

Knowledge-creation in student software-development teams

Author:

Mzwandile M. Shongwe¹

Affiliation:

¹Department of Information Studies, University of Zululand, South Africa

Correspondence to:

Mzwandile Shongwe

Email:

shongwem@unizulu.ac.za

Postal address:

Private Bag X1001, Kwadlangezwa 3886, South Africa

Dates:

Received: 20 Feb. 2014

Accepted: 13 Oct. 2014

Published: 06 Feb. 2015

How to cite this article:

Shongwe, M.M., 2015, 'Knowledge-creation in student software-development teams', *South African Journal of Information Management* 17(1), Art. #613, 8 pages. <http://dx.doi.org/10.4102/sajim.v17i1.613>

Note:

This is a revised version of the paper that was presented at the 4th International Conference on Information Systems Management and Evaluation (ICIME), Ho Chi Minh City, Vietnam, 13–14 May 2013.

Copyright:

© 2015. The Authors. Licensee: AOSIS OpenJournals. This work is licensed under the Creative Commons Attribution License.

Read online:

Scan this QR code with your smart phone or mobile device to read online.

Background: Knowledge-creation is a field of study that has gained popularity in recent times. Knowledge-creation is the creation of new ideas or new innovations. In computing, software-development is regarded as knowledge-creation. This is because software-development involves the creation of a new innovation (software). Knowledge-creation studies in this field tend to focus mainly on knowledge-creation activities in business organisations. They use experienced, professional software-development teams as subjects, largely ignoring novice student development teams. This has denied the field of computing valuable knowledge about how novice teams create knowledge.

Objectives: The study addressed this gap in the literature by investigating knowledge-creation in student software teams.

Method: An ethnographic study was conducted on six student teams developing software in a management-information systems (MIS) course. They were conducting a systems-development project at a university during a term of study. Data were collected over a period of four months through participant observation and interviews.

Results: The results reveal knowledge-creation activities such as problem definition, brainstorming, programming and system documentation. Students use the Internet, books, class notes, class presentations, senior students and professional software developers as sources of information. Mobile phones and BlackBerry devices facilitate knowledge-creation. Challenges to knowledge-creation are the lack of material and financial resources, a lack of technical skills, a lack of time, students staying off-campus and ambivalent team members.

Conclusion: The conclusion drawn from this study is that student teams are capable of creating knowledge (a working system) just like professional teams, but the knowledge-creation process is slightly different.

Introduction

Knowledge-creation is a field of study that has gained popularity in recent times (Mitchell & Boyle 2010). Knowledge-creation is the creation of new ideas or new innovations, for example, a new product, service or process. In computing, software-development is regarded as knowledge-creation (Bailin 1997). Software-development is a process of expressing someone's intentions of how a machine should work (Bailin 1997). Bailin further states that knowledge is generated by building, operating and maintaining the system during the software-development process. This knowledge includes lessons learned during the development of the system, experience gained over time when using the system and broader knowledge acquired over time by the development and customer organisations. Dorairaj, Noble and Malik (2012:1) give a knowledge-management perspective of software-development as '... a series of knowledge-intensive activities that encompasses gathering requirements, analysing problems, designing, coding, testing, and ensuring the software remains up-to-date and bug-free'. Software-development is regarded as knowledge-creation because it is an innovative process which leads to the generation of new ideas and the creation of new software products.

Studies have been conducted in the field of computing to investigate the phenomenon of knowledge-creation. For example, Linden and Cybulski (2009), Wan *et al.* (2010) and Arent and Norbjerg (2000) to name but a few investigated knowledge-creation processes in pattern mining, requirement-elicitation process (REP) and software-process improvement (SPI) respectively. The conclusion of these studies was that all these activities are knowledge-creation activities. The problem with these studies is that they tend to focus mainly on knowledge-creation in business organisations. They also tend to use experienced, professional software-development teams as

subjects, largely ignoring public institutions such as higher education institutions and thus ignoring the development teams amongst novice students. This has denied the field of computing valuable knowledge about how novice teams create knowledge. In fact, these novices are future developers, and yet, they are largely ignored by researchers. For example, we do not know what the knowledge-creation activities in software-development teams amongst students entail. What channels of communication facilitate knowledge-creation in student teams? What information sources do students use to create knowledge? And what challenges do student teams face when they create knowledge? A lot has been written about professional, experienced teams. For example, we know what the knowledge-creation activities in professional teams entail (Poh & Erwee 2004). We also know that professional teams use technology and networks of peers (communities of practice) to facilitate knowledge-creation and transfer (Alavi & Leidner 2001; Kimmerle, Cress & Held 2010; Wasko & Faraj 2000; Wenger 1998). We are also aware that rising costs, limited time to complete projects and abandoned projects are some of the challenges concerning knowledge-creation faced by professional development teams (Heeks 2002; Lindvall, Rus & Sinha 2002; Lyytinen & Robey 1999). However, we do not know what is happening in student teams. The study set out to answer these questions in the context of software-development as knowledge-creation.

