knowledge acquisition (Case I/Extension)	+	+	Benefit
Well-defined work (Case I/Extension)	–	+	Limitation
Knowledge characteristics			
Specialization (*new*)	–	+/-	Benefit/Limitation
Social characteristics			
Feedback (Case I/Extension)	–	+	Limitation
Social interaction (Extension)	+	+	Benefit
Outcomes			
Individual outcomes			
Motivation (Extension)	+	+	Benefit
Job outcomes and correlates			
Job monotony (Extension)	–	–	Benefit
Cognitive effort (Extension)	+	–	Limitation
Workload (Case I/Extension)	+	–	Limitation
Productivity (Case I/Extension)	–	+	Limitation

Social Conflicts among professionals in the team, as exemplified below:

> *"Man, you try and try, and try to help, but the person doesn't want to be helped. (…) So, most of the times we just don't care about her."*

> *"(…) Sometimes the team is not so friendly because they were used to work with the other guy."*

Finally, the *Social Interaction* can be positively or negatively related to the rotations, in the sense that individuals can start, keep or drop contact with co-workers when they are moved among projects, as shown below:

> Positive - *"You'll start to work in a new project, with new people, so you end up starting new relationships. I think this is important."*

> Negative - *"It's negative, when you lose contact with people that you used to work with."*

In summary, the extended analysis of Case I resulted in a broader view of factors related to P2P rotations that were not directly related to motivation of satisfaction factors.

4.4. Case II - Replication

In the analysis of Case II – Replication, we performed open and axial coding on the interview transcripts looking for factors related to the use of P2P rotations using no pre-formed codes. This replication found 15 factors, one of them was new, and 14 had been found in the other previous studies (Case I, SLR, and Case I – Extension). In Table 8, we show all factors found in Case II – Replication, identifying benefits and limitations as previously done.

The results of the replication were consistent with all 14 factors found in the previous three studies, increasing the strength of the evidence. Only three factors previously identified, were not found in the replication: Time Consuming, Knowledge Transfer, and Social Conflicts. The potential impact of P2P rotations on job specialization appeared as a new factor in the replication, which we further discuss below.

Participants emphasized the negative relationship of the use of P2P rotations and job specialization, consistent with findings from the literature [19]. All participants that mentioned job specialization in the interviews made it clear that rotations would (potentially) decrease the opportunity for the individuals to specialize in a given type of task, technology, or business domain. The difference among participants was with respect to whether this impact was perceived as positive or negative for their work. For those participants that have a positive attitude toward becoming a specialist, this was perceived as having a negative impact on their work, as exemplified below:

> *"There are people that don't want to be moved. You know, some people just prefer to be a specialist."*

> *"If you are comfortable with what you are doing (being a specialist). Maybe this is not good (rotation)."*

> *"You keep moving, you never get to fully know or understand anything, so in the end, you are not a specialist in anything."*

On the other hand, some professionals seem to have a different attitude toward job specialization, as emphasized by this participant:

> *"I don't think it's interesting when you become a specialist. (…) So the changes are very healthy."*

For those participants that did not value specialization, the impact of the P2P rotations on specialization was not perceived as negative, but as a potential benefit. We shall discuss the implications of this finding in Section 5.

4.5. Synthesis of results

In this section, we present the synthesis of the benefits and limitations of P2P job rotation in software engineering constructed using meta-ethnographic techniques (Section 3.6). In Table 9, we summarized the synthesis of all factors found in the five studies. Altogether, we synthesized 19 factors and classified them as potential benefits (8), limitations (9), and benefit/limitation (2) of the use of P2P rotations in software engineering practice. To our best knowledge, this is the most comprehensive study of the potential impacts of P2P job rotation in software engineering published so

Table 9
Synthesis of the evidence.

Factors	Correlation with factors				Impact of factor on work	Benefit/ limitation
	Case I	SLR - SwEng (P2P)	Case I (extension)	Case II (replication)		
Organizational factors						
Communication		+ Communication	+ Communication	+ Communication	+	Benefit
Difficult to plan		+ Difficult to plan		+ difficult to plan	–	Limitation
Time consuming		+ Time consuming			–	Limitation
Team factors						
Team flexibility		+ Team flexibility		+ Team flexibility	+	Benefit
Knowledge exchange		+ Knowledge exchange	+ Knowledge exchange	+ Knowledge exchange	+	Benefit
Knowledge transfer	+ Knowledge transfer		+ Knowledge transfer		–	Limitation
Social conflicts			+ Social conflicts		–	Limitation
Work characteristics						
Task characteristics						
Task variety	+ Work variety	+ Task variety	+ Work variety (t)	+ Work variety (t)	+	Benefit
Learning opportunity	+ Knowledge acquisition		+ Knowledge acquisition	+ Knowledge acquisition	+	Benefit
Task identity	- Well-defined work		- Well-defined work	- Well-defined work	+	Limitation
Knowledge characteristics						
Skill variety			+ Work variety (s)	+ Work variety (s)	+	Benefit
Specialization				- Specialization	+/-	Benefit/limitation
Social characteristics						
Feedback from others	- Feedback		- Feedback	- Feedback	+	Limitation
Social support – friendship opportunities			+/- Social interaction	+ Social interaction	+	Benefit/limitation
Outcomes						
Individual outcomes						
Motivation			+ Motivation	+ Motivation	+	Benefit
Job outcomes and correlates						
Job monotony			- Job monotony	- job monotony	–	Benefit
Cognitive effort			+ Cognitive effort	+ cognitive effort	–	Limitation
Workload	+ Workload		+ Workload	+ Workload	–	Limitation
Productivity	- Performance		- Performance	- Productivity	+	Limitation

far. In Table 11 (Appendix A), we present the definitions of each factor listed in Table 9.

This synthesis provides a broader view of the potential effects of P2P rotations than previous research in software engineering. Using three different types of research methods (systematic literature review, qualitative case studies, and meta-ethnography), we achieved two important results.

First, we enlarged the body of evidence, with different studies adding new evidence. This is clear when we analyze the results summarized in Tables 5-8. Starting with a set of two benefits and five limitations in Case I, the subsequent studies, including the meta-ethnographic synthesis, were progressively adding new factors until the final set of 19 factors was consolidated (Table 9). This shows the relevance of using mix-methods research and replication designs when investigating the complex phenomenon in the natural settings.

Second, we improved the strength of the evidence by confirming findings among the studies. The analysis of the results in Tables 5-8 also shows that, besides adding fresh evidence, new studies also confirmed previous findings. These confirmations happened across different types of studies:

- Between Case Studies and SLR: we confirmed the SLR findings with primary evidence from the qualitative case studies. In this sense, Task Variety, Knowledge Exchange, Team Flexibility, and Communication were identified as potential benefits in the SLR and confirmed by at least one of the case studies. Similarly, the potential limitation related to rotations being Difficult to Plan was also found in the SLR and confirmed in one case study. Only one factor found in the SLR was not confirmed by the case studies, namely P2P rotations being Time Consuming.
- Between Case Studies: by using a replication logic, we were able to produce confirmations for 12 of the total set of 14 factors found in Case I and Case I – Extension. Therefore, Case II – Replication produced the intended result of increasing the strength of the evidence with respect to the first case study.
- In the meta-ethnography: the process of translating evidence among studies only found reciprocal translations, in which the meaning of the concepts was consistently translated across the studies. No refutational translation was found, increasing the strength of the evidence in the synthesis stage.

Therefore, our findings show a very good strength of evidence, considering the consistency of findings among studies (reliability) and a number of different results with consistent findings (replications). We have only two factors with contrasting evidence (Specialization and Social Support), thus with low reliability, which could be explained by individual attitudes toward the factors, as discussed above. In those cases, the reader should note that the contrasting evidence was found within a single study, not across studies. Only three factors were found in a single study (low replication): Time Consuming, Social Conflicts, and Specialization. The remaining 14 factors were consistent in at least two studies, with six of them found in studies using different research methods, pointing to high reliability and replication.

Further investigation on the interacting effects of these factors is needed. Nevertheless, the strength of the evidence of these results indicates that we have a consistent and comprehensive set of factors that can be used in future research and also inform practice about the potential positive and negative effects of P2P rotations. We discuss these implications in the next section.

Table 10
Comparing findings with other fields.

Factors	Studies from other areas	This synthesis	Benefits/Limitations	
			Studies from other areas	This synthesis
Organizational factors				
Organizational commitment (not found)	+[18]		Benefit	
Communication	+[39]	+SLR/Case I/Case II	Benefit	Benefit
Time consuming	+[39]	+SLR	Limitation	Limitation
Team factors				
Knowledge exchange	+[39]	+SLR/Case I/Case II	Benefit	Benefit
Knowledge transfer	**+[39]**	**-Case I**	**Benefit**	**Limitation**
Work characteristics				
Task characteristics				
Task variety	+[21]	+SLR/Case I/Case II	Benefit	Benefit
Learning opportunity	+[21] +[2]	+Case I/Case II	Benefit	Benefit
Social characteristics				
Social support – friendship opportunities	+[21]	+Case I/Case II	Benefit	Benefit
Outcomes				
Individual outcomes				
Motivation	+[21]	+Case I/Case II	Benefit	Benefit
Satisfaction (not found)	+[18]		Benefit	
Career development (not found)	+[1]		Benefit	

5. Discussion

In this section, we discuss the findings of our research. We start comparing our findings with results from other areas. We then discuss the implications of our results for research and practice. Finally, we discuss limitations and threats to validity.

5.1. Comparing findings with other research fields

In Table 10, we compare the benefits and limitations found in the studies from other research fields (Section 2.2) with our findings. Those studies reported 11 benefits and limitations, and our study confirmed eight of them in the software engineering context.

Only one factor produced contradictory evidence between the two areas (highlighted in boldface in Table 10). Knowledge Transfer was considered a benefit in study [39] and a limitation in Case I. This could be explained because study [39] investigated rotations in the context of new product development (NPD) projects, for which Knowledge Transfer is likely to be a natural or even mandatory activity, whereas in software engineering this activity increased workload on the rotated person without a perceived benefit, as identified in Case I and discussed in detail by Santos et al. [34].

Three factors were not found in our studies: Organizational Commitment, Career Development, and Satisfaction. Regarding Satisfaction, one possible explanation is that individuals naturally conflate Satisfaction and Motivation, as discussed by França et al. [13]. Therefore, it is possible that P2P rotations were in fact related to Satisfaction, but our participants conflated this concept with Motivation in the interviews. In general, this could also indicate that the potential effects of job rotations are complex and results are likely to be strongly dependent on the type of job. This reinforces the need for studies, such as ours, focusing on the specific characteristics of software engineering practice.

5.2. Implications for research

In this study, we consolidated a comprehensive list of factors representing potential positive and negative impacts of P2P rotations in software development practice. However, the interacting effects of these factors have not been studied in empirical software engineering research. The studies analyzed in the SLR did not address job rotation as their primary goal. Our primary studies (Case I and Case II – Replication) showed important findings regarding the set of potential benefits and limitations, but we only performed a limited set of studies regarding the relationships among these factors [34]. We believe that this topic offers relevant and challenging problems to software engineering researchers.

The study of Hseih and Chao [19] reveals important issues to be considered in research in our field. They found that in high-technology industries, the design benefits (and also the limitations) of job rotation and job specialization might be quite different from those in low-tech industries, where job rotations were employed to reduce the negative effects of job specialization. They argue that in high-tech industries, fast technological advances and technical challenges inherent in the new technologies have made jobs much more professional and specialized, leveraging on specialist knowledge and expertise. These jobs have more task variety, task autonomy, task identity, and task feedback, than low-tech ones. Some of our findings support this view, as discussed in the results.

Therefore, according to Hseih and Chao [19] job specialization, not job rotation would be an effective way to increase satisfaction and lower the effects of job burnout. Further, the authors show that in high-tech jobs, job rotation has a negative relationship with task variety, contrary to what would be expected in past job design theories. Our findings only partially agree with their conclusions. We found that software engineers have contrasting attitudes toward specialization, with some individuals having a positive attitude toward becoming specialists and others preferring more variety regarding tasks or skills to experience. The effect of low and high specialization requires further investigation.

We must address the nature of the software development tasks concerning characteristics (autonomy, variety, identity, significance, and feedback). As discussed before, variety in software engineering could mean several different things: to work on another software system in a different business or application area (increasing Skill Variety), but keeping the same role, e.g. as a developer (without changing Task Variety); or changing the role from developer to tester (increasing Task Variety), but working on the same system (potentially without significant change in Skill Variety); changing role and business area at the same time (increasing both Task and Skill Variety). Further, it is clear that both types of variety are related, as one would potentially require new skills to perform other tasks. We need to investigate the distinct and interacting effects of these two types of variety.

Second, we must investigate the role of specialization in software engineering more deeply. We argue that P2P rotations could increase job specialization and tap the benefits of modern software development task design characteristics. Therefore, P2P could raise professional efficacy and decrease exhaustion, contributing to reduce the effects of job burnout as well as positively affecting motivational factors. However, we also showed that the attitude toward job specialization is not uniform among software engineers. Some individuals prefer to specialize, whereas others might prefer to work on a wider range of tasks and with different technologies and application domains. It is also possible that a certain level of skill specialization can be achieved even changing tasks, as discussed above.

Consistent with the complex nature of the phenomenon and a large number of factors involved, mixed-method research, including longitudinal case studies and replications, seems to be an appropriate methodological choice. Therefore, precise conceptual and operational definitions of constructs become important to integrate qualitative and quantitative findings, and generate consistent knowledge about the phenomenon.

Finally, research in software engineering needs to broaden its focus to address individual factors, in particular, those related to motivation and job satisfaction. In our SLR, only three studies addressed one motivational factor: Task Variety. Further, we also must broaden our focus on the outcomes of the job to include job burnout, voluntary turnover, loafing, absenteeism, etc. together with technical factors such as productivity and efficacy. Our case studies provided preliminary insights into these factors and their interacting effects, but further research is needed.

5.3. *Implications for industrial practice*

The number of potential benefits and limitations were large. Interacting effects between these factors may produce unexpected results. Therefore, understanding these potential effects is important in practice and our synthesis offers a starting point.

P2P rotations might be used to achieve organizational goals related to project resource allocation as well as individual motivational needs. However, it seems likely that addressing these needs might yield potentially conflicting outcomes that may not be straightforward to predict. Conducting empirical studies in industrial practice, in industry-research collaborative efforts, may provide relevant empirical evidence in specific organizational contexts.

The potential effects of P2P rotations seem to be moderated by individual characteristics. This was clear for Social Interaction, which was perceived to be positively and negatively related to the use of rotations by different participants. On the other hand, the clear negative effect on specialization was understood as a benefit for those with a negative attitude toward specialization and as a limitation to individuals that desire to be a specialist in some technology or task.

Our findings showed important distinction, in the software engineering context, between Task and Skill Variety. This distinction was also found in the literature on work design [27], but had not been investigated before in software engineering research. According to our findings, in software engineering Task Variety is associated with performing different types of tasks in the software development life cycle, whereas Skill Variety is related to the application of different technologies and work in different application domains, regardless of performing the same task. Both types of variety have positive motivational effect on individuals and are clearly interrelated. A P2P rotation can increase variety on one of the dimensions or in both. It may also not change the level of both types of variety. One key factor that must be evaluated when performing a rotation is the amount of variety (Tasks and Skills) involved. Little variety (moving a person to the same role in a project with similar technical requirements or to work on the same business domain applying the same technologies) creates less opportunity to learn and, therefore, would bring less direct benefits to the rotated person. On the other hand, too much variety may create a long and steep learning curve for the rotated person and increase the burden on the RPH, which could cause loss of performance and increase cognitive workload. It is desirable to plan a rotation taking the amount of variety into consideration whenever possible in practice.

Similarly, it is also important to take into consideration if the rotated person will need to still support the source project after the rotation. This creates the need of Knowledge Transfer that can increase cognitive workload and decrease performance. It is possible that creating practices to increase knowledge redundancy in the software teams will make rotations more effective by reducing the need of Knowledge Transfer. However, the literature does not show evidence about the effectiveness of knowledge redundancy to counteract the negative effects of job rotations.

Another important factor is the moment to perform a rotation. It is important to have a balance on the frequency of rotations of a given person. Rotations performed too often, and in the middle of the project development decrease, the perception of well-defined work (Task Identity) and might have a negative impact on Feedback from Others. On the other hand, leaving a person for too long in the same project is not desirable because software engineers value variety (task and/or skill), as shown by our synthesis of evidence.

In summary, certain desirable goals with the use of P2P job rotation can be conflicting, and some of these conflicts may not be reconciled due to practical reasons. Practitioners must be aware of these potential conflicts when planning rotations of personnel among software teams. Our synthesis provides a comprehensive list of potential factors to use in this planning.

5.4. *Limitations and threats to validity*

This study synthesizes the results of four different studies (three case studies, one of which being a replication, and a Systematic Literature Review). Reporting and synthesizing results of mix-method studies is always a challenge in terms of balancing the amount of information and the depth of explanation of any given study. Our choice in this article has been to increase the breadth of information about the evidence and sacrifice a little of the details regarding some methodological aspects of the studies that, in some terms, have already been published in other venues as summarized in Fig. 2. For instance, the case study protocol and data analysis are detailed in [34] and the SLR in [35]. We acknowledge that this may be a limitation as it makes the current manuscript not fully self-contained with respect to some methodological issues, but has at the same time, allowed us to give more detailed information about the pieces of evidence, which was the ultimate focus of the manuscript.

Another limitation of our study relates to the SLR, which currently includes studies published until July 2015. Our experience with secondary and tertiary systematic studies (including extensions and updates) has shown that yearly updates of SLR tend to produce very small changes in the evidence set and rarely (if ever) result in evidence that invalidates the previous findings. Moreover, on the one hand, a full-scale update on the systematic literature review presented by Santos et al. [35] would not be feasible considering the extra time and effort required for search, selection, and analysis phases compared to the added value of potential new sources (if any) identified. On the other hand, an ad-hoc addition of manually searched studies would be incorrect from a methodological point of view, as it may have introduced bias to the study. To the best of our knowledge, this is the first attempt to systematically review the literature on job rotation in software engineering research and provide results that are relevant to guide future research that is beneficial for industrial practice as well. We, therefore, acknowledge the limitation but are also confident that the evidence reported is valid and relevant.

We addressed threats to validity from an interpretative epistemological perspective, as discussed by Merriam and Tisdell [26]. In this perspective, *construct validity* is related to the precise and clear-cut definition of constructs that is consistent with the meanings assigned by the research participants. In our study, this notion of validity extends to the consistent meaning of the constructs across the different studies. To enhance construct validity in the case studies, we employed member checking to ensure our interpretations were consistent with those of the participants. We then compared and contrasted the initial construct definitions with those used in the studies analyzed in the SLR. We used the theoretical literature on work design to resolve ambiguities and inconsistencies. We used meta-ethnographic translations as a systematic method to perform these comparisons to achieve higher construct validity. Finally, we compared our findings from software engineering with the literature from other fields (Section 5.1), as recommended by Eisenhardt [8], also to sharpen construct definition.

Internal validity, or *credibility*, is related to the extent that the results match the meanings and knowledge constructed in the investigated context. To increase credibility in the case studies, we tried to achieve maximum variation collecting data from participants in different projects, with different roles, and with different perspectives regarding rotations. We also performed a replication of the first case study looking for further evidence. We then synthesized the results of primary studies with the findings from the SLR. This synthesis produced no contradictory evidence, increasing our confidence in the credibility of our findings.

Consistency refers to whether the researchers did not make any inference that cannot be supported by the data. To increase consistency, we performed all data analysis in pairs. In the case studies, coding was performed by one researcher and reviewed by two other researchers. Member checking with participants was also used to check the consistency of our interpretations. In the SLR, search, selection, extraction, and analysis were all performed in pairs and checked by at least one other researcher. Inconsistencies among researchers were resolved in consensus meetings. We used the Framework Method to enhance the consistency of data analysis among the researchers [14].

Finally, our case studies (Case I, Case I – Extension) and the replication (Case II – Replication) were conducted in the context of Brazilian software organizations. Different cultural practices and issues such as the legal framework that regulates job relationships might have influenced the results. Therefore, we do not claim generalization of our results to a large population in a positivist perspective. Instead, consistent with our interpretive perspective, we believe that the use of multiple sources of data and research methods supported good analytical generalization increasing the potential of *transferability* [26] of the findings to other contexts. In fact, we should keep in mind that the organizations are representative of software companies and are structured with roles and responsibilities comparable to companies of other countries. As summarized by the context details in Table 2 and Table 3, we can see how the profile of participants are heterogeneously distributed in terms of the factors such as team roles, genders, age, education, time in the job and experience with rotation. Being thus typical of any software organization, we are confident in concluding that the results are representative. Moreover, to further strengthen the results of these case studies, we are currently performing a replicated multi-country, mix-method (qualitative and quantitative) study involving software organizations from Brazil, Canada, and Italy in the attempt to apply similarity-based generalization [16].

6. Conclusions

We synthesized evidence from multiple sources (literature review, case studies, and replication) about the benefits and limitations of project-to-project job rotation in software engineering practice. Before our study, evidence related to P2P rotations in software engineering was restricted to studies that did not investigate this problem as their primary goal. Further, this evidence was highly concentrated on organizational and team level factors, with individual factors highly neglected.

In this study, we found eight benefits, nine limitations, and two factors considered both benefits and limitations of P2P rotations in software engineering. The different studies used in our synthesis yielded complementary and confirmatory evidence, emphasizing the importance of conducting mix-method research when the investigated phenomenon is complex and involves multiple variables. We found no contradictory evidence and six factors were identified in studies using different research methods (SLR, case studies, and meta-ethnography), contributing to the strength of the evidence.

Our findings show that P2P rotation is related to important benefits and limitations at different levels: organizational, workgroup, and work characteristics. It is also related to important outcomes such as motivation and productivity. The main limitations were associated with the potential increase in intra-group social conflicts and individual cognitive effort and a temporary decrease in individual productivity. We discussed that these factors have interacting effects that create a challenge for the effective application of P2P rotations in practice in previous sections (Section 5.3). We warned practitioners that they must be aware of these challenges and carefully plan rotations considering a broad set of factors.

We discussed the implications of our findings for software engineering research and proposed some directions for future work (Section 5.2). In particular, we believe that the study of a phenomenon such as job rotation, which involves multiple factors and their interacting effects, should be addressed by the use of multiple research methods. Further, the development of longitudinal studies is important to check the interplay between short-term negative effects and long-term benefits of the use of P2P rotation, as discussed by Santos et al. [34]. We believe that our synthesis provides a solid conceptual framework for future investigations.

Acknowledgments

Fabio Q. B. da Silva holds a research grant from CNPq #314523/2009-0. Cleyton V. C. Magalhães and Ronnie E. S. Santos are PhD students and receive a scholarship from CNPq. FACEPE also supported this research under the MSc scholarship IBPG-0651-1.03/12. We thank the reviewers of IST Journal for valuable comments and criticisms that greatly improved our work. Last but not least, we are very grateful to all participants in case studies I and II for dedicating their time and attention to our research.

Appendix A. – definitions of concepts

In Table 11, we present the definitions of each concept resulting from our synthesis of evidence. First column shows the name of the concept, the second column provides a definition from the literature, and the third column gives a definition of the concept in the context of software engineering.

Table 11
Definition of concepts.

Factors	Concept from literature	Definition
Organizational factors		
Communication	Communication is the process whereby individuals and groups transmit a variety of information within different areas of the organization using several techniques available [3] [38].	Communication is the process from which software engineers can interconnect to handle information or express ideas and thoughts along the organization. For instance, two developers of different projects talk about problems related to leadership in their workplace.
Difficulty to plan	Perceived task difficulty is characterized for the belief that a specific assignment requires much effort to be successfully done [43].	Difficulty to Plan refers to a task or assignment in software development that requires much planning effort, that is, a task that is not easy to be planned. For instance, the use of job rotation among software projects is a practice difficult to plan because there are too many factors involved to be considered.
Time-consuming	Following the Cambridge English Dictionary a time-consuming activity is defined as a task that requires an amount of time to be done.	Time-consuming is defined as a large amount of time required to execute a given task in software development.
Team factors		
Team flexibility	Flexibility refers to adaptation in response to changes in the work environment, that is, an ability to choose among different techniques in reaction to a range of changes [33].	Team flexibility is the degree in which a software team can respond to a variety of changes in the workplace. That is, how easy a team can adapt to variations in the environment, domain or in the scope of their tasks. For instance, a software team has to adapt their process due to a critical change in the main requirements of a system being developed.
Knowledge exchange	Knowledge exchange is the opportunity to share knowledge and learn from coworkers' experiences [12].	Knowledge exchange represents the opportunities that individuals have to share their knowledge and experiences to co-workers in the same project or in a different project at the organization, with the possibility of receive knowledge from those with whom his/her knowledge is being shared. For instance, developers from two different projects exchange information and share their experience on the use of a specific framework.
Knowledge transfer	Knowledge transfer refers to the process by which knowledge moves from one party of the organization to another. This process is characterized by the transference a specific understanding from one unit, e.g. individual, team or department, to another [44].	Broadly, knowledge transfer is an issue related to the transference of knowledge from one part of the organization to another. In software engineering knowledge transfer refers to the process by which an individual needs to transfer a specific understanding to another individual located in a different software project in the organization. This process is focused on giving information; thus, there are few opportunities for receiving knowledge (as in the exchange process). For instance, a designer who needs to act as a tutor of a group of designers from another project to help them with responsive interfaces.
Social conflicts	Social conflicts refer to relations in which two or more individuals believe they have incompatible objectives [23].	Social conflict is not one of the beneficial types of conflicts. This problem is characterized by two or more people having incompatible needs, goals or viewpoints, and this differences end up causing a non-friendly relationship behavior at work, e.g., two developers quarrelling over the time to perform a task.
Work characteristics		
Task characteristics		
Task variety	"Task variety refers to the degree to which a job requires employees to perform a wide range of tasks on the job. As such, it is similar to notions of task enlargement discussed in the literature" [27].	In software engineering, Task Variety refers to the scenario in which software engineers can perform a wide range of tasks. Thus, Task Variety is related to the role performed in the project. For instance, a developer that is assigned to perform tests activities or an analyst that also team leader.
Learning opportunity	Acquisition of useful knowledge is the knowledge that the individual believes that he/she can acquire as part of, or a reward for, a task execution, and which can be useful for their life, career, performance, etc. [12].	Learning opportunity refers to the knowledge and the understanding that an individual acquire at work. For instance, a developer that used to work with Java and learned C# while performs a specific task in a project, or a testers that learn issues related to configuration management process.
Task identity	Task identity reflects the degree to which a job involves a whole piece of work, the results of which can be easily identified [37] [27].	Task Identity reflects a well-define work. That is, the degrees to which software engineers perceive their work as a process with beginning, middle, and end. That is, the steps to be performed to complete their assignments and the expected results of this work can be easily understood to who will perform it. For instance, a software engineer that is allocated in a project during a whole sprint, or an analyst that work in a project during the whole process of software requirements, or a tester allocated to perform a test suite to validate a component.
Knowledge characteristics		
Specialization	Job specialization can be defined as the process whereby work activities become fragmented, simplified, and repetitive and limit the autonomy or discretion exercised by individual workers [40].	Specialization is related to the level of specific and specialized knowledge and skill that a given work requires to be performed. In other words, specialization requires deep understanding and expertise in a particular topic. For instance, a developer working with a specific framework, using a specific technology, for a long time.
Skill variety	"Skill variety reflects the extent to which a job requires an individual to use a variety of different skills to complete the work [17]" [27].	In software engineering, skill variety refers to the diversity of different skills that the software development process requires from professionals. Thus, Skill Variety is related to the variety on technology or process related issues. For instance, the ability to work with backend and with interfaces, or the ability to fit in different domains using new technologies.

Table 11 (*continued*)

Factors	Concept from literature	Definition
Social characteristics		
Feedback from others	Feedback from others reflects the degree to which others in the organization provide information about performance [27].	Feedback from others refers to the frequency in which managers, leaders, supervisors and co-workers give direct and clear information about the effectiveness of an individual's performance at work, e.g., a manager that provides to their team individual information about their performance at the end of a sprint.
Social support – friendship opportunities	Social interaction refers to the means of relationship building among members in a given community [20].	Social interaction can be seen as the process by which software engineers react and relate to others around him/her in the workplace. That is, how inter-relationships (friendship) among co-workers are created and maintained in a software organization.
Outcomes		
Individual outcomes		
Motivation	Motivation refers to "being turned on to one's work because of the positive internal feelings that are generated by performing well" [17]. Thus, motivation refers to the desire to work [12].	Motivation is a broad term used in the context of this research to describe the desire and the interest of software engineers to perform a task.
Job outcomes		
Job monotony	Job monotony refers to tasks or jobs characterized by repetitiveness or lack of variety [15].	Job monotony refers to de development of repetitive tasks, which reflect the lack of variety at work. For instance, a software tester that perform the same suite of tests in the same type of product for a long time.
Workload	Workload refers to the relationship of work demands with time and resources [25].	Workload refers to the amount of work to be performed, in particular when a specific individual received more work than he/her can handle in a given period. For instance, a developer needs to finish his/her daily assignments and the daily assignments of another developer, because this other person needs to go to a business travel.
Cognitive effort	Mental (cognitive) effort refers to the amount of capacity or resources that is actually allocated to accommodate the task demands [30].	Cognitive effort can be defined as the amount of mental effort required to perform a given task in the software development process.
Productivity	Work productivity refers to the capacity of a worker to produce good results through his/her occupation [10]. That is, the ability to conclude the activities just as (or better than) the plans [12].	Productivity is related to the extent that the development of a task in software development process is performed with effectiveness and quality.

References

[1] M. Alei, P.S. Shahrezaei, A theoretical structure for strategic human resource management in project oriented manufactures, Cumhuriyet Sci. J. 36 (4) (2015) 2088–2096.

[2] T. Brady, N. Marshall, A. Prencipe, F. Tell, Making sense of learning landscapes in project-based organizations, Third European Conference on Organizational Knowledge, Learning and Capabilities, 2005.

[3] Y. Brunetto, R. Farr-Wharton, Does the talk affect your decision to walk? A comparative pilot study examining the effect of communication practices on employee commitment post-managerialism, Manage. Decis. 42 (3/4) (2004).

[4] M. Campion, L. Cheraskin, M. Stevens, Career-related antecedents and outcomes of job rotation, Acad. Manage. J 37 (6) (1994) 1518–1542.

[5] P.F Coyne, An Evidence-based Model of Job Rotation, Middlesex University, London, UK, 2011.

[6] F.Q.B. da Silva, S.S.J.O. Cruz, T.B. Gouveia, L.F. Capretz, Using meta-ethnography to synthesize research: A worked example of the relations between personality and software team processes, in: ACM/IEEE international symposium on empirical software engineering and measurement, 2013, pp. 153–162. http://dx.doi.org/10.1109/ESEM.2013.11.

[7] S.M. Easterbrook, J. Singer, M. Storey, D. Damian, Selecting empirical methods for software engineering research, in: F. Shull, J. Singer, D.I.K. Sjøberg (Eds.), Guide to Advanced Empirical Software Engineering, Springer, 2008, pp. 285–311.

[8] K.M. Eisenhardt, Building theories from case study research, Acad. Manage. Rev. 14 (4) (1989) 532–550.

[9] T. Eriksson, J. Ortega, The Adoption of Job Rotation: Testing the Theories, University of Aarhus, Aarhus School of Business, Department of Economics, 2004 Working Papers 04-3.

[10] R. Escorpizo, Understanding work productivity and its application to work-related musculoskeletal disorders, Int. J. Ind. Ergon. 38 (2008) 291–297.

[11] T.E. Fægri, T. Dybå, T. Dingsøyr, Introducing knowledge redundancy practice in software development: experiences with job rotation in support work, Inf. Softw. Technol. 52 (10) (2010) 1118–1132.

[12] A.C.C. França, A Theory of Motivation and Satisfaction of Software Engineers, Center for Informatics, Federal University of Pernambuco, Recife, 2014.

[13] A.C.C. França, H. Sharp, F.Q.B. da Silva, Motivated software engineers are engaged and focused, while satisfied ones are happy, in: Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), Torino, Italy, 2014.

[14] N.K. Gale, G. Heath, E. Cameron, S. Rashid, S. Redwood, Using the framework method for the analysis of qualitative data in multi-disciplinary health research, BMC Med. Res. Method. 13 (2013) 117. http://www.biomedcentral.com/1471-2288/13/117.

[15] A.M. Game, Workplace boredom coping: Health, safety, and HR implications, Pers. Rev. 36 (5) (2007) 701–721.

[16] S. Ghaisas, P. Rose, M. Daneva, K. Sikkel, R.J. Wieringa. Generalizing by similarity: Lessons learnt from industrial case studies. International Workshop on Conducting Empirical Studies in Industry, 2013. Proceedings, art. no. 6618468, pp. 37–42. DOI: 10.1109/CESI.2013.6618468.

[17] J.R. Hackman, G.R. Oldham, Motivation through the design of work: test of a theory, Organ. Behav. Hum. Perform. 16 (1976) 250–279.

[18] W.H. Ho, C.S. Chang, Y.L. Shih, R.D. Liang, Effects of job rotation and role stress among nurses on job satisfaction and organizational commitment, BMC Health Serv. Res. 9 (1) (2009) 8.

[19] A. Hsieh, H. Chao, A reassessment of the relationship between job specialization, job rotation and job burnout: example of Taiwan's high-technology industry', Int. J. Hum. Resour. Manage. 15 (6) (2004) 1108–1123.

[20] H. Jang, L. Olfman, I. Ko, J. Koh, K. Kim, The influence of on-line brand community characteristics on community commitment and brand loyalty, Int. J. Electron. Commer. 12 (3) (2008) 57–80.

[21] K. Kaymaza, The effects of job rotation practices on motivation: a research on managers in the automotive organizations, Bus. Econ. Res. J. 1 (3) (2010) 69.

[22] B. Kitchenham, S. Charters, Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University, 2007.

[23] L. Kriesberg, Social theory and the de-escalation of international conflict, Annual Millennium Conference Lecture, London School of Economics, 1982.

[24] P.P. Kuijer, W.H. Vries, A.J. Van Der Beek, J.H. Van Dieen, B. Visser, M.H. Fringsdresen, F.M. Paul, et al., Effect of job rotation on work demands, workload, and recovery of refuse truck drivers and collectors, Hum. Factors 46 (3) (2004) 437–448.

[25] C. Maslach, S.E. Jackson, M.P Leiter, in: Maslach Burnout Inventory Manual, Consulting Psychologists Press, Palo Alto, CA, 1996, pp. 19–26.

[26] B.S. Merriam, E.J Tisdell, Qualitative Research: A Guide to Design and Implementation, Forth Edition, Jossey-Bass, San Francisco, 2016.

[27] F.P Morgeson, S.E. Humphrey, The work design questionnaire (WDQ): developing and validating a comprehensive measure for assessing job design and the nature of work, J. Appl. Psychol. 91 (2006) 1321–1339.

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

[29] A. Ollo-Lopez, A. Bayo-Moriones, M. Larraza-Kintana, The relationship between new work practices and employee effort, J. Ind. Relat 52 (2) (2010).

[30] F.G.W.C. Paas, J.J.G. Van Merrinboer, Instructional control of cognitive load in the training of complex cognitive tasks, Educ. Psychol. Rev. 6 (4) (1994).

[31] M. Petticrew, H. Roberts, Systematic Reviews in the Social Sciences, Blackwell Publishing, 2006 336.

[32] A. Richardson, M. Douglas, R. Shuttler, M.R. Hagland, Critical care staff rotation: outcomes of a survey and pilot study, Nurs. Crit. Care 8 (2) (2003) 84–89.

[33] R. Sanchez. Strategic flexibility in product competition. Strategic Manage. J., 16, pp. 135–159, 1995.

[34] R.E.S. Santos, F.Q.B da Silva, C.V.C. Magalhães, C.V.F. Monteiro, Building a theory of job rotation in software engineering from an instrumental case study, in: Proceedings of the 38th ACM/IEEE International Conference on Software Engineering (ICSE), Austin, USA, 2016.

[35] R.E.S. Santos, F.Q.B da Silva, C.V.C. Magalhães, Benefits and limitations of job rotation in software organizations: a systematic literature review, in: Proceedings of the 20th the International Conference on Evaluation and Assessment in Software Engineering (EASE), Limerick, Ireland, 2016.

[36] C.B. Seaman, Qualitative methods in empirical studies of software engineering, IEEE Trans. Softw. Eng. 25 (4) (1999) 557–572.

[37] H.P. Sims, A.D. Szilagyi, R.T. Keller, The measurement of job characteristics, Acad. Manage. J. 19 (1976) 195–212.

[38] A. Smidts, A. Pruyn, C. van Riel, The impact of employee communication and perceived external prestige on organisation identification, Acad. Manage. J. 44 (5) (2001) 1051–1062.

[39] K.E. Soderquist, G.P Prastacos, Knowledge transfer in NPD projects: lessons from 12 global corporations, European Conference on Organizational Knowledge, Learning And Capabilities, 2002.

[40] T.C. Taveggia, R.A. Hedley, Job specialization, work values, and worker dissatisfaction, J. Vocat. Behav. 9 (1976) 293–309.

[41] M.S. Viteles, Man and Machine Relationship: The Problem of Boredom, in: R.B. Ross (Ed.), Proceedings of the Annual Fall Conference of the Society for Advancement of Management, New York, 1950, pp. 129–138.

[42] R. Wood, Human Resources Management, AHMA, Michigan, 1995.

[43] R.E. Wood, Task complexity: definition of a construct, Organ. Behav. Hum. Decis. Process. 37 (1986) 60–82.

[44] R. Wijk, J.J.P. Jansen, M.A. Lyles, Inter- and intra-organizational knowledge transfer: a meta-analytic review and assessment of its antecedents and consequences, J. Manage. Stud. (2008) 830–853.

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

SLR Selected Studies – P2P rotations

[JOB01] M.A. Alnuem, A. Ahmad, H. Khan, Requirements understanding: a challenge in global software development, International Conference on Computer Software and Applications, 2012.

[JOB02] K.T. Ang, J.Y.L. Thong, C.S. Yap, It implementation through the lens of organizational learning: a case study of insuror, International Conference on Information Systems, 1997.

[JOB03] B. Birk, G. Heller, I. John, K. Schmid, T. Von Der Maßen, K. Muller, Product line engineering, the state of the practice, IEEE Softw. 20 (6) (2003) 52–60.

[JOB08] A.C.C. França, A.L.C. Felix, F.Q.B. da Silva, Towards an explanatory theory of motivation in software engineering: a qualitative case study of a government organization, International Conference on Evaluation and Assessment in Software Engineering, 2012.

[JOB09] C.D. Fowdar, S.D. Nagowah, Knowledge management practices in IT project based companies in an African country, International Conference on Computer & Information Science, 2002.

[JOB11] A. Kankanhallia, B.C.Y. Tan, K. Wei, M.C. Holmes, Cross-cultural differences and information systems developer values, Decis. Support Syst. (2004).

[JOB12] M. Kukko, N. Helander, P. Virtanen, Knowledge management in renewing software development processes, Hawaii International Conference on System Sciences, 2008.

[JOB13] C. Melo, D.S. Cruzes, F. Kon, R. Conradi, Agile team perceptions of productivity factors, Agile Conference, 2011.

[JOB14] N.B. Moe, T. Dingsøyr, T. Dybå, Overcoming barriers to self-management in software teams, IEEE Softw. (2009).

[JOB15] C. Passos, D.S. Cruzes, M. Mendonça, Applying theory of reasoned action in the context of software development practices, International Conference on Evaluation and Assessment in Software Engineering, 2013.

[JOB17] H.P. Siy, J.D. Herbsleb, A. Mockus, M. Krishnan, G.T. Tucker, Making the Software Factory Work: Lessons from a Decade of Experience, International Conference on Software Metrics Symposium, 2011.