Knowledge management in the software industry: how Scrum activities support a knowledge management cycle

Gestão do conhecimento na indústria de software: como as atividades do Scrum dão suporte um ciclo da gestão do conhecimento

Nelson Tenório Doutor em Ciência da Computação. Pontifícia Universidade Católica do Rio Grande do Sul – Brasil. nelson.tenoriojr@gmail.com

Danieli Pinto Mestra em Gestão do Conhecimento nas Organizações. UniCesumar – Centro Universitário de Maringá – Brasil. danicne@gmail.com

Marcio José Silva Mestre em Gestão do Conhecimento nas Organizações. UniCesumar – Centro Universitário de Maringá – Brasil. marciocne@hotmail.com

Iara Carnevale de Almeida Doutora em Informática. Universidade de Évora – Portugal. iara.carnevale.almeida@gmail.com

Flávio Bortolozzi Doutor em Engenharia de Computação. Université de Technologie de Compiègne – França. flavio.bortolozzi.53@gmail.com

ABSTRACT

Knowledge is the primary asset of the software industry which increases the complexity of the software development activities. Knowledge Management (KM) is an important practice that offers processes namely KM cycles and addressed to knowledge capture, creation, storage, dissemination, sharing, and use. Thereby, understanding different means to consolidate individual's knowledge within the software industry is relevant to software development practitioners and researchers. Therefore, this article discusses how Scrum activities support KM cycle to convert the individual's knowledge into common knowledge within software development teams. Our analysis points out that KM cycle combined with Scrum practices is a potential tool to disseminate individual knowledge across software development team members.

Keywords: Knowledge processes; Software development; Agile methods.

RESUMO

O conhecimento é o principal ativo da indústria de software aumentando o que aumenta a complexidade das atividades desenvolvidas. A Gestão do Conhecimento (GC) é uma prática importante composta por processos, que forma os ciclos da GC e direcionam a captura, criação, armazenamento, disseminação, compartilhamento e uso do conhecimento. Desse modo, entender os diferentes meios para consolidar o conhecimento do indivíduo dentro da indústria de software é relevante para os profissionais e pesquisadores de desenvolvimento de software. Portanto, este artigo discute como as atividades do Scrum suportam o ciclo KM para converter o conhecimento do indivíduo em conhecimento comum dentro das equipes de desenvolvimento de software. Nossa análise aponta que o ciclo de GC combinado com as práticas do Scrum é uma ferramenta potencial para disseminar o conhecimento individual entre os membros da equipe de desenvolvimento de software.

Palavras-chave: Processos do conhecimento; Desenvolvimento de software; Métodos ágeis.

Recebido em 28/02/2019. Aprovado em 14/04/2019. Avaliado pelo sistema *double blind peer review*. Publicado conforme normas da APA.
http://dx.doi.org/10.22279/navus.2020.v10.p01-13.928

1 INTRODUCTION

Software development is essentially performed using agile methods. Agile methods aim to accelerate the development and maintenance of software products by establishing shorter deadlines, better quality results, risk mitigation and adding value to the final product (Dyba & Dingsoyr, 2008; Cámara et al., 2016). The use of agile practices allows software development teams to adapt themselves to the client's changing needs through the interaction and collaboration among stakeholders, which results in better project outcomes (Rola, Kuchta, & Kopczyk, 2016). The agile method offers an incremental and evolutionary process for the development of products (Pinoa, Pedreira, García, Luaces, & Piattini, 2010). Beck et al. (2001) emphasize that Scrum is one of the easiest agile methods to deploy, because it is simple and also because it has few control practices, ensuring transparency, inspection, and adaptation of the project (Lei, Ganjeizadeh, Jayachandran, & Ozcan, 2017; Schwaber & Sutherland, 2016). However, Scrum does not include technical practices, it emphasizes values and management practices only (Pinoa et al., 2010).

The software industry presents peculiar characteristics regarding other segments of the market. They differentiate themselves by developing intensive activities on knowledge and generating high added-value products (Bjornson & Dingsoyr, 2008; Chen, Ragsdell, & O'Brien, 2014). The globalization of markets and the intensive use of knowledge in the development of new products have contributed to the software industry becoming a high technology sector (Binuyo, Oyebisi, Olayinka, & Afolabi, 2015) making it complex (Nawinna, 2011). Thus, it is necessary and fundamental that people's knowledge is incorporated into the processes and activities of the organization, that is, one of KM's roles (Meihami & Meihami, 2014; Aurum, Daneshgar, & Ward, 2008). In this sense, for knowledge to be useful to the organization, it must be externalized and socialized among the individuals that compose it (Wiig, 1997). Therefore, efficiently managing knowledge resources in a software development environment is critical to maintaining organizational knowledge and promoting innovation (Aurum et al., 2008).

The KM literature presents cycles consisting of distinct processes of capture, creation, dissemination, sharing, storage, and use of knowledge (Dalkir, 2011). The cycles allow KM to be systematized, which contributes to both the implementation of organizational learning and to the use of knowledge, creating value and enabling organizations to become innovative and sustainable in the market in which they operate. However, companies that use Scrum for software development simplify their processes and amplify the focus on individual's tacit knowledge (Nerur, Mahapatra, & Mangalaraj, 2005). Thus, KM performed with Scrum can support knowledge transferring processes in software development outsourcing (Brito, Figueiredo, Venson, Canedo, & Ribeiro Junior, 2017).

Therefore, based on the results of our previous work, shown in Tenório et al. (2017), we present the following research question: 'How are Scrum practices is integrated to the KM cycle?' We understand that the Scrum methodology supports the incorporation of knowledge in the organizations' processes and it is, this way, directly related to the KM cycles' establishment activities and the execution of improvements. To answer our research question, we carried out an exploratory study through the literature regarding KM and the Scrum and comparing Scrum practices with the KM cycles.

This paper is important once it shows how Scrum practices can be used to systematize and incorporate the knowledge into the software teams, this way, not being used o only as a method of control. In addition, this research provides support (both to researchers and workers of the software industry) for deploying new practices and processes and creating new knowledge. This review will also help build a common understanding of the Scrum method and KM processes. Our findings point out that Scrum enables knowledge to flow among software development team members through a knowledge cycle based on the capture, creation, storage, dissemination, sharing, and use of knowledge. The results of such research will be relevant to the software industry.

The article is organized as follows: in Section 2, we give an overview of Scrum, we present the KM cycle and show the integration between KM and Scrum. Section 3 describes the methods used for this review. Section 4 reports the relation between Scrum events and KM cycles. Section 5 concludes and provides recommendations for further research on agile software development.

2 THEORETICAL BACKGROUND

2.1 Scrum

One of the major innovations in software development was the introduction of the agile methods that fostered process adaptation in a constantly changing environment (Vlaanderen, Jansen, Brinkkemper, & Jaspers, 2011; Highsmith & Consortium, 2002).

The "Agile Manifesto",established in 2001, allowed the introduction of important concepts based on (a) individuals and interactions rather than processes and tools; (b) running software rather than extensive documentation; (c) client collaboration rather than contract negotiation; (d) responding to changes rather than following a plan (Beck et al., 2016). These concepts incorporated new methods, tools, techniques and the use of the best practices in the software development processes. As a result, agile methods have introduced a new vision on how to develop software (Dingsøyr, Nerur, Balijepally, & Moe, 2012). This has differentiated them from traditional methods by having a set of values, principles, and practices with a major focus on people rather than processes (Cristal, Wildt, & Prikladnicki, 2008). In addition, agile methods can enhance productivity growth in software development teams (Melo, Cruzes, Kon, & Conradi, 2013), causing a positive impact on both productivity and quality, e.g., knowledge sharing, stakeholders' active participation, self-organizing teams, and reduced documentation (Ahmed, Ahmad, Ehsan, Mirza, & Sarwar, 2010).

Zhang and Dorn (2011) indicate that agile methods are more adaptive and flexible than the traditional ones because they are suitable for scenarios where requirements suffer constant changes and where results must be delivered in short time frames. It should be noted that agile methods are characterized by the division of the software development process into shorter cycles, with deliveries to the client (internal and external) at the end of each cycle, thus enabling the monitoring of changes to the requirements by the developers at the beginning of each cycle and therefore reducing project risks (Eloranta, Koskimies, Mikkonen, 2016).

Among agile methods, Scrum is one of the most used in the software industry. One of the main characteristics of this method is that it enables people to solve complex and adaptive problems, delivering quality products in a productive and creative way (Schwaber & Sutherland, 2016). According to Nerur et al. (2005), Scrum focuses on how team members should work in order to produce a flexible software system in a constantly changing environment. Its structure revolves around product management and development practices, dividing the software development process into equal phases (called Sprints) which include monitoring and feedback activities. These activities are carried out at daily and quick meetings with the whole team, which aims to identify and correct deficiencies and/or impediments during the execution of processes (Dönmez, Grote, & Brusoni, 2016). Therefore, Scrum is characterized by being an iterative, incremental and team-based method (Cervone, 2011).

Scrum is based on the collaboration among team members to incorporate value into the final product (Zhang & Dorn, 2011). According to Schwaber and Sutherland (2016), Scrum is based on three main cornerstones: transparency, inspection, and adaptation. Transparency aims to make the team aware of all project objectives in a uniform, clear and concise manner. Inspection is a way to ensure constant revisions in the process to avoid misuse of the scope. Finally, adaptation allows for emerging changes throughout the project to be analyzed and adopted.

Planning, preparation, work development, and delivery are the phases that guide the Scrum cycle (Schwaber, 2004). The planning phase situates the project and expectations among stakeholders; the preparation phase is responsible for establishing and prioritizing requirements; the work development phase implements the requirements established in Sprints, with the addition of functionalities at the end of each Sprint and the delivery phase executes the implantation of the system. Scrum also establishes fundamental practices regarding team members' roles, events, and artifacts necessary to accomplish what is called "Scrum Cycle", presented in Figure 1.

The roles define the necessary competencies for the project team, which consists of the Product Owner, Scrum Team, and Scrum Master. The Product Owner represents the interest of the team. Their function is to express with transparency what should be done, determining requirements, functionalities, objectives and the delivery plan (Marçal, Freitas, Soares, & Belchior, 2007). The Scrum Team is comprised of developers, designers, testers, and architects. It has the authority and responsibility for many aspects of the software development, including planning, scheduling, assigning tasks to stakeholders and decision-making to achieve the proposed goals (Moe, Dingsøyr, & Dybá, 2010). The Scrum Master plays the role of the facilitating leader of the process. It clarifies the progress of the project and propagates Scrum values to the entire team, ensuring the practice of daily meetings and maintaining team performance (Lei et al., 2017).

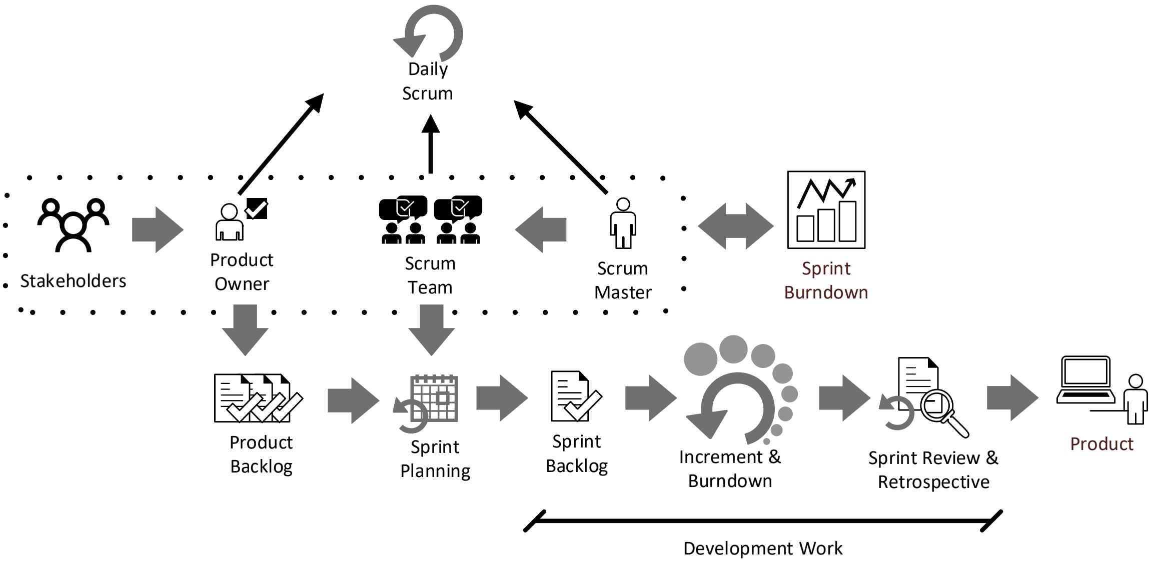

Figure 1. **Scrum cycle**

Source: The authors.

The events include the execution of Sprints, a key element of Scrum. Sprints allow for transparency and judicious inspection of the project. While the artifacts comprise project objectives and the design of Sprints, consisting of the Product Backlog, Sprint Backlog and product increment (Schwaber, 2016).

2.2 Knowledge Management Cycles

The KM cycles comprise a set of processes and practices that seek to create, collect, share and promote the dissemination of knowledge in the organizational environment in order to help people carry out their work and learn from one another (Ortegón, Lasso, & Steil, 2016). However, these processes cannot be performed without concrete goals, and they must be directed towards the organization. Thus, KM proposes a systematic process that takes place through the development of interrelated and interdependent stages. In this sense, it is considered that information in transit represents valuable knowledge which is called the KM cycle (Ortegón et al., 2016).

A KM cycle makes it possible to identify the route through which knowledge travels within an organization. This way, it becomes clear when a KM process is being executed and deserves more attention. A cycle can also follow an action that is related to knowledge, locating the extension of one or more activities through stages. It also enables processes or stages to be verified from the moment the knowledge is created until it is made available or used by the organization (Dalkir, 2011; Evans, Dalkir, & Bidian, 2014). According to Dalkir (2011), KM objectives are as follows: i) to describe how knowledge is used; ii) to raise awareness about the necessary KM activities; (iii) to reduce the complexity of processes; iv) to design a KM solution.

According to the Europe Committee for Standardization (Europe Committee for Standardization) (CEN, 2004), there are over one hundred and fifty models of KM cycles with similar objectives. Therefore, it is necessary to understand some of the KM cycles proposals, thus delimiting the processes contained in some of these approaches.

Goldoni and Oliveira (2010) propose a KM cycle based on the creation, storage, dissemination, utilization, and mediation of knowledge. The creation process consists of acquiring and adding new knowledge to the existing one. In the storage process knowledge is encoded and stored in knowledge bases. The dissemination process is responsible for the diffusion of knowledge within the organization when the use of the knowledge acquired is necessary. Finally, measurement is the process of evaluating the results achieved with the use of KM.

Table 1 shows the main KM cycles. It is observed that KM processes, carry out a cycle which enables the creation of new knowledge, available from the last stage. The processes of the cycles presented in Table 1 are similar to one another and very similar to the proposal of Goldoni and Oliveira (2010). However, each author considers a different situation when designing their cycle model. According to the analysis presented by Dalkir (2011) and Evans et al. (2014), the Wiig cycle predicts the existence in a business environment composed of three basic elements: business, resources and the ability to act. The authors also point out that the processes, contained in the Meyer and Zack cycle, are derived from the development of information products. Moreover, the Bukowitz and Williams cycle sketches its stages from the idea of how organizations generate and maintain a minimum pool of knowledge for value creation. Finally, the McElroy cycle is formed by processes formatted from a series of feedbacks, having greater participation of the individuals involved in the processes.

Wiig (1993)	**Meyer and Zack (1997)**	**Bukowitz and Williams (1999)**	**McElroy (2003)**
Build	Acquire	Get	Knowledge production
Hold	Refine	Use	Organizational knowledge
Pool	Store	Learn	Knowledge integration
Use	Distribute	Contribute	Distributed organizational knowledge base
	Present	Assess	Business processing environment
		Build/Sustain or Divest	

Table 1. **Main KM models and their stages**

Source: Adapted from Dalkir (2011)

Tenório et al. (2017) presents a grouping of KM processes in three stages: i) capture and creation, ii) dissemination and sharing; iii) acquisition and use. The authors propose that after knowledge has gone through the capture and creation process, it advances to the dissemination and sharing process. In this process, the evaluation of the knowledge function and usefulness to the organization is carried out. Moving on to the acquisition and use process, knowledge is incorporated into the organizational environment through a language which is common to all the ones involved. The authors indicate that when knowledge is used to create new knowledge, there is a KM cycle to it, and that cycle starts again. It is observed that although the cycles shown in Table 1 comprise different phases, all seek to demonstrate knowledge in an organized and systematized way. The purpose of it is to identify and make organizational knowledge common to all so that the benefits from KM are effective.

2.3 Integrating Knowledge Management Cycle and Scrum Activities

Agile methods influence communication, knowledge management, and the organizational structure, emphasizing feedback cycles, small development iterations, continuous planning and team self-organization, providing process transparency among stakeholders (Dönmez et al., 2016). Furthermore, the use of agile methods allows for the change of requirements throughout software development which shows the continuous interaction between clients and developers (Bannerman, Hossain, & Jeffery, 2012).

In a systematic literature review on the use of agile methodologies in software development identified that KM is one of the most popular topics among agile researchers. According to the authors, this is because agile methodologies promote a continuous learning environment (Dingsøyr et al., 2012).

Dorairaj, Noble and Malik (2012) emphasize that Scrum activities encompass four distinct KM processes: creation, coding, transfer, and application of knowledge. In addition, the authors state that agile teams must be multifunctional and have the skills necessary to perform software development activities.

Sungkur and Ramaswmy (2014) worried about the inefficiency and delays that can occur when exchanging knowledge among the members of the agile distributed team. They have proposed a tool namely Knowledg4Scrum, to mediate and storing all the information exchanged among the members of the team. They used data mining techniques to promote the creation of new knowledge. According to the authors, the proposed tool supports the processes of creation, storage, dissemination and application of knowledge.

Miklosik, Hvizdová, and Zák (2012) state that Scrum bases itself on the KM principles in order to manage projects efficiently. The authors compare the Scrum methodology and the KM principles. They conclude that Scrum helps implement KM practices in organizations. Takpuie and Tanner (2016) have proposed a theoretical model to identify the characteristics necessary for the transfer of tacit knowledge among Scrum team members.

Morampudi and Raj (2013) created an evaluation matrix among Scrum stages, containing the phases and subphases of the KM framework stages and the model of knowledge transfer framework. They suggested as the main contribution the identification of the weaknesses and the choice of the best practices that fit the methodology.

Singh, Singh, and Sharma (2014) conducted an empirical study on the relationship between KM practices and Scrum in software engineering organizations of India which utilize agile methodologies in their work. The authors came to the conclusion that most of the investigated firms use agile methodologies and that they help manage knowledge by providing the right technologies to support KM processes as a whole.

Eloranta and Koskimies (2012) proposed several models to align knowledge architecture management with Scrum. According to the authors, the way that KM is integrated into Scrum depends on the organization culture, that is, it depends on how organizations work while using Scrum.

3 METHOD

Knowledge is the fundamental raw material of the software industry. The intention of sharing and incorporating new knowledge, monthly, is what makes representatives of companies of this sector, from a certain region located in South of Brazil, meet. On April 17th, 2016 we were invited to attend one of these meetings. On that day 52 representatives from 11 different companies were present. The participants play different roles within the software development companies in which they work, such as director, business analyst and project manager. Throughout the meeting, we were given the time to ask questions to all the participants about the practices used to maintain knowledge in their organizations. One of the questions addressed the methodologies used for developing software, and the use of the Agile Scrum method was unanimous among the participants.

As the agile Scrum method follows a cycle, just like the KM, then we tried to understand how the two of them relate. For this, an exploratory study was carried and out shows, in general, how the knowledge of the individuals involved in the implementation and validation of software, behaves in the face of Scrum practices.

3.1 Knowledge Management Cycle Adopted

Due to the lack of a single KM cycle present in the literature, in this paper, we adopted the KM cycle from Tenório et al. (2017), Evans et al. (2014), and Goldoni and Oliveira (2010) as shown in Figure 2.

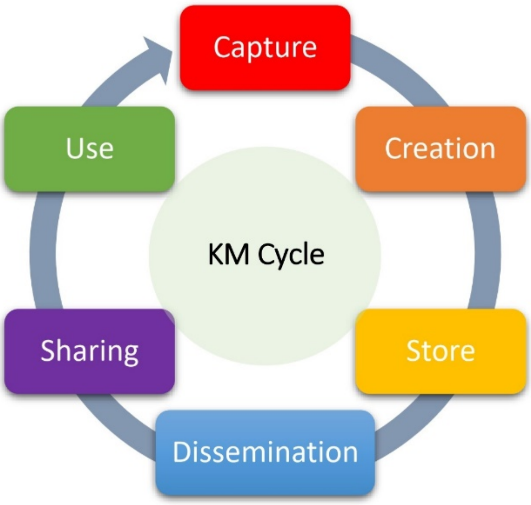

Figure 2. **The adopted KM cycle**

Source: The authors

Capture process corresponds to the identification of the explicit knowledge that was generated from the existence of tacit knowledge. Capturing consists of searching either within the organization itself or in an external environment, for information or knowledge which already exists and can be applied to a given situation (Evans et al., 2014). Tenório et al. (2017) suggest that in the capture phase there are implicit processes that lead to the production or creation of new knowledge. The capture phase takes place after there is a necessity to create new knowledge.

Creation process consists in the continuous updating of the organizational knowledge, promoting the interaction between individuals' tacit and explicit knowledge (Lyles & Schwenk., 2007). Evans et al. (2014) state that it is at this stage that new knowledge is created. Tenório et al. (2017) argue that the creation process is directly related to the capture process. Moreover, the creation process may also happen based on information stored by the organization.

Store process consists of organizing the knowledge by using the information that can be accessed and transferred, making it available to everyone in a structured way (Milton, Shadbolt, Cottam, & Hammersley, 1999). This stage is after the capture and creation processes, thus making acquired or created knowledge part of the intangible assets of the organization. The storage structure should provide a quick and categorized query, in a way that subjects are not randomly stored, but rather by type or nature of problem (Evans et al., 2014).

Dissemination process is characterized by the moment in which codified knowledge has the potential to be shared (Evans et al., 2014). This helps ensure that important information is not lost because it has already been evaluated and can be disseminated more quickly and stored later. Krogh and Roos (1995) state that the dissemination of knowledge is related to individual competencies. In this sense, it is believed that the dissemination phase differs from the sharing phase because it is an existing possibility on a certain subject and will only be stored after they are validated in some way. Tenório et al. (2017) also point out that the dissemination process evaluates the possibility of incorporating knowledge into the organizational environment.

Sharing process comprises the sharing of existing knowledge with individuals, groups, teams, departments, and organizations. Thus, this phase can be carried out immediately after the storage phase or depending on the needs (Ipe, 2003; Evans et al., 2014). The knowledge-sharing stage poses a challenge to organizations due to the unstructured nature of tacit knowledge and the numerous barriers that prevent knowledge from flowing in the organizational environment (Asrar-Ul-Haq & Anwar, 2016). In this sense, Evans et al. (2014) suggest that a combination of technologies and tools support the right time to make this knowledge explicit. For Tenório et al. (2017) this process has an intimate relationship with the existing knowledge flow within the organization.

Finally, the Use process is linked directly to the effective application of knowledge in the organization. As explicit as knowledge may be available, at this stage, in order to use it, tacit knowledge will be applied by the individuals. There are many ways in which knowledge can be used, such as the creation of communities of practice (Evans et al., 2014). Tenório et al. (2017) found out that the use process is according to how knowledge will be applied.

4 RESULTS

The Scrum framework consists of a set of activities such as Product Backlog, Sprint Backlog, Increment, Sprint Planning, Development work, Daily Scrum, Sprint Review, Sprint Retrospective, and Sprint Burndown. All these elements are part of the KM cycle since they support all the knowledge of the organization. The results are presented according to the activities of the Scrum method. Based on the description presented in the previous section, an integration between KM cycle and Scrum activities is shown in Table 2.

	KM Cycle					
Scrum activities	*Capture*	*Creation*	*Store*	*Dissemination*	*Sharing*	*Use*
Product Backlog	●		●	●	●	●
Sprint Backlog			●	●		
Increment					●	●
Sprint Planning				●	●	
Development Work	●	●				●
Daily Scrum	●	●		●	●	●
Sprint Review				●	●	
Sprint Retrospective	●	●		●	●	
Sprint Burndown				●	●	

Table 2. **Integration between Scrum activities and KM processes**

Source: The authors

For the creation of the Product Backlog artifact, a discussion of the project to be developed is required. At this stage, the Product Owner must capture the information in the client's environment (using printed documents, previously used forms, monitoring routine activities in the company, etc.), get them stored and disseminate them to the team. When discussing requirements, team members share their experiences through information exchange and determine which items are relevant to product development. In addition, it is at this stage that team members determine which paths to follow. Previous experiences help trace those paths. This way, the sharing and use of pre-existing knowledge among individuals occur.

The storage and dissemination of knowledge take place in the next phase, that is, the Sprint Backlog, since that artifact is responsible for defining the list of tasks that must be executed by the Scrum Team within a Sprint.

The increment artifact is responsible for the sharing of the results obtained at the end of a Sprint, and the new features that will be implemented into the client's environment. Therefore, this artifact shows the new functionalities that have been captured and used.

In the Sprint Planning event, the Product Owner will choose which Product Backlog functionalities will be developed in the current Sprint; in this case, knowledge dissemination takes place among team members. Thus, the sharing of knowledge takes place within the Scrum Team, during the Sprint Backlog.

The development work phase is intended to implement the features set out in the Sprint Backlog. The capture, creation, and use of knowledge by the different members of the Scrum Team take place at this stage because during this phase a new product will be created and tested.

Daily Scrum meetings allow all team members to know what activities are being done by each one of members. By making this report, team members disseminate knowledge within the whole team, providing the listener with the information that may help them in their work, so that knowledge can be captured, created, shared and used.

At the Sprint Review meeting, a comparison between the delivered increment and the goals established in the Sprint Planning, involving the dissemination and sharing of knowledge, is performed. The Sprint Retrospective phase promotes the dissemination of the best practices and the sharing and creation of new knowledge, that is, the capture of the information that can be used to develop new projects and create new knowledge. Finally, the Sprint Burndown, through a graphic record, disseminates and shares what has already been done and what remains to be done.

5 DISCUSSION

Following the main question of this article which investigates how Scrum practices are related to the KM cycle, we observed in Table 2 while Scrum is carried out, KM cycles are directly related to Scrum activities. Thus, Figure 3 presented the integration between the KM cycle and Scrum activities.

Software development teams carry out knowledge-intensive activities. Therefore, they need highly valuable knowledge (Dorairaj et al., 2012). Dingsøyr et al. (2012) state that software development is a knowledge-building activity and therefore knowledge management should be an attractive prospect when exploring the creation of knowledge within software development teams, in general, and within agile teams. Effective knowledge management can be achieved by the implementation of diverse knowledge management processes, e.g., knowledge creation, storage, dissemination, and application (Sungkur & Ramasawmy, 2014).

The use of agile methodologies in software development influences team members' communication (Dönmez et al., 2016). Generally, agile methods allow for close interaction among team members by enabling efficient and effective sharing of information among team members (Bannerman et al., 2012). Looking at Figure 3, we note that Scrum provides this interaction among team members through their activities. Also, the knowledge required for the development of agile software is very context-dependent, and it is often difficult to transfer context to different projects, even within the same organization (Andrej et al., 2012). According to Sungkur and Ramasawmy (2014), Scrum activities provide conditions for the KM cycle to take place within software development teams, as they seek to incorporate pre-existing knowledge into the running activities, helping in the decision-making process. When we relate the activities of Scrum to the KM cycle, we observe a constantly learning environment, since KM promotes that. In this sense, it is expected that the software development team is open to change (Dingsøyr et al., 2012).

According to Dorairaj et al. (2012), agile teams are multifunctional and should promote the sharing of specific knowledge through frequent interactions and collaboration with the client. This can also be seen in Figure 3. While Scrum activities are being executed, the KM cycle is also in constant execution. However, feedback cycles and small iterations only occur when the individuals involved have some knowledge to share or to put into execution (Andrej et al., 2012). Therefore, Scrum can be considered as an important KM tool, since the activities performed enable the sharing and incorporation of knowledge to the whole process.

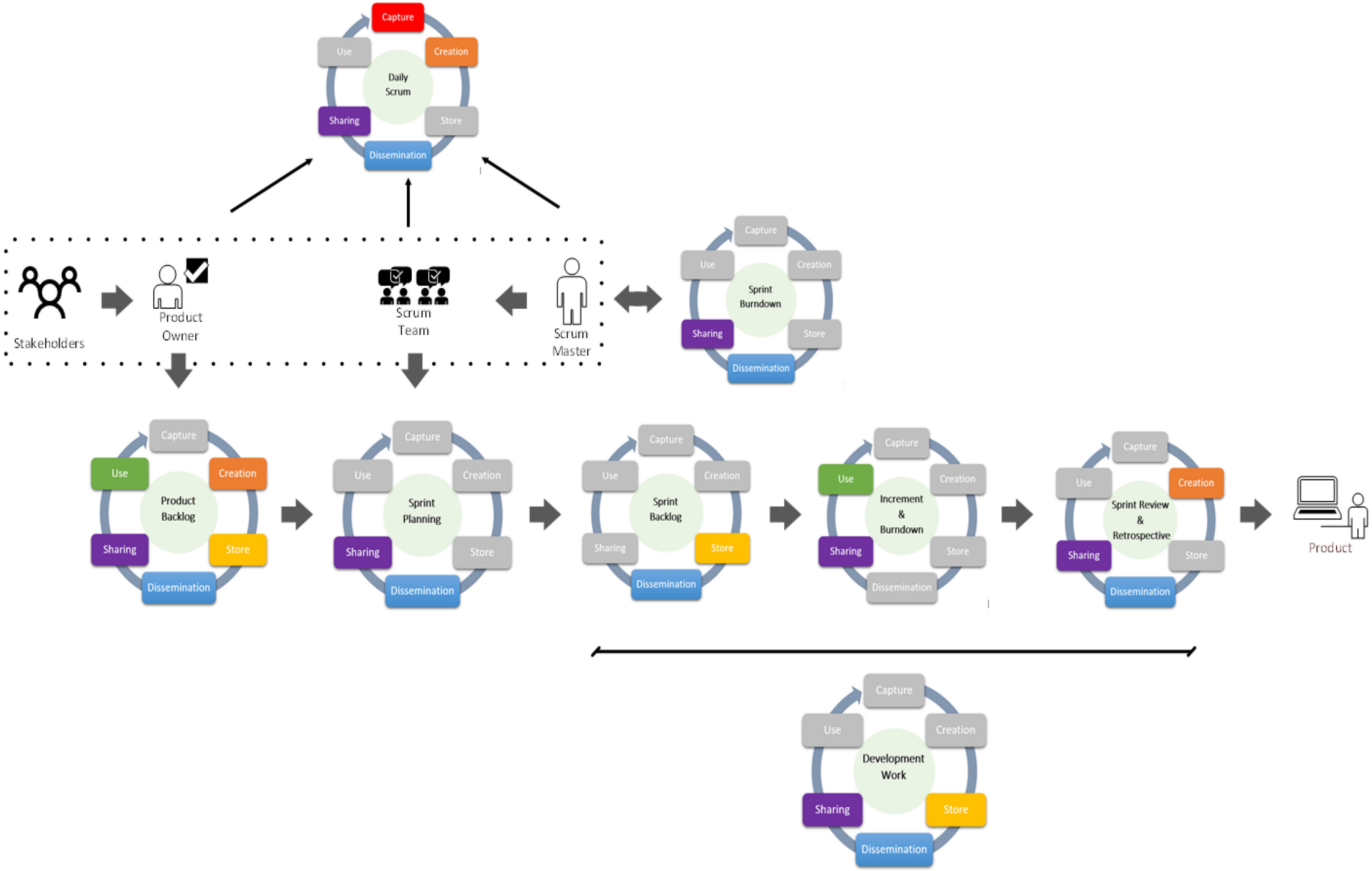

Figure 3. **KM Cycle related to Scrum activities**
Source: The authors

6 CONCLUSION

This article suggested that Scrum activities can support KM cycle in order to convert tacit knowledge to explicit knowledge within software development teams. In this sense, we have shown how Scrum activities are related to the processes of a KM cycle which support knowledge capture, creation, storage, dissemination, sharing, and use. We observed that Scrum activities could be directly associated with the KM cycles once its practices allow to incorporate tacit knowledge into the organization, thereby increasing knowledge transfer among individuals. Our results are means to stimulate new research in software industry addressed to systematize the knowledge of team members to creation of new ideas and, consequently, increase the innovation of software products. However, we limited our exploratory research to the literature, and we did not identify whether the KM cycle really completes itself at the software development teams accomplishing Scrum practices in a real environment. Thus, the challenge is to carry on in-depth research to assess our empirical results in a practical context. Finally, we encourage Scrum researchers and practitioners to investigate Scrum activities supported by different KM practices and tools in order to clarify the comprising about the benefits of KM processes for the software industry.

REFERENCES

Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. Z. (2010). Agile Software Development: Impact on Productivity and Quality. In *Proceedings of the 2010 IEEE ICMIT*, Singapore.

Andrej, M., Hvizdová, E., & Zák, S. (2012). Knowledge management as a significant determinant of the competitive advantage sustainability. *Journal of Economics*. 60(10), 1041-1058.

Asrar-Ul-Haq, M. & Anwar, S. (2016). A systematic review of knowledge management and knowledge sharing: trends, issues, and challenges. *Cogent Business & Management*, 3.

Aurum, A., Daneshgar, F., Ward, J. (2008). Investigating Knowledge Management practices in software development organisations - an Australian experience. *Information and Software Technology*, 50(6), 511–533.

Bannerman, P. L., Hossain, E., & Jeffery, R. (2012). Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? In *Proceedings of Hawaii International Conference on System Sciences*.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M.,… Thomas, D. (2001). *Manifesto for Agile Software Development*. Retrieved from http://www.agilemanifesto.org/

Binuyo, G. O., Oyebisi, T. O., Olayinka, A., Afolabi, B. S. (2015). Evaluation of the Factors influencing the Indigenous Software Products Development in Nigeria. *International Journal on Advances in ICT for Emerging Regions (ICTer)*, 7(3), 1–8.

Bjornson, F. O. & Dingsoyr, T. (2008). Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, 50(11), 1055–1068.

Brito, M. F., Figueiredo, R. C., Venson, E., Canedo, E. D., & Ribeiro Junior, L. C. M. (2017). Knowledge Transfer in a Management Process for Outsourced Agile Software Development. In *Proceedings of 50th Hawaii International Conference on System Sciences*.

Cámara, J., Correia, P., Lemos, R., Garlan, D., Gomes, P., Schmerl, B., & Ventura, R. (2016). Incorporating Architecture-Based Self-Adaptation into an Adaptive Industrial Software System. *Journal of Systems and Software*, 122, 507-523.

CEN. (2004). European Guide to good Practice in Knowledge Management. *Management Science*, 50(3), 352–364.

Cervone, H. F. (2011). Understanding agile Project management methods using Scrum. *International Digital Library Perspectives*, 27(1).

Chen, H., Ragsdell, G., & O'Brien, A. (2014). Identification of Tacit Knowledge Associated with Experience: a Chinese Software Industry Study. In *Proceedings of 15th European Conference on Knowledge Management*. Portugal.

Cristal, M., Wildt, D., & Prikladnicki, R. (2008). Usage of SCRUM Practices within a Global Company. In *Proceedings of International Conference on Global Software Engineering*.

Dalkir, K. (2011). *Knowledge Management in Theory and Practice*. London: The MIT Press.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: towards explaining agile software development. *Journal System Software*, 85(6), 1213–1221.

Dönmez, D., Grote, G., & Brusoni, S. (2016). Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. *Information and Organization*, 26, 63-83.

Dorairaj, S., Noble, J., & Malik, P. (2012). Knowledge management in distributed agile software development. In *Agile Conference (AGILE)*.

Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: *A systematic review. Information and Software Technology*, 50, 833–859.

Eloranta, V. P. & Koskimies, K. (2012). Aligning architecture knowledge management with Scrum. In *Proceedings of the WICSA/ECSA 2012 Companion. Helsinki.*

Eloranta, V. P., Koskimies, K., & Mikkonen, T. (2016). Exploring ScrumBut – an empirical study of Scrum anti-patterns. *Information and Software*, 74, 194-203.

Evans, M., Dalkir, K., & Bidian, C. A. (2014). Holistic View of the Knowledge Life Cycle: The Knowledge Management Cycle (KMC) Model. *Electronic Journal of Knowledge Management*, 12(2), 85-97.

Goldoni, V. & Oliveira, M. (2010). Knowledge management metrics in software development companies in Brazil. *Journal of Knowledge Management*, 14(2), 301-313.

Highsmith, J. & Consortium, C. (2002). What is agile software development? *The Journal of Defense Software Engineering*, 15, 4-9.

Ipe, M. (2003). Knowledge Sharing in Organizations: a conceptual framework. *Human Resource Development Review*, 2(4), 337-359.

Krogh, G. V. & Roos, J. (1995). A perspective on knowledge, competence and strategy. *Personnel Review*, 24(3), 56-76.

Lei, H., Ganjeizadeh, F., Jayachandran, P. K., Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67.

Lyles, M. & Schwenk, C. R. (2007). Top management, strategy and organizational knowledge structures. *Journal of Management Studies*, 29(2), 155–174, https://doi.org/10.1111/j.1467-6486.1992.tb00658.x

Marçal, A., Freitas, B. C. C., Soares, F. S. F., & Belchior, A. D. (2007). Mapping CMMI project management process areas to SCRUM practices. In *Proceedings of the Software Engineering Workshop*.

Meihami, B. & Meihami, H. (2014). Knowledge management a way to gain a competitive advantage in firms (evidence of manufacturing companies). *International Letters of Social and Humanistic Sciences*, 14, 80-91.

Melo, C.O., Cruzes, D. S., Kon, F., & Conradi, R. (2013). Interpretative case studies on agile team productivity and management. *Information and Software Technology*, 55(2), 412-427.

Milton, N., Shadbolt, N., Cottam, H., & Hammersley, M. (1999). Towards a knowledge technology for knowledge management. *International Journal Human-Computer Studies*, 51, 615-641.

Moe, N. B., Dingsøyr, T., & Dybá, T. A. (2010). A teamwork model for understanding an agile team: a case study of a Scrum Project. *Information and Software Technology*, 52, 480-491.

Morampudi, N. S. & Raj, G. (2013). Evaluating Strengths and Weaknesses of Agile Scrum Framework using Knowledge Management. *International Journal of Computer Applications*, 65 (23).

Nawinna, D. P. (2011). A model of knowledge management: delivering competitive advantage to small & medium scale software et al industry in Sri Lanka. In *Proceedings of 6th International Conference on Industrial and Information Systems*, 2011.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48(5), 73-78.

Ortegón, A. M., Lasso, A. L., & Steil, A. V. (2016). Organizational Strategy and the Knowledge Management Cycle: Bukowitz and William´s Model in Practice. *Espacios*, 37(7), 11.

Pinoa, F.J., Pedreira, O., García, F., Luaces, M. R., Piattini, M. (2010). Using Scrum to guide the execution of software process improvement in small organizations. The Journal of Systems and Software, 83, 1662–1677.

Rola, P., Kuchta, D., & Kopczyk, D. (2016). Conceptual model of working space for Agile (Scrum) project team. *Journal of Systems and Software*, 118, 49-63.

Schwaber, K. & Sutherland, J. (2016). *The Scrum Guide^{TM}* - The definitive guide to Scrum: the rules of the game. Retrieved from http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100

Schwaber, K. (2004). *Agile Project Management with Scrum*. EUA: Microsoft Press.

Singh, A., Singh, K., & Sharma, N. (2014). Agile knowledge management: a survey of Indian perceptions. *Innovations in Systems and Software Engineering*, 10, 297.

Sungkur, R. & Ramasawmy, M. (2014). Knowledge4Scrum, a novel knowledge management tool for agile distributed teams, *VINE*, 44(3), 394-419.

Takpuie, D. & Tanner, M. (2016). Investigating the Characteristics Needed by Scrum Team Members to Successfully Transfer Tacit Knowledge During Agile Software Projects. *The Electronic Journal Information Systems Evaluation*, 19(1), 36-54.

Tenório, N., Pinto, D., Vidotti, A. F., Santos, M. O., Urbano, G. C., & Bortolozzi, F. (2017). Tool Based on Knowledge Management Process: An Interview Protocol to Gather Functional Requirements from Software Industry Experts. *Matter: International Journal of Science and Technology*, 3(1), 45-54.

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: applying SCRUM principles to software product management. *Information and Software Technology*, 53.

Wiig, K. M. (1997). Knowledge management: an introduction and perspective. *Journal of Knowledge Management*, 6–14.

Zhang, X. & Dorn, B. (2011). Agile Practices in a Small-Scale, Time-Intensive web development Project. In *Eight International Conference on Information Technology*, 2011.