This article is structured as follows: The following section presents the theoretical foundations on knowledge-creation. That is followed by the methodology and then the results and discussions. Lastly, I present conclusions and recommendations for further research.

Theoretical foundations

This section will briefly highlight the theoretical foundations of knowledge-creation. Firstly, I shall discuss the distinction between data, information and knowledge, and then I shall discuss knowledge-creation.

Data, information and knowledge

The concepts of knowledge, data and information may seem similar, but their meaning may not be uniformly understood. Data is unprocessed information (Hey 2004). It could be bits and numbers with no meaning (Rumizen 2002). Information is data that has been processed for a useful purpose and can be used for decision-making. Chaffey and Wood (2005) describe information as data with value to the understanding of a subject and in a context. Knowledge is '... information possessed in the mind of individuals: it is personal information related to facts, procedures, concepts, interpretations, ideas, observation and judgment' (Alavi & Leidner 2001:109). Knowledge could take several perspectives or forms: a state of mind, an object, a process, a condition of having access to information or a capability (Alavi & Leidner 2001). In software-development, knowledge could be seen from the perspective of either the process or the object. The object is the output or end-product (the software). The process

perspective entails the software-development process, for example, generating new ideas, refining them and applying them in the development process. Knowledge could be tacit or explicit (Nonaka & Takeuchi 1995; Polanyi 1962). Tacit knowledge is what the knower knows, knowledge that is derived from experience and that embodies beliefs and values (Nonaka & Takeuchi 1995). It is personal and difficult to formalise, making it difficult to communicate and share with others (Elfving & Funk 2006; Gladstone 2000; Li & Gao 2003). Explicit knowledge is knowledge that has been articulated and, more often than not, captured in the form of text, tables, diagrams and product specifications (Nickols 2000). In software-development, tacit knowledge resides in the developer's mind. It is the hidden knowledge in the form of the experiences of the developer. Explicit knowledge is the end product (the system) and the documentation.

Knowledge-creation

Knowledge-creation is a series of processes and activities that add value to produce an outcome or output such as a product, service or process (Mitchell & Boyle 2010). Holsapple and Singh (2001:84) define knowledge-creation as '... an activity that produces knowledge by discovering it or deriving it from existing knowledge'. It is the coming into existence of new knowledge (Phan & Peridis 2000). It involves the definition of a problem and makes use of complex and discontinuous events to solve the problem (Parent *et al.* 2000; Styhre, Roth & Ingelgard 2002). Knowledge-creation involves the monitoring, production and transfer of knowledge resources within and outside of the organisation (Holsapple & Singh 2001). When defined as a process, knowledge-creation refers to the generation of new ideas that reflect a significant enrichment of existing knowledge. It is defined as an outcome or output when activities and initiatives are undertaken to generate new ideas or objects. It means that new knowledge is diffused, adopted and embedded as new products, services and systems (Mitchell & Boyle 2010). Knowledge-creation takes place at individual, group, organisational and inter-organisational levels (Nonaka & Takeuchi 1995). It is widely agreed that knowledge-creation takes place first at individual level and then amplified at group, organisational and inter-organisational levels (Nonaka & Takeuchi 1995). In fact, it is believed that organisations only offer a space or context in which the knowledge created by individuals is amplified and applied in organisational routines and processes. The organisation could be what Nonaka, Toyama and Konno (1998) refer to as 'ba' – a shared context where knowledge is created.

Knowledge-creation activities

According to Nonaka and Takeuchi (1995), knowledge is created in five steps: sharing tacit knowledge, creating concepts, justifying concepts, building a prototype and cross-levelling knowledge. At the sharing stage of tacit knowledge, individuals share emotions, feelings and mental models through face-to-face interactions. Creating concepts involves the sharing of both tacit and explicit knowledge. After sharing

tacit knowledge, organisational members then articulate the tacit knowledge into explicit knowledge through written concepts. Justifying concepts is a phase whereby created concepts are determined whether they are useful or not in the organisation. The justified concepts are then converted into a tangible or concrete product or model called an archetype. After knowledge has been created, it is levelled across the organisation. This could be inter-organisational or intra-organisational (Nonaka 1994; Nonaka & Takeuchi 1995). This notion is also shared by Wan *et al.* (2010). Examples of knowledge-creation activities in software-development are the following: constructing a software routine, brainstorming, discovering a pattern, solving a problem or achieving a creative insight (Holsapple & Singh 2001).

Knowledge-creation theories

A number of theories are used to study knowledge-creation in organisations. Widely used theories include, amongst others, the organisational learning theory (Argyris & Schon 1978), the learning-organisation (Senge *et al.* 1994), the theory of organisational knowledge-creation or SECI (Nonaka & Takeuchi 1995), the knowledge-integration theory (Grant 1996) and the communities of practice theory (Wenger 1998). Argyris and Schon's organisational learning theory explains how knowledge is created by the learning of individuals and groups in organisations which facilitate learning at organisational level. Senge's learning-organisation theory defines the principles of a learning organisation. These principles are personal mastery, systems thinking, team learning, mental models and building a shared vision. These principles are believed to facilitate learning in departments and sections of the organisation, which leads to the whole organisation learning and becoming a learning organisation. Nonaka and Takeuchi's theory of organisational knowledge-creation explains how individuals, teams and entire organisations create knowledge by the conversion of tacit and explicit knowledge through four processes: socialisation, externalisation, combination and internalisation. Grant (1996) focuses on knowledge-integration and innovation within a firm. Grant explores the coordinating mechanisms of how individuals in a firm integrate knowledge to foster innovation. Wenger's theory explains how small, informal teams in organisations share and create knowledge from each other through social interactions. Lee and Cole's (2003) community-based model of knowledge-creation explain how knowledge is created by a software community of practice operating outside of formal organisational boundaries.

These theories focus mainly on how knowledge is created in organisations (except Lee and Cole's [2003] model). They explain that knowledge is created, firstly, at individual level and at group level before it becomes organisational knowledge. They also present different scenarios under which knowledge is created. The main limitation of these theories and knowledge-creation literature in general is that the focus is mostly on knowledge-creation by professionals in business organisations. Little has been done to investigate amateur, student teams despite the fact that they are the

future innovators. The current study tries to fill this gap in the literature by focusing on students' knowledge-creation activities.

Knowledge-creation in software-development

Software-development is regarded as knowledge-creation (Bailin 1997). It is a knowledge-intensive task, which involves a process of discovery and invention, accumulation, analysis, cognition and integration (Bailin 1997; Dingsoyr 2002; Mitchell & Nicholas 2006). This means that software-development processes (systems analysis, coding, documentation, testing, etc.) are knowledge-creation processes. During these activities, individuals and teams acquire knowledge from different sources; apply their previously acquired knowledge; store acquired knowledge in databases, routines, and procedures and share their knowledge with others to produce a knowledge output (the software product).

A number of studies have been conducted to investigate the knowledge-creation phenomenon in computing. A few examples can suffice. Arent and Norbjerg (2000) investigated software-process improvement (SPI) and concluded that they are knowledge-creation activities. Kess and Haapasalo (2002) investigated knowledge-creation through a software-project review process, and they constructed a tool to improve the software-development process. Klint and Verhoef (2002) investigated how the principles of knowledge-management could be applied in knowledge-creation. Linden and Cybulski (2009) investigated knowledge-creation in pattern mining. This led to the enrichment of the knowledge-creation framework proposed by Wickramasinghe and Lichtenstein. Morner and Von Krogh (2009) explored the conditions under which successful knowledge-creation takes place. They proposed three conditions (perceptibility, systemic memory and modularity) under which the knowledge-creation process could be stabilised. Wan *et al.* (2010) investigated knowledge-creation in requirement-elicitation process (REP) and concluded that it is a knowledge-creation process. Neves *et al.* (2011) conducted a systematic review of knowledge-creation in agile software-development teams and found two knowledge-creation activities, comprehensive software documentation and responding to change as knowledge-creation activities. Spohrer *et al.* (2013) also investigated knowledge-creation activities in peer programming teams working on information-systems development in a large software organisation. They found that knowledge is created by exchanging opinions, solutions, and code, and developers who are observing their peers thus learn from their actions. They found that knowledge is created at individual and team level. If we look at the literature, we can conclude that student teams in software-development are rarely studied. In the list of studies mentioned above, none have used students as subjects, hence the importance of this study.

Methodology

An interpretive, qualitative study was conducted on student software-development teams. Interpretive studies assume that people create their own subjective meanings as they interact

with the world around them (Lee 1991; Olikowski & Baroudi 1991). Ethnography was chosen as a research method. Myers (1999) defines ethnography as a study where the researchers immerse themselves in the lives of the people being studied. Ethnographers aim to place the phenomena being studied in the social and cultural context of the people studied. Myers (1999) further states that the ethnographer has to spend a considerable amount of time in the field and to collect data not only through interviews and documents but also through participant observation. In this study, a class of six software-development project teams (24 students) was studied. The groups were supposed to develop an e-commerce interactive website for a client. The researcher joined the class at the beginning of the semester before the project started. He was briefed about the project just like the students and given all necessary documentation. He attended all lectures, presentations and meetings that the teams had. These activities took place in the students' computer laboratory, lecture halls and the library. The aim was to be part of the teams as if the researcher was one of them. Lectures were held once a week for the whole semester. Presentations were held based on need such as when there was a milestone deliverable (e.g. project plan). The teams also had meetings almost daily to discuss and develop the system. During these activities (lectures, presentations and team meetings), the researcher observed and interviewed teams. During observation, detailed notes were taken on how the students developed the system. The researcher observed how, where and what ideas team members shared and the impact of these ideas on the completion of the project. Group interviews were conducted to supplement observation data. They were used to seek further clarification on certain development processes such as the information sources and communication channels used by teams in their projects. The study was conducted for the whole semester, which was four months long. Qualitative content analysis was then used to analyse the responses. Qualitative content analysis is a method for the subjective interpretation of text data through coding and themes or category identification (Hsieh & Shannon 2005). In this study, data were coded and classified into categories. Ethnography has been used before by Suchman (1995) and Myers and Young (1997), amongst others, to study information-systems development.

Results and discussions

This section responds to the research questions posed in the study.

Knowledge-creation activities

This section answers the following question: What knowledge-creation activities do student teams undertake? The results reveal a number of activities which are undertaken at individual and team level.

Individual knowledge-creation activities

The results indicate that individual team members performed a number of activities during the knowledge-creation process. Major individual activities were coding,

documentation, preparing slides for presentations, designing the user interface, research and information gathering, sourcing finance and other relevant activities assigned to the individual. One respondent stated: 'I was mainly the website designer, but I also did documentation such as project plan, feasibility study and other activities'. Another reported that '... as an individual, I created forms in the prototype'. Others reported on other minor activities that they did such as writing minutes and arranging future meetings. The importance of individual knowledge activities was emphasised by all group members. One member stated:

'The contribution of each member was great. For every task we had to break it down to individual activities. In so doing we shared ideas which enabled us to finish the project quicker'.

These results are consistent with the knowledge-creation literature which highlights the importance of the individual in knowledge-creation, for example Grant (1996). Simon (as cited in Grant 1996:112) shares the same sentiments. Nonaka and Takeuchi (1995) present an excellent account of how individuals create knowledge which is then infused into the whole organisation. In this particular case, individuals used their creative abilities to build the different parts of the system. They used their own existing and newly acquired knowledge to carry out the tasks at hand. These individual activities contributed immensely to the completion of the project. The completed individual tasks would then be taken to the next level of discussion and further refinement. That was at the team level. The team's knowledge-creation activities are discussed in the following section.

Knowledge-creation activities of the team

The tasks given to individuals were then consolidated and adopted at group level. Each individual would bring their tasks to the team for further discussions. The team would reject or adopt the task as team knowledge. Tasks such as coding, preparing documentation and presentations and others were discussed at team level. In fact, all individual tasks were further discussed by the teams before they were accepted to be project knowledge. One such important task that was discussed at team level was problem definition, that is, what the project entailed. At first, students had no idea about what to do. One student lamented: 'I had no idea what I was supposed to do'. These sentiments were echoed by other students who reported that, '... we did not have a clue on what to do'. The teams then converged, discussed the problem and sought assistance from the lecturer and senior students. That was when they understood what they were supposed to do. They reported: 'It was after we met as groups that we shared ideas and came to understand what we had to do'. Morne and von Krogh (2009) also state that, in knowledge-creation, the first step usually lies in a single software problem with which a programmer is confronted. Nonaka and Takeuchi (1995) refer to such a state as intentional creative chaos. According to Nonaka and Takeuchi (1995), 'intentional creative chaos' is a situation whereby management creates a sense of crisis in the organisation by proposing challenging goals. Organisational members respond by defining the

problem and finding solutions to the problem. In student teams, the lecturer evokes a crisis by giving students a huge real-life problem, which they have never come across before and which they have to deliver to a real customer within a short space of time. It is after they have met as a team that they properly defined the problem, understood it and planned how to solve it.

Knowledge-creation in teams is also emphasised by a number of knowledge-creation theories. These theories (Hedlund 1994; Nonaka & Takeuchi 1995; Senge *et al.* 1994; Wenger 1998) agree that teams play a crucial part in knowledge-creation. It is at team level where rough ideas are refined and developed into knowledge.

Information sources for knowledge-creation

A number of information sources were used by student teams during the knowledge-creation process. Information sources such as class presentations, the lecturer and lecture notes, text books, the Internet and external sources (senior students and professional software developers) were used.

Class presentations were a good source of information. After a major milestone, for example system analysis, budgeting, prototyping and documentation, the group would present the deliverable to the whole class and lecturer. Other teams and the lecturer would then interrogate the presenting group by asking questions and commenting about the deliverable. Presenters were criticised, praised and advised on how to do the project better. One student stated that '... presentations helped us to know what we did right, where to correct and how to improve the project'. Another one concurred that '... presentations helped us to get clarity as to what is really required from us'. All the groups agreed that the presentations helped them to gain new knowledge about the project in one way or the other. Teams stated that they gained knowledge from the presentations from other teams. They learned communication and presentation skills and to work under pressure. It boosted their morale when they realised that they were making progress. On an individual note, team members indicated that class presentations helped them to gain knowledge from others.

The lecturer also played a crucial role in the knowledge-creation process. Apart from giving them the problem to solve, the lecturer was also involved in helping the teams solve the problem. Apart from teaching, the lecturer played the role of advisor and mentor. At times, classes would be conducted to teach students how to carry out certain activities of the project. Before the start of each major deliverable (e.g. systems analysis), the lecturer would conduct a lecture on the next activity. This helped students understand what was required of them and how to tackle the problem at hand. The lecturer also played a crucial role during the presentations. He would criticise the students' work, offer advice on how the work should be done and give credit to teams that did well. Students would also consult the lecturer for advice whenever they had problems with their projects. This was

confirmed by a respondent who stated that '... the lecturer gave us motivation to finish the project'. Another one concurred that '... the lecturer helped by explaining what [was] expected'. Another one stated that '... the lecturer made us to attend classes, and provided us with technical skills'. Not all students were happy with the lecturer's role though. Some indicated that the lecturer did not play any role in their knowledge-creation process. They stated: 'The lecturer did not play any role in the successful completion of the project. We struggled while he was present'. Others stated that '... the lecturer had little impact on the project'.

Students also regularly used the Internet for information. The teams indicated that they used the Internet to research how to do the activities of the project, for example, how to budget and how to do systems analysis. They also looked for technical information such as coding. They could surf the Internet to look at how to solve a specific technical problem. One team indicated that they used the Internet to find help on how to set-up the Apache, MySQL, PHP for Windows (WAMP) interface. The w3schools website is said to have provided much support to the teams. They also relied on books, senior students and professionals to complete their projects. Because this was an MIS group, most teams relied on senior students in computer science for technical support such as linking the database to the website and other coding activities. Surprisingly, they also sought help from other competing groups. One group stated that '... we sought help from other groups to do some of the activities'.

This leads to the conclusion that student teams seem to use readily available sources of information for knowledge-creation. These sources are slightly different from sources used by professional teams. Professional teams seem to use mostly in-house databases (experience repositories) containing knowledge from past projects (Basili, Caldiera & Rombach 1994; Spohrer *et al.* 2013). However, the sources that students used were also good enough because they enabled the teams to create new knowledge.

Channels of communication that facilitate knowledge-creation

Different communication channels were used by teams during knowledge-creation. Mobile devices such as mobile phones and BlackBerrys and the Internet were mostly used for information transfer that facilitated knowledge-creation. Teams used these channels of communication to share ideas. They used mostly emails, chat services such as WhatsApp application (WhatsApp), BlackBerry Messenger (BBM), Mxit and social networking sites (Facebook and Twitter). They stated that they used some of these channels at their convenient time and places to look for information and that they shared it faster than other devices. These results are consistent with results found by Kyobe and Shongwe (2011) who concluded that students use devices such as mobile phones and social networks for learning purposes. Juárez-Ramírez and Ocegueda-Miramontes (2013) also found similar results. They concluded that social networks (Facebook in particular)

are used widely by students in software-development to solve problems and share knowledge.

Challenges of knowledge-creation

There were a number of challenges facing the teams involved in knowledge-creation. The major challenge was financial resources. Before the start of the project, teams had to identify a client for whom they had to develop a system. The client had to be rurally based. This meant that the teams had to travel to rural areas to find the client and collect information about their business before developing the system. All the teams indicated that they did not have enough money for such an exercise. The average distance they had to travel was 50 km. They did not have enough money for travelling and subsistence. They also lacked financial resources for doing activities such as photocopying and printing documents. One team stated that '... we did not have money to go and interview the client the way we could have liked. We also did not have money for printing our documents'. Another group concurred that '...as a group, we did not have the funds to help us successfully achieve our goal'.

Another challenge that the teams faced was a lack of technical skills. This is not a surprise because these were amateur teams. One team stated that '... we had no skills and experience of web design'. The main technical problem that most teams faced was connecting the database to the website (the WAMP platform).

A lack of material resources was another challenge that the teams faced. The material resources that they lacked were computers, software and a dedicated computer laboratory for their project. Some teams had only one laptop computer to work with. The university's computer laboratory did not have the PHP software, which made it difficult for teams to complete the project. The teams also complained greatly about the ever-crowded school laboratories. One group indicated: 'We encountered the challenge of limited resources that would have helped with the organising, planning, and research about the project'.

Time pressure was another big challenge that student teams faced. Teams stated that there was not enough time to complete the project. They complained that each milestone was to be submitted earlier than they would have liked. They also complained that they had to do courses other than the project. One student lamented: 'I had to juggle between activities of the project and other courses'. Another one concurred: 'The challenge was to balance between the amount of work the project had and studying for other courses'.

Not all students were resident students. Some teams were made up of on-campus and off-campus students. Off-campus students frequently had to travel from off-campus to the campus for group meetings. At times, they had to leave meetings earlier in order for them to travel off-campus. This caused challenges because it delayed the completion of the project. Some group members could not turn up for

meetings, which also affected the quality of ideas contributed to the team. One student stated that:

... since I am staying off-campus, my difficulty was always travelling to university and going back in the late hours of the day when it is not safe out there.

Other teams complained of lazy team members who did not complete their given activities. Strode and Clark (2007) report similar challenges in their study of student projects. They report that student projects have the following constraints: time and commitment, experience level, scope and complexity, technology and the need to meet the assessment criteria. Issues of time, commitment and technology seemed to be serious challenges to students. Wang (2009) also found that a lack of time was a major challenge in students' software projects. Faculty should look at how this could be addressed. However, these challenges are also faced by professional teams (e.g. time and financial challenges) as reported by Rus and Lindvall (2002).

Successful completion of the project

The teams were then asked to indicate whether they have successfully completed the project. All the teams indicated that they did complete their projects successfully. They indicated that they have successfully built a working e-commerce interactive website. One group stated that, '... although it was hard, we successfully created an e-commerce site that will help our client expand their business'. Another group stated: 'We were able to create a website that is working'. The groups indicated that, although they successfully completed the project, it was a challenging task.

Knowledge gained from the project

All the teams indicated that they had learnt a lot from the project. They indicated that the project had improved their knowledge and experience in software-development ('I have learned how to create a complete website'). Some respondents indicated that they had learnt to work in a team, a skill that is essential in the job market: 'I believe I gained both knowledge and experience when it comes to programming and working in a team'. Others indicated that they had learnt to work under pressure and in a group to achieve a singular goal.

A summary of the research findings is presented in Table 1.

Recommendations and further research

I recommend, therefore, that university departments engaged in such projects should look at the challenges that student teams face because they affect learning. Departments are also encouraged to formally adopt technology devices such as smart phones for teaching because they play an important role in learning. Further research is recommended to investigate the quality of the knowledge that students create and also to investigate the impact that the challenges have on knowledge-creation.

TABLE 1: A summary of research findings.

Research questions	Findings
What knowledge-creation activities take place in student teams?	At the individual level: coding, documentation, preparing slides for presentations, designing the user interface, research and information gathering, sourcing finance. At team level: brainstorming which led to the refinement, consolidation and adoption of individual tasks.
What channels of communication facilitate knowledge-creation in student teams?	Emails, chat services such as Whats Application (WhatsApp), BlackBerry Messenger (BBM), Mxit and social networking sites (Facebook and Twitter).
What information sources do students use to create knowledge?	Class presentations, the lecturer, lecture notes, text books, the Internet and external sources (senior students and professional software developers).
What challenges do student teams face when they create knowledge?	Lack of material and financial resources, lack of technical skills, lack of time, students staying off-campus and ambivalent team members.

Limitations of the study

The major limitation of this study is that it was conducted in only one university and in only one department. This compromises the representativeness of the total population.

Conclusion

The article contributes to the literature by showing how knowledge is created in student software-development teams. The literature on knowledge-creation covers mostly knowledge-creation in professional teams in business organisations. The article presents a different perspective by focusing on knowledge-creation in amateur teams in an education institution. Several conclusions are drawn from the study: (1) Student software-development teams can create knowledge (a working system) just like professional software-development teams; (2) knowledge is created by the whole team, but the knowledge-creation process starts from individual team members; (3) class presentations, lectures, the Internet and senior students and professional developers are major sources of information that students consult to create knowledge; (4) a lack of technical skills, limited time and a lack of resources are major hindrances to knowledge-creation.

Acknowledgements

Competing interests

The author declares that there are no financial and personal relationship(s) that may have inappropriately influenced him when he wrote this article.

References

- Alavi, M. & Leidner, D.E., 2001, 'Knowledge management systems: Issues, challenges and benefits', in S. Barnes (ed.), *Knowledge management systems: Theory and practice*, pp. 15–32, Thompson Learning, Oxford.
- Arent, J. & Norbjerg, J., 2000, 'Software improvement as organisational knowledge creation: A multiple case analysis', Proceedings of the 33rd International Conference on System Sciences, Hawaii, USA, 4–7 January 2000, p. 105.
- Argyris, C. & Schon, D., 1978, *Organisational learning: A theory of action perspective*, Addison-Wesley Publishing Company, Reading.
- Bailin, S., 1997, 'Software development as knowledge creation', *International Journal of Applied Software Technology* 3(1), 75–89.
- Basili, V.R., Caldiera, G. & Rombach, D.H., 1994, 'The experience factory', *Encyclopedia of Software Engineering* 2, 469–476.
- Chaffey, D. & Wood, S., 2005, *Business information management: Improving performance using information systems*, Pearson Education Limited, London.
- Dingsoyr, T., 2002, 'Knowledge management in medium-sized software consulting companies', *Empirical Computing* 7(4), 383–386.
- Dorairaj, S., Noble, J. & Malik, P., 2012, 'Knowledge management in distributed agile software development', Paper presented at the Agile Conference, Dallas, Texas, United States of America, 13–17 August.
- Elfvig, S. & Funk, P., 2006, 'Enabling knowledge transfer in product development and production: Methods and techniques from artificial intelligence', Paper presented at the 1st Nordic Conference on Product Lifecycle Management, Goteborg, Sweden, 25–26 January.
- Gladstone, B., 2000, *From know-how to knowledge: The essential guide to understanding and implementing knowledge management*, Industrial Society, London.
- Grant, R.M., 1996, 'Toward a knowledge-based theory of the firm', *Strategic Management Journal* 17, 109–122. <http://dx.doi.org/10.1002/smj.4250171110>
- Hedlund, G., 1994, 'A model of knowledge management and the N-Form corporation', *Strategic Management Journal* 15, 73–90. <http://dx.doi.org/10.1002/smj.4250151006>
- Heeks, R., 2002, 'Information systems and developing countries: Failure, success, and local improvisations', *The Information Society* 18(2), 101–112. <http://dx.doi.org/10.1080/01972240290075039>
- Hey, J., 2004, 'The data, information, knowledge, wisdom chain', viewed 16 July 2008, from <http://scholar.google.co.za/scholar?q=The+Data,+Information,+Knowledge,+Wisdom+Chain+%2Bhey+2004&hl=en&um=1&ie=UTF-8&oi=scholar>
- Holsapple, C.W. & Singh, M., 2001, 'The knowledge chain model: Activities for competitiveness', *Expert Systems with Applications* 20(1), 77–98. [http://dx.doi.org/10.1016/S0957-4174\(00\)00050-6](http://dx.doi.org/10.1016/S0957-4174(00)00050-6)
- Hsieh, H-F. & Shannon, S.E., 2005, 'Three approaches to qualitative content analysis', *Qualitative Health Research* 15(9), 1277–1288. <http://dx.doi.org/10.1177/1049732305276687>
- Juárez-Ramírez, R. & Ocegueda-Miramontes, R.P.-R.V., 2013, 'Supporting the software development process using social media: Experiences with student projects', Proceedings of the Institute of Electrical and Electronics Engineers 37th Annual Computer Software and Applications Conference Workshops, Kyoto, Japan, 22–26 July 2013.
- Kess, P. & Haapasalo, H., 2002, 'Knowledge creation through a project review process in software production', *International Journal of Production Economics* 80(1), 49–55. [http://dx.doi.org/10.1016/S0925-5273\(02\)00242-6](http://dx.doi.org/10.1016/S0925-5273(02)00242-6)
- Kimmerle, J. Cress, U. & Held, C., 2010, 'The interplay between individual and collective knowledge: Technologies for organisational learning and knowledge building', *Knowledge Management Research & Practice* 8(1), 33–44. <http://dx.doi.org/10.1057/kmp.2009.36>
- Klint, P. & Verhoef, C., 2002, 'Enabling the creation of knowledge about software assets', *Data & Knowledge Engineering* 41(2/3), 141–158. [http://dx.doi.org/10.1016/S0169-023X\(02\)00038-1](http://dx.doi.org/10.1016/S0169-023X(02)00038-1)
- Kyobe, M.E. & Shongwe, M.M., 2011, 'Investigating the extent to which mobile phones reduce knowledge transfer barriers in student project teams', *South African Journal of Information Management*, 13(1).
- Lee, A.S., 1991, 'Integrating positivist and interpretive approaches to organisational research', *Organisational Science* 2(4), 242–365. <http://dx.doi.org/10.1287/orsc.2.4.342>
- Lee, G.K. & Cole, R.E., 2003, 'From a firm-based to a community-based model of knowledge creation: The case of the Linux kernel development', *Organisation Science* 14(6), 633–694. <http://dx.doi.org/10.1287/orsc.14.6.633.24866>
- Li, M. & Gao, F., 2003, 'Why Nonaka highlights tacit knowledge: A critical review', *Journal of Knowledge Management* 7(4), 6–14. <http://dx.doi.org/10.1108/13673270310492903>
- Linden, T. & Cybulski, J., 2009, 'Knowledge creation in an application domain: A hermeneutic study in ICKM 2009', Proceedings of the 6th International Conference on Knowledge Management: Managing knowledge for global and collaborative innovations, Hong Kong, China, 3–4 December 2009, pp. 1–13.
- Lindvall, M., Rus, I. & Sinha, S., 2002, 'Technology support for knowledge management', in S. Henninger & F. Maurer (eds.), *Advances in learning software organization: Lecture notes in computer science*, pp. 94–113, Springer Verlag, Berlin.
- Lyytinen, K. & Robey, D., 1999, 'Learning failure in information systems development', *Information Systems Journal* 9(2), 85–101. <http://dx.doi.org/10.1046/j.1365-2575.1999.00051.x>
- Mitchell, R. & Boyle, B., 2010, 'Knowledge creation measurement methods', *Journal of Knowledge Management* 14(1), 67–82. <http://dx.doi.org/10.1108/13673271011015570>
- Mitchell, R. & Nicholas, S., 2006, 'Knowledge creation in groups: The value of cognitive diversity, transactive memory and open-mindedness norms', *Electronic Journal of Knowledge Management* 4(1), 67–74.
- Morner, M. & Von Krogh, G., 2009, 'A note on knowledge creation in open-source software projects: What can we learn from Luhmann's theory of social systems',

- Systems Practice and Action Research* 22(6), 431–443. <http://dx.doi.org/10.1007/s11213-009-9139-7>
- Myers, M.D., 1999, 'Investigating information systems with ethnographic research', *Communications for the Association of Information Systems* 2(23), 1–20.
- Myers, M.D. & Young, L.W., 1997, 'Hidden agendas, power, and managerial assumptions in information systems development: An ethnographic study', *Information Technology & People* (10)3, 224–240. <http://dx.doi.org/10.1108/09593849710178225>
- Neves, F.T., Correia, A.M.R., Rosa, V.N. & Neto, M., 2011, 'Knowledge creation and sharing in software development teams using Agile methodologies: Key insights affecting their adoption', Proceedings of the 6th Information Systems and Technologies Iberian Conference (CISTI), Chaves, Portugal, 15–18 June 2011, pp. 1–6.
- Nickols, F.W., 2000, 'The knowledge in knowledge management', in J.W. Cortada & J.A. Woods (eds.), *The knowledge management yearbook 2000–2001*, pp. 12–21, Butterworth-Heinemann, Boston.
- Nonaka, I., 1994, 'A dynamic theory of organisational knowledge creation', *Organisation Science* 5(1), 14–37. <http://dx.doi.org/10.1287/orsc.5.1.14>
- Nonaka, I. & Takeuchi, H., 1995, *The knowledge-creating company*, Oxford University Press, New York.
- Nonaka, I., Toyama, R. & Konno, N., 1998, 'SECI, Ba and Leadership: A unified model of dynamic knowledge creation', *Long Range Planning* 33(1), 5–34. [http://dx.doi.org/10.1016/S0024-6301\(99\)00115-6](http://dx.doi.org/10.1016/S0024-6301(99)00115-6)
- Olikowski, W.J. & Baroudi, J.J., 1991, 'Studying information technology in organisations: Research approaches and assumptions', *Information Systems research* 2(1), 1–28. <http://dx.doi.org/10.1287/isre.2.1.1>
- Parent, M., Gallupe, R.B., Salisbury, Wm.D. & Handelman, J.M., 2000, 'Knowledge creation in focus groups: Can group technologies help?', *Information & Management* 38(1), 47–58. [http://dx.doi.org/10.1016/S0378-7206\(00\)00053-7](http://dx.doi.org/10.1016/S0378-7206(00)00053-7)
- Phan, P.H. & Peridis, T., 2000, Knowledge creation in strategic alliances: Another look at organisational learning', *Asian Pacific Journal of Management* 17(2), 201–222. <http://dx.doi.org/10.1023/A:1015857525048>
- Poh, L-G. & Erwee, R., 2004, Knowledge creation and integration in project teams: A study of international telecommunications companies in Singapore', *Journal of Management & World Business Research* 1(1), 23–44.
- Polanyi, M., 1962, 'Tacit knowing: It's bearing on some problems of philosophy', *Reviews of Modern Physics* 34(4), 601–616. <http://dx.doi.org/10.1103/RevModPhys.34.601>
- Rumizen, M.C., 2002, *The complete idiot's guide to knowledge management*, CWL Publishing Enterprise, Madison.
- Rus, I. & Lindvall, M., 2002, 'Knowledge management in computing', *Institute of Electrical and Electronic Engineers Software* 19(3), 26–38.
- Senge, P.M., Ross, R., Smith, B., Roberts, C. & Kleