Group learning: an analysis of individuals in software development teams who perform interdependent tasks

Letícia Rodrigues da Fonseca*

Faculty of Business Administration,
University of Vale do Rio Verde,
Unincor, Brazil
Email: leticia.rodrigues.vga@gmail.com
*Corresponding author

Marcelo Ribeiro Silva

Business and Administration School (Esan),
Federal University of Mato Grosso do Sul,
UFMS, Brazil
Email: marcelo.ribeiro@ufms.br

Sheldon William Silva

Faculty of Business Administration,
Federal University of Lavras,
Ufla, Brazil
Email: sheldonwilliamsilva@gmail.com

Guilherme Marques Pereira

School of Language Science,
University Vale do Sapucaí,
Univás, Brazil
Email: guilherme.marques@bol.com.br

Abstract: This paper aims to understand how group learning occurs within software development organisations (SDOs). The specific objectives are: 1) understand and describe the learning process in teams that perform interdependent tasks according to the addressed theoretical perspective; 2) understand and describe how developers apply the resources and infrastructures available in these organisations for their learning. Qualitative research of a multiple case study was conducted in four SDOs applying semi-structured interviews. The subjects interviewed were involved in the software development process. The analyses were performed with the qualitative data analysis software Weft_QDA. The results show that group learning does occur in SDOs through information and communication technologies, which allow developers to interact and share their knowledge. There is an asset repository that stores the knowledge acquired during the projects, equivalent to a transactive memory system (TMS).

Keywords: group learning; interdependent tasks; software development organisations; SDOs.

Reference to this paper should be made as follows: da Fonseca, L.R., Silva, M.R., Silva, S.W. and Pereira, G.M. (2021) 'Group learning: an analysis of individuals in software development teams who perform interdependent tasks', *Int. J. Innovation and Learning*, Vol. 29, No. 1, pp.89–105.

Biographical notes: Letícia Rodrigues da Fonseca is a Doctor in Business Administration, a Professor of two Postgraduate Programs (Master's) in Sustainability in Water Resources and Management and Planning and Teaching, both at the University of Vale do Rio Verde, Unincor, Brazil. Her research areas are organisational learning, knowledge management, management of information technologies, sustainability management, and innovative teaching practices.

Marcelo Ribeiro Silva is a Doctor in Business Administration, a Professor of undergraduate courses of Business Administration, Managerial Process, and Public Management, and a Professor of Postgraduate Program of Master's in Public Administration in the Federal University of Mato Grosso do Sul, UFMS, Brazil. His research areas are group learning, collective learning, organisational learning and public management.

Sheldon William Silva is a Doctorate student in Administration of Federal University of Lavras, UFLA, Brazil. He was also an Effective Professor of Basic, Technical and Technological Education at the Federal Institute of Northern Minas Gerais – Araçuaí Campus in the area of Administration and Business. His research areas are enthusiastic and user of active learning methodologies, and entrepreneurship in contents of the management area.

Guilherme Marques Pereira is a Doctor in Science of Language, and a Professor of Undergraduate and Postgraduate courses at the Varginha's Cenecista College. His research areas are agribusiness, financial management of organisations, investment management, and language sciences with an emphasis on discourse analysis.

1 Introduction

The Brazilian market for software and services has increased considerably in the last years. According to ABES – Brazilian Association of Software Companies (ABES, 2016), Brazil is considered the sixth-largest market in the world, with investments of US$152 billion. The domestic market, which includes hardware, software, and services, generated US$60 billion, representing 3.3% of the Brazilian GDP and 2.7% of total investments in information technology (IT) worldwide. From this amount, US$11.3 billion came from the software market, a growth of 30.2% compared to the previous year (2014). ABES has also identified about 13,950 companies in the national market that are dedicated to the development, production, and distribution of software.

However, the fierce competition is manifested in the same proportion as the opportunities. National software competes against each other and those from foreign countries, since there are not many trade barriers or legal restrictions against importing international products (Tonini et al., 2008). Moreover, this industry is distinguished by the rapid pace of evolution, which brings frequent changes to products, services, processes, and technology itself (Maurer et al., 2002; Niasi, 2009; Toniolo, 2011; Corniani, 2015).

According to Albertin (1999), the current economy denominated 'economy of knowledge' is based on the application of human knowledge to everything it produces and how it is produced. Knowledge is considered the most important element in the processes related to products and services, from their development to their delivery and support in the application. The added value is acquired through human intelligence instead of the physical effort of workers. Innovation, rather than access to resources or capital, becomes the critical factor, because in this new economy, being able to enter and remain in the market is difficult when the products have a competitive life of no more than one year, one month, one week, or a couple of hours, as it happens to financial products (Chuang et al., 2016; Elliott and Pedler, 2018).

Clients are becoming more demanding as they expect companies to provide better quality, differentiated products, and low prices. For this reason, the key assets in organisations will be those able to develop new products and services, meeting the expectations of the market and answering to the proposition of the new economy: make your own products obsolete before your competition does (Perez and Famá, 2015; Tidd and Bessant, 2015; Chuang et al., 2016).

It is believed essential for software development organisations (SDOs) to create a work environment conducive to continuous learning, even at the group level, in order to deal with the challenges of the new economy and remain competitive in the market. Software development is an individual activity, as its result depends on the particular way in which the developer learns how to apply his technological knowledge to transform the user requirement into a computational artefact. Nevertheless, it also depends on the collective efforts of a team with skills and competence that complement each other through interdependent tasks which will generate the final product (Tonini et al., 2008; Marks and Lockyer, 2004; Gondal and Khan, 2008; Lin, 2018; Raelin, 2018).

This problematisation led to the following research questioning: how do the members of a development team interact and share their expertise and conceive new knowledge from the analysis and combination of each other's expertise? How can the members of a development team access the knowledge generated in the group and employ it in the future?

In this context, the main objective of this study is to understand how group learning occurs within SDOs. The specific objectives are

1 understand and describe the learning process in teams that perform interdependent tasks according to the addressed theoretical perspective

2 understand and describe how developers apply the resources and infrastructures available in these organisations for their learning.

Software production is considered a socio-technical activity due to the interaction between developers with specific skills of development technology and methodology (Sawyer, 2004; Huckman et al., 2009). Thus, this study tried to analyse one of the group learning perspectives which explores the task mastery in teams with interdependent tasks to understand how developers learn at the group level to perform their duties (Edmondson et al., 2007). From this perspective, group learning is the result of the communication and the coordination that build the knowledge shared by the individuals about the team, the tasks, the resources and the context (Edmondson et al., 2007; Elliott and Pedler, 2018).

This approach studies how the group promotes learning to enhance the quantity and quality of their collection of knowledge available to perform the task. The focus is on the coding, storage, retrieval and communication of information between team members. Scholars who advocate this perspective seek to understand the relationship between team cognitive systems and team task performance, referring to team-level cognitive constructs with terms such as shared mental models, transactive memory systems (TMSs), social cognition, among others. All these terms are similar, as they describe cognitive systems present in the team that encode, store, retrieve and communicate knowledge, and are also applied to predict task performance (Edmondson et al., 2007; Silva et al., 2016).

Shared cognitions of a team emerge from both combination and compilation processes, and involve the exchange of information, ideas, knowledge and perceptions within the group over time. This exchange may occur through different means, such as social interactions, shared experiences about the task; formal interventions; among others. Shared cognitions provide the 'basis' for teamwork (Kozlowski and Bell, 2008; Silva et al., 2016).

According to Lewis et al. (2005), the performance of a team working with product development depends on the collaborative processes used by its members to combine and integrate their unique knowledge. Furthermore, Moreland and Myaskosvsky (2000) explains that as the members collaborate, they encode, interpret and remember information together and, as a result, the knowledge created is infused in the structures and processes of the team.

A TMS refers to a collective memory system for encoding, storing, retrieving and sharing the group knowledge (Hollingshead, 2001; Wegner, 1986). According to Kozlowski and Bell (2008), this concept was introduced to explain how close relationships promote the development of a shared memory. Each team member relies on the others as an external memory aid and, by doing so, becomes part of a more extensive system. For Lewis et al. (2005), these systems are developed over time, as the members communicate, observe their actions and come to rely on each other's ability to be in charge of different, and complementing tasks.

Members of teams that have developed a TMS remembered to collective apply a more significant amount of task-critical knowledge, to coordinate members' interactions more efficiently and had a higher performance when compared to teams without a TMS (Liang et al., 1995; Moreland and Myaskosvsky, 2000).

As Lewis et al. (2005) described, several laboratory studies were essential to bring the TMS concept and its effects to the attention of researchers and professionals. For these authors, one goal of those studies was to demonstrate how TMS enhances task-performance in tasks for which it was first developed. However, for Lewis et al. (2005), this approach does not include the fact that most organisational workgroups, as well as in SDOs, perform a variety of tasks, both in a single project and in subsequent flows of projects over time (Waller et al., 2002).

On the other hand, software development environment requires dynamic and highly complex activities, and time is one of the critical success factors (CSFs) because the market is multimillionaire and extremely competitive. Therefore, knowing the effects TMS has in an environment with dynamic tasks is essential to understand the real impact of such effects in SDOs.

At first, SDOs followed the traditional paradigm of software development known as 'life cycle', which was designed to provide a sequence of activities recognised as ideal ones to conceive a computational artifact and that did not allow alternatives, described as:

1 understanding the requirements
2 modelling
3 encoding
4 tests.

These activities are interdependent and include more than one person in the development process, and the input for the next phase depends on the result of the previous one.

After 'Agile Manifesto', many SDOs are choosing to follow this new development paradigm that puts more value in the interaction between software experts rather than in processes and tools applied in the task. According to this paradigm, there is no strict process of development, and developers are encouraged to learn how to innovate and make better decisions that allow superior results in the software development process (Chen and Lin, 2019).

An excellent example of this development practice is the technique called minimum viable product (MVP), which is derived from the Lean start-up methodology. MVP seeks the development of prototypes or reduced and simplified versions of a system to validate its viability with the customer, before delivering the final product.

This technique values the interaction and the learning by allowing the experts to develop new solutions during the product design process, better understand the customers' requirements, identify the processes that must be improved, in addition to establishing an environment with fewer uncertainties and delivering the final product in a shorter time (Tripathi et al., 2019).

For Tindale et al. (2008), team members work as information banks that, when interconnected, originate a more substantial database which can be accessed by all members at any given time. However, in order to make use of this information, team members must first recognise it as useful and share it, so the 'group unit abstracts it'.

Thus, group learning is a complex process, once it involves an individual, a common and coherent understanding, as well as a social context that motivates people to build and maintain a shared cognition (Zaccaro et al., 2008; Abbariki et al., 2017; Raelin, 2018).

A TMS is developed in a team when its members interact and discuss seeking to abstract and retain the knowledge about each other's skills and abilities (Abbariki et al., 2017). Coordinated and effective teamwork requires all its members to know 'who-knows-what' and 'who-can-do-what'. This knowledge stored in the TMS, which is interconnected with everyone in the group, allows the establishment of informal liability systems so that key individuals – specialists in performing specific tasks – can be summoned by the team to apply their skills whenever it is required (Edmondson et al., 2007; Wilson et al., 2007; Silva et al., 2016; Lin, 2018).

The interconnections between team members are usually established following the workflow. Learning tends to occur along with these interconnections, and the essential knowledge for the task is stored in and accessed from a TMS (Kozlowski and Bell, 2008; Silva et al., 2016).

According to Edmondson et al. (2007), a team with a functional TMS shows a better performance in their tasks by making fewer errors. Besides, the team can easily coordinate their tasks, once their members know precisely what to do and when to do it. The task is performed faster because there is no need for lengthy discussions to find a solution. The TMS also enables the team members to provide feedback to each other.

Identifying effective and ineffective behaviour can trigger an alert when a task is not performed correctly (Ellis et al., 2008). TMS allows information that is not valid for the task to be selected or rejected and make assumptions about how the environment will respond to certain actions based on the stored knowledge (Day, 1994).

Edmondson et al. (2007) also believe that some team characteristics can inhibit the development or functionality of a TMS. For instance, large-sized teams may not interact effectively. Moreover, task rotation can generate individual expertise in several functions, making interdependent tasks unnecessary.

The development of a TMS can also be affected when the team presents an inferior performance compared to its members in memory tasks, and when there is a tendency to discuss shared knowledge rather than personal knowledge, jeopardising the understanding and recognition of individual skills and abilities. Some context characteristics may also influence it. In a dynamic context, with great task rotation, individuals may demonstrate difficulty in retaining unutilised knowledge, whereas that may not occur in a stable context (Edmondson et al., 2007; Silva et al., 2016).

Therefore, work teams with interdependent tasks must cease to be a group of individuals in order to become a cohesive unit in which there are interaction and communication, so it is possible to gain a common understanding about the work process and the required knowledge to perform it and thus, resulting in learning at the group level (Sessa and London, 2008; Raes et al., 2015; Raelin, 2018).

2 Methodology

This research applied a qualitative approach, considering the problem and objectives proposed in this study. It seemed the most adequate since there are no reliable explanations to the proposed problem, hence the need to adopt an exploratory and descriptive approach (Godoy, 1995). For Berg (2004), the qualitative research answers questions by investigating social environments. It allows the researcher to share in the understanding and perceptions of the individuals who live in those environments, how to interpret people's behaviours and the meanings they attribute to the experienced situations.

This paper can be considered a multiple case study, when the researcher identifies the need to study several individual cases that keep an important correlation with each other in order to understand a phenomenon as a whole (Yin, 2006). Therefore, we can construct a more straightforward and useful typology by focusing on the theoretical (or descriptive) purposes or research objectives of a case study to describe, explain, or interpret a particular 'case' and which can be either inductive or theory-guided. They are explicitly structured by a well-developed conceptual framework that focuses attention on some theoretically specified aspects of reality and neglects others (Levy, 2008).

Benefits with a multiple case study are that the author can analyse the data within each situation and across different situations. The researcher studies multiple cases to understand the similarities and differences between the cases and therefore, can provide the literature with essential influences from its differences and similarities. Other benefits are that the evidence generated from a multiple case study is strong and reliable, and the author can clarify if the findings from the results are valuable or not (Gustafsson, 2017).

Four SDOs were studied following Yin's (2006) argument. In this author's view, even when the research is based upon two case studies, the possibility of replicating the results in contexts with similar conditions is higher than in a single case study. Table 1 brings a brief description of the companies participating in this case study, which have decided not to disclose their business names.

Table 1 Studied companies and their main characteristics

SDO	*Activity description*	*Main characteristics*
A	Offers information technology solutions: customised project development; financial market and public administration products; outsourcing.	Established in 1991, with regional branches in the southeast, south and northeast Brazil and clients in over 13 Brazilian states. Employs around 250 people.
B	Specialises in developing systems for academic management. Competitive advantage related to the professional management of educational institutions processes in a simple, but technological and efficient way.	Established in 1989, with branches in the south and southeast Brazil and clients including public and private educational institution. Employs around 150 people.
C	Its broad portfolio includes a software factory in southeast Brazil. It is certified in the CMMI (SEI) level 3 and ISO 9001-2000 (ABNT) standards to meet market demands.	Established in 1991, has clients from public and private sectors. Employs around 1.500 people, of whom 70 work in the software factory.
D	Provides services to the corporate and government market. Performs outsourcing projects in systems development and maintenance all over Brazil, through its software factory or outsourcing.	Established in 1994, with branches in the southeast Brazil. Employs around 200 people.

Table 2 Characterisation of interviewees

SDO	*Interviewed employee*	*Job description*	*Time with the company*
A	Developer 1A	Director of operations	4
	Developer 2A	Project manager	3
	Developer 3A	Requirement analyst	3
B	Developer 1B	Customer service team leader	7
	Developer 2B	Business specialist	8
	Developer 3B	Analyst and programmer	6
C	Developer 1C	SEPG manager	12
	Developer 2C	Project leader	13
	Developer 3C	Analyst and designer	8
D	Developer 1D	Technology manager	7
	Developer 2D	Analyst and programmer	5
	Developer 3D	Analyst and programmer	12

The authors interviewed professionals involved in the software production process of the participant companies who had been hired for at least one year to obtain information from people with a consolidated perception of their work environment. The developer interviewed with the least time at the SDO company had been working there for three years. Table 2 presents the characterisation of the developers who participated in this study.

It should be noted that this paper is part of a much larger study resulting from a doctoral thesis. Although the interviews were conducted in November 2011, the data was sent again to the researched companies in June and July 2016 to be updated and revalidated. Thus, new information was aggregated, and others disregarded when compared to the original data.

There were three interviewees in each SDO due to data saturation, i.e., when comparing the answers of the second interview with the first, there were too many similarities with few specifics. Therefore, three interviews in each organisation were enough to meet the objectives of this study.

The data collection method adopted was the semi-structured interview. This method was the only one applied because the studied companies did not allow the participant and non-participant observation nor the documentary analysis.

The interviews were transcribed by the authors using a text editor. Afterwards, the files *.txt were inputted in the Weft_QDA software, which processes qualitative data through four basic functions: organised data storage – in analytical or demographic categories elaborated by the researchers; data search and classification by categories: interview transcripts, field notes, documents, reflections or observations; data correlation established through several queries; search results available in texts or tables.

A thorough reading of transcripts enabled the identification of excerpts from texts that described how SDOs work teams learn to perform interdependent tasks at the group level through the coding method.

Coding is the process in which the data being analysed is defined, and it involves identifying and registering one or more pieces of text or other data, as parts of the general framework that, somehow, exemplifies the same theoretical and descriptive idea. Usually, several passages are identified and then related to an idea. Thus, all the text, among other elements, that refers to the same thing or exemplifies the same thing is grouped and coded with the same name. Coding is a way to index or categorise the text to establish a structure of thematic ideas about it (Gibbs, 2009).

3 Results and discussion

This research tried to identify and understand how the TMS is formed within development teams in the SDOs that participated in this study based on the interaction, communication, encoding, storage and retrieval of knowledge between their members, processes which characterise learning at the group level according to the theoretical perspective reviewed in this paper.

Regarding the research questioning that aims to understand how the members of a development team interact and share their expertise and conceive new knowledge from the analysis and combination of each other's expertise, the results determine that SDOs use information and communication technology to promote the interaction and the sharing of knowledge and experiences between developers. When software experts share their 'personal' meanings, the data they are based on are challenged, and the other team members examine the reasoning and logic that led to their conclusion. Learning occurs at this exact moment, as new knowledge is generated from the combination and analysis of personal meanings from the team members. This meaning, called 'collective', is the one shared by everybody, consisting of standards, strategies and assumptions that determine how the work must be done (Dixon, 1997, 1999; Baborska-Narozny et al., 2017; Lin, 2018).

It was possible to identify that developers learn collectively by using discussion forums, described as message boards, that allow their posting of doubts, problems and information, so that team members can offer solutions, support propositions of the participants and apply the stored knowledge regarded as useful in future situations which show the same context. There, all members of a group can read and discuss the subjects being addressed (Baborska-Narozny et al., 2017; Lin, 2018). In general, the organisation and proper functioning of virtual discussion forums depend on moderators or administrators who add, edit, delete or remove inappropriate topics and messages (Mill, 2006). According to developers' statements:

> We have some mechanisms here in the software plant for knowledge management. So, we have discussion forums. The analysts feed those discussion forums (Developer 1C).

> For example, today, we use a kind of forum. Well, there are some experts responsible for moderating those forums. There are forums, there, for Dot.Net, for Java, for the process. And there, you can post questions (Developer 2B).

> We have a mailing list that the team uses to communicate about topics related to our projects. We also use it to get tips on how to do something in the right way, in the best way. There is an exchange of information (Developer 2D).

Developers also learn collectively by using Wikis, websites fully editable in which users can read or add content. This functionality makes Wikis an excellent collaboration tool for the online environment. Several authors can gradually edit and update content in the Wiki site and, over time, this content becomes common knowledge to the entire group of collaborators at Wiki (Augar et al., 2004; Lin, 2018).

Wiki differs from discussion forums on the basis that it is a bank of web pages that works as an encyclopaedia. Wiki contributors add content about a specific subject, increasing the number of pages on this 'virtual book'. On the other hand, a forum allows its members to ask questions related to a specific topic in order to obtain answers and solutions from the participants, who can also discuss each other's answers according to their repertoire of knowledge and experience. In the developers' own words:

> So, we have a Wiki. The input is made by analysts (Developer 2C).

> There is a Wiki. The Wiki, I mean, follows the market standard. It is open so that anyone can contribute and all, read articles, add stuff, and it keeps growing (Developer 2A).

Furthermore, developers learn collectively by sharing their knowledge through the repositories that store the knowledge generated in the SDO. Take subversion, for instance, which is a free and open-source code system used to control the versions and manage files, directories and the changes made over time. Its core is recognised as a 'repository', where data is stored to be shared. The individuals with access to this repository can read, write on these files and share the stored information through the company's intranet (Collins-Sussman et al., 2007; Lin, 2018). According to some developers:

> This system has the same interface of a web portal or site that is hosted in the company's intranet, where we put/add all our documentation (Developer 2B).

> We have a factory web portal which, somehow, contains added knowledge (Developer 1C).

As for the questioning that seeks to understand how the members of a development team can access the knowledge generated in the group and employ it in the future, it was determined that SDOs contribute to the development of TMSs by establishing that the knowledge acquired during projects and the knowledge related to standard process are stored in a repository so that all developers can access it. Moreover, in order to learn how to execute their jobs, developers use the descriptions of the organisation's standard process stored in this repository as a source of knowledge:

> Look, speaking of our company, we have the process management website with the descriptions of the entire development process adopted by the company (Developer 1A).

The standard process can be defined as a group of settings for basic processes that guides all processes in an organisation. In an SDO, the standard software process refers to "a basic process that guides the establishment of a software development process which is common to all projects" [Couto, (2007), p.116].

For the development team to be able to perform their task according to the SDO's expectations, it is required that their members understand this standard software process, i.e., the set of activities, methods and practices used in software production.

The analysis of one of these documents stored in the repository which contributes to the teams' understanding of 'who-knows-what' and 'who-can-do-what' showed that it refers to the assignment mapping of each task:

> So, let's say like this, that today, the company hires someone. This person does not know where anything is in the company. If this person opens the site in the *SharePoint*[1], they can learn which tools they are going to work with, where to find the installation manuals for these tools and how to install them. Then, they do not have to wait for the IT guy to arrive with the equipment and install it, so they can start working. Just show them the website, its structure, and they can check it and start working (Developer 1B).

> From the project, you can usually see the documentation about your job. You do not even get to see all the documents of the project. However, you can see everything you need and what your job requires (Developer 2A).

The repository is used to store both the knowledge acquired during the projects and their 'lessons learned'.

SDOs consider errors as part of the learning process by determining that any errors made during the projects and their respective solutions are recorded in the document 'lessons learned', which is stored in the company's knowledge asset repository. This record contributes to the continuous improvement of projects as it allows developers to analyse it and learn with past mistakes in order to prevent them in the future.

Thus, before starting a new project, developers must identify if there are 'lessons learned' from similar past projects in the SDO's knowledge asset repository. After pondering over the mistakes previously experienced, it is possible to develop new ideas and, from them, create new schemes and theories to obtain better results in future projects. According to some developers:

> So, lessons learned, that is what we call them, are, in fact, problems found during the *Sprint*[2], whether it is about a deadline or lack of documentation or information, all of that is recorded. We put that in a repository so that we can go back in future projects and say: “Ah, they had problems with this, with that.” Then, this does not happen again (Developer 1B).

> The lessons are stored together, with the final Sprint, in this tool we have (Developer 1D).

> Ah, we have a document of lessons learned. It is available in the repository. We have a shared repository that we use nowadays. So, then, all the team has access to that documentation (Developer 2B).

> Every new project cycle we have this [...] It is a document that, actually, is in our repository of assets as the lessons learned in that project. You can make it either during the project or after you finish it. It is one of the artifacts that we create, there, during the project (Developer 1C).

The repository also allows the reuse of knowledge related to the best practices adopted in previous projects.

> This knowledge of best practices turns out to be described in the documents of process, models and all. And we end up having a more technical one, which are the tools we developed, the methods to perform specific tasks. So, all of that, let’s say, that could be applied in another project, is filed in this repository. However, for example, nowadays, we have… There must be, I do not know, about 12 projects designed, and everybody has access to any of them. Sometimes, someone is being considered for another project, and they have it there, they can look at the matrix, check the plan, the schedule, anything that may affect it (Developer 1A).

> We can refer to a knowledge database. And we can make suggestions for this knowledge database. We can reuse assets, that is like: “Ah, this was used in a project three months ago and it worked.” Someone goes there, check it, and it already has the assets and all the workflow set (Developer 1D).

> We have a repository of asset reuse in the company that receives input of this kind of information: the best practices (Developer 1B).

The knowledge acquired in past projects, especially during the process of solving problems, allows the creation of a frequently asked questions (FAQs), consisting of questions and answers on those problematic situations previously experienced that is available to all developers:

> We have the FAQ for solutions, and in fact, our ombudsman tool, our contact centre, is integrated into this FAQ. You can create an FAQ based on a problem that has been solved, I mean, then, you can store that knowledge. You also have the support staff that handles that very well, from problem solutions to best practices. And they also have a monthly newsletter where they send those tips, and those tips are stored in the FAQ too (Developer 1A).

These repositories, as well as the information and communication technologies previously mentioned, can be compared to a TMS. A TMS is developed in teams that perform interdependent tasks when their members interact and discuss with each other to abstract and retain the knowledge about each other’s skills and abilities. This interaction generates collective knowledge stored in the TMS, which is interconnected with all the team members so that each individual knows precisely ‘who-knows-what’ and 'who-can-do-what'. Thus, the team will be able to know who is supposed to act in certain situations, in addition to better coordinate task performance.

Therefore, the asset repositories contribute to the sharing of knowledge related to the standard process and to the developers' job, which allows the team to establish practical solutions in a short time. Furthermore, the storage and sharing of knowledge generated in past projects not only prevents the recurrence of errors but also allows the reuse of best practices.

4 Conclusions

Regarding the first specific objective that aims to understand and describe the learning process in teams that perform interdependent tasks, the results showed that this phenomenon occurs in SDOs mainly when the individuals interact by sharing information, reasoning and results through information and communication technologies. This kind of interaction, which allows the development of a collective knowledge referring to the standards, strategies and premises that determine how the task should be performed, will generate a repertoire of common knowledge similar to a TMS, as all the team members can access this knowledge that is stored in the shared 'memory' of the team.

Thus, all the members will know exactly 'who-knows-what' and 'who-can-do-what', ensuring the success of the task once the team will be able to know when a member should use his specific skills, when the task methodology should be modified for better performance, besides performing projects in a shorter time, as it will not be necessary to have lengthy discussions about what to do or who should be in charge.

As for the second specific objective, which aims to understand and describe how developers apply the resources and infrastructures available in these organisations for their learning, the results showed that the technological structure available, specifically collaborative information and communication technologies, are the primary resources applied in group learning, such as forums, Wikis, and data repertoires available in the intranet.

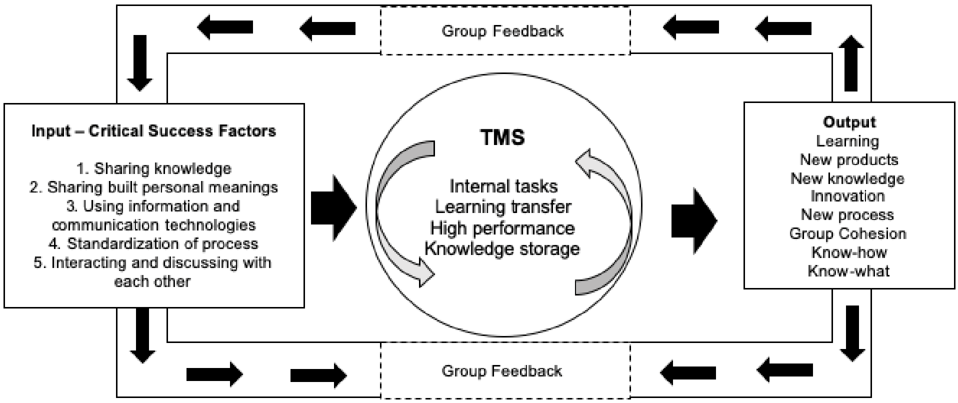
Therefore, this study has met its main objective by ascertaining that group learning occurs in SDOs when software experts share their individual knowledge, i.e., the knowledge their job is based on, besides the reasoning and the logic leading to their conclusions, and that are all analysed or even contested by the other team members. Learning occurs at this exact moment, as new knowledge is generated from the combination and analysis of personal meanings from the team members.

Moreover, the study results allowed the identification of CSF in these SDOs, which enhance the efficiency of the team learning process. As Allameh et al. (2014) explain, there is a great variety of reasons that can affect the relationship between the CSF in organisational learning, such as culture, IT, human resources management, and organisational structure.

The CSFs can be defined as "areas in which results will ensure successful affordable performance for the organization" (Rockart and Morton, 1984).

From left to right, the first rectangle represents the input of necessary elements to the learning process, here identified as the CSFs. Thus, the first CSF is knowledge sharing, which occurs when developers share their knowledge to make the team's knowledge about the task equal in order to obtain a satisfactory performance in the projects, once the activity of developing software must be performed in teams due to its task interdependency (Oumaya and Gharbi, 2017; Lin, 2018).

Figure 1 Group learning model



Source: Adapted from Fonseca et al. (2019)

The second CSF is about the fact that individuals build personal meanings that become accessible when they are shared through interaction. At that moment, the team assesses the reasoning and logic of those meanings, accepting them as valid or not. When a personal meaning is rejected, it is rebuilt by the team to generate better ideas and more effective actions. This joint reflection results in a collective meaning, shared by all team members. In organisations, collective meaning refers to the knowledge that determines how the task must be performed. It can be considered a ‘glue’ which allows the individuals to work together, in consonance with each other (Dixon, 1997, 1999; Adelopo et al., 2017; Abbariki et al., 2017; Oumaya and Gharbi, 2017; Raelin, 2018).

The third CSF can be identified when the team uses information and communication technologies that allow the interaction and sharing of knowledge. Through these technologies, developers express their ideas and experience with each other, contributing not only to the establishment of a collective meaning, but also to the storage and reuse of the ‘useful’ knowledge in future situations (Dixon, 1997, 1999; Baborska-Narozny et al., 2017; Lin, 2018).

The results showed that SDOs determine that both the knowledge acquired during projects and the knowledge related to the standard process are stored in a repository so that all developers can access them (Silva et al., 2016; Lin, 2018).

The main documents stored, and which contribute to the teams’ understanding of ‘who-knows-what’ and ‘who-can-do-what’, refer to the description of the standard process and the assignment mapping of each task.

The fourth CSF is related to the description of the standard process. It is believed they also contribute to the development of individual mental models of software experts. In individual learning, mental models are intimate images that allow people to understand how the world works. They establish which stored knowledge is relevant to a particular situation, influencing how people determine their actions and define their strategies for the environment in which they are.

A mental model is not a representation of reality itself, but rather how someone learns this reality, acting as a 'source-code' that manages the acquisition, retention, use and elimination of information and experiences (Kim, 1993).

For developers, the standard-process is the 'ideal one' to produce software. Thus, learning what must be done through those descriptions, the individual will be able to not only select and store the essential knowledge required to perform a task but also eliminate or discard those considered useless.

Therefore, a TMS is generated in software development teams from their use of information and communication technologies and data repositories that allow the interaction, communication, encoding, storage and retrieval of knowledge between their members.

Teams that perform interdependent tasks develop a TMS when their members interact and discuss with each other, thus becoming the fifth CSF, to abstract and retain the knowledge about each other's skills and abilities. Regarding the practical implications as to the contribution to the software industry, some developers still show a profile similar to the ones characteristic of the old teams that operated in the early days of programmed computer systems, described as learned people with highly individualist practices.

However, although software development is an individual activity, it also depends on the collective efforts of a team that will apply their different skills and knowledge in programming, design, database, and so forth, to create a computational artefact.

Therefore, the collaboration between developers is essential for the success of the projects, and in this context, SDOs must understand the factors conducive to group learning in the organisation for better performance. Furthermore, when consulting databases such as Science Direct, Scopus, Ebsco and Proquest, no empirical researches were aiming to understand how learning at group level occurs in SDOs based on the theoretical perspective addressed in this study, specifically in Brazilian SDOs.

As for the theoretical implications, the most significant and relevant contribution of this study refers to the proposed generic theoretical model, which can be applied in several SDO contexts, including in foreign countries.

References

Abbariki, M., Snell, R.S. and Easterby-Smith, M. (2017) 'Sharing or ignoring tacit knowledge? A comparison of collective learning routines at two sites', *Journal of General Management*, Vol. 42, No. 4, pp.57–67.

ABES (2016) *Data about Brazilian Software Market*, Brazilian Association of Software Companies [online] http://central.abessoftware.com.br-/Content/UploadedFiles/Arquivos/Dados%202011/ABES-Publicacao-Mercado-2015-digital.pdf (accessed 29 April 2016).

Adelopo, I., Asante, J., Dart, E. and Rufai, I. (2017) 'Learning groups: the effects of group diversity on the quality of group reflection', *Accounting Education*, Vol. 26, Nos. 5–6, pp.553–575.

Albertin, A.L. (1999) *Electronic Commerce: Model, Aspects and Contributions of its Application*, São Paulo, Atlas.

Allameh, S.M., Rezaei, A. and Bagheri, M.M. (2014) 'The mediating role of organizational learning between knowledge management success factors and organizational innovation: a conceptual framework', *Journal of Comparative Neurology*, Vol. 2, No. 2, pp.1444–8939.

Augar, N., Raitman, R. and Zhou, W. (2004) 'Teaching and learning online with Wikis', Paper published in the *Proceedings of 21st Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) Conference*, Perth, 5–8 December.

Baborska-Narozny, M., Stirling, E. and Stevenson, F. (2017) 'Exploring the efficacy of Facebook groups for collective occupant learning about using their homes', *American Behavioral Scientist*, Vol. 61, No. 7, pp.757–773.

Berg, B.L. (2004) 'Methods for the social sciences', *Qualitative Research Methods for the Social Sciences*, Pearson Education, Boston.

Chen, T.C.T. and Lin, C.W. (2019) 'An innovative yield learning model considering multiple learning sources and learning source interactions', *Computers and Industrial Engineering*, Vol. 131, pp.455–463 [online] https://www.sciencedirect.com/journal/computers-and-industrial-engineering.

Chuang, M-Y., Chen, C-J. and Lin, M-j.J. (2016) 'The impact of social capital on competitive advantage: the mediating effects of collective learning and absorptive capacity', *Management Decision*, Vol. 54, No. 6, pp.1443–1463.

Collins-Sussman, B., Fitzpatrick, B. and Pilato, C. (2007) *Version Control with Subversion: For Subversion 1.4*, TBA, California.

Corniani, A.A. (2015) 'Cultural impacts within an enterprise due to ERP software deployment', *Caleidoscópio*, Vol. 1, No. 6, pp.61–75.

Couto, A.B. (2007) 'CCMMI: integration of training models and systems maturity', *Ciência Moderna*.

Day, G.S. (1994) 'Continuous learning about markets', *California Management Review*, Vol. 36, No. 4, pp.9–31.

Dixon, N.M. (1997) 'The hallways of learning', *Organizational Dynamics*, Vol. 25, No. 4, pp.23–34.

Edmondson, A.C., Dillon, J.R. and Roloff, K.S. (2007) '6 three perspectives on team learning: outcome improvement, task mastery, and group process', *The Academy of Management Annals*, Vol. 1, No. 1, pp.269–314.

Elliott, T. and Pedler, M. (2018) 'Action learning', *Research and Practice*, Vol. 15, No. 1, pp.18–27.

Ellis, A.P., Porter, C.O.L.H. and Wolverton, S.A. (20078) 'Learning to work together: an examination of transactive memory system development in teams', in *Work Group Learning*, pp.115–140, Psychology Press.

Fonseca, L.R., Silva, M.R., Silva, S.W. and Pereira, G.M. (2019) 'Continuous-learning work environment: a study with developers in software development organizations', *Knowledge Management and E-Learning: An International Journal*, Vol. 11, No. 3, pp.281–303.

Gibbs, G. (2009) *Qualitative Data Analysis: Qualitative Research Collection*, Bookman Editora, Porto Alegre.

Godoy, A.S. (1995) 'Qualitative research: fundamental types', *Revista de Administração de Empresas*, Vol. 35, No. 3, pp.20–29.

Gondal, A.M. and Khan, A. (2008) 'Impact of team empowerment on team performance: case of the telecommunications industry in Islamabad', *International Review of Business Research Papers*, Vol. 4, No. 5, pp.138–146.

Gustafsson, J. (2017) *Single Case Studies vs. Multiple Case Studies: A Comparative Study*, Thesis, Halmstad University, Halmstad, Sweden.

Hollingshead, A.B. (2001) 'Cognitive interdependence and convergent expectations in transactive memory', *Journal of Personality and Social Psychology*, Vol. 81, No. 6, p.1080.

Huckman, R.S., Staats, B.R. and Upton, D.M. (2009) 'Team familiarity, role experience, and performance: evidence from Indian software services', *Management Science*, Vol. 55, No. 1, pp.85–100.

Kim, D.H. (1993) 'The link between individual and organizational learning', *Sloan Management Review*, Vol. 35, No. 1, pp.37–50.

Kozlowski, S.W. and Bell, B.S. (2008) 'Team learning, development, and adaptation', in Sessa, V.I. and London, M. (Eds.): *Work Group Learning: Understanding, Improving and Assessing How Groups Learn in Organizations*, pp.15–44, Erlbaum, Mahwah, NJ.

Levy, J.S. (2008) 'Case studies: types, designs, and logics of inference', *Conflict Management and Peace Science*, Vol. 25, No. 1, pp.1–18.

Lewis, K., Lange, D. and Gillis, L. (2005) 'Transactive memory systems, learning, and learning transfer', *Organization Science*, Vol. 16, No. 6, pp.581–598.

Liang, D.W., Moreland, R. and Argote, L. (1995) 'Group versus individual training and group performance: the mediating role of transactive memory', *Personality and Social Psychology Bulletin*, Vol. 21, No. 4, pp.384–393.

Lin, J.W. (2018) 'Effects of an online team project-based learning environment with group awareness and peer evaluation on socially shared regulation of learning and self-regulated learning', *Behaviour & Information Technology*, Vol. 37, No. 5, pp.445–461.

Marks, A. and Lockyer, C. (2004) 'Producing knowledge: the use of the project team as a vehicle for knowledge and skill acquisition for software employees', *Economic and Industrial Democracy*, Vol. 25, No. 2, pp.219–245.

Maurer, T.J., Pierce, H.R. and Shore, L.M. (2002) 'Perceived beneficiary of employee development activity: a three-dimensional social exchange model', *Academy of Management Review*, Vol. 27, No. 3, pp.432–444.

Mill, D.R.S. (2006) *Distance Education and Virtual Teaching Work: About Technology, Spaces, Times, Social Relations of Sex and Collectivity in the Media Age*, Doctoral dissertation, 252pp, Belo Horizonte: FAe/UFMG, Tese de doutorado em Educação.

Moreland, R.L. and Myaskovsky, L. (2000) 'Exploring the performance benefits of group training: transactive memory or improved communication?', *Organizational Behavior and Human Decision Processes*, Vol. 82, No. 1, pp.117–133.

Niasi, M. (2009) 'Software process improvement implementation: avoiding critical barriers', *CROSSTALK. The Journal of Defense Software Engineering*, Vol. 22, No. 1, pp.24–27.

Oumaya, S. and Gharbi, L. (2017) 'Individual and collective absorptive capacities of new external knowledge: the case of Tunisian small and medium-sized enterprises (SMEs)', *International Journal of Technology Management & Sustainable Development*, Vol. 16, No. 3, pp.209–227.

Perez, M.M. and Famá, R. (2015) 'Strategic characteristics of intangible assets and economic performance of the company', *Unisanta Law and Social Science*, Vol. 4, No. 2, pp.107–123.

Raelin, J.A. (2018) 'What are you afraid of: collective leadership and its learning implications', *Management Learning*, Vol. 49, No. 1, pp.59–66.

Raes, E., Kyndt, E., Decuyper, S., Van den Bossche, P. and Dochy, F. (2015) 'An exploratory study of group development and team learning', *Human Resource Development Quarterly*, Vol. 26, No. 1, pp.5–30.

Rockart, J.F. and Morton, M.S. (1984) 'Implications of changes in information technology for corporate strategy', *Interfaces*, Vol. 14, No. 1, pp.84–95.

Sawyer, R.K. (2004) 'Creative teaching: collaborative discussion as disciplined improvisation', *Educational Researcher*, Vol. 33, No. 2, pp.12–20.

Sessa, V.I. and London, M. (2008) *Work Group Learning: Understanding, Improving and Assessing How Groups Learn in Organizations*, Psychology Press, New York.

Silva, M.R., da Silva, C.C., da Fonseca, L.R. and da Silva, L.C. (2016) 'The transactive memory system and group learning', *Revista de Administração FACES Journal*, Vol. 15, No. 2, pp.47–65.

Tidd, J. and Bessant, J. (2015) *Gestão da inovação-5*, Bookman Editora, Porto Alegre.

Tindale, R.S., Stawiski, S. and Jacobs, E. (2008) 'Shared cognition and group learning', in *Work Group Learning*, pp.97–114, Psychology Press, New York.

Tonini, A.C., de Carvalho, M.M. and de Mesquita Spinola, M. (2008) 'Contribution of quality and maturity models in software process improvement', *Production*, Vol. 18, No. 2, pp.275–286.

Toniolo, C.M. (2011) 'Reuse of requirements for embedded systems product families', *Revista da Universidade Ibirapuera*, Vol. 2, No. 2, pp.31–35.

Tripathi, N., Oivo, M., Liukkunen, K. and Markkula, J. (2019) 'Startup ecosystem effect on minimum viable product development in software startups', *Information and Software Technology* [online] https://www.sciencedirect.com/journal/information-and-software-technology.

Waller, D., Montello, D.R., Richardson, A.E. and Hegarty, M. (2002) 'Orientation specificity and spatial updating of memories for layouts', *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 28, No. 6, p.1051.

Wegner, P. (1986) 'Classification in object-oriented systems', in *ACM Sigplan Notices*, ACM, Vol. 21, No. 10, pp.173–182.

Wilson, J.M., Goodman, P.S. and Cronin, M.A. (2007) 'Group learning', *Academy of Management Review*, Vol. 32, No. 4, pp.1041–1059.

Yin, R.K. (2006) *Case Study: Planning and Methods*, Bookman Editora, Porto Alegre.

Zaccaro, S.J., Ely, K. and Shuffler, M. (2008) 'The leader's role in group learning', in *Work Group Learning*, pp.15–44, Psychology Press, New York.

Notes

1 *SharePoint* is a platform for web applications developed by Microsoft, used to create portals and intranets to promote content management in companies.

2 *Sprint* is considered the main practice of *scrum*, where the scrum team implements the work items defined in the *product backlog*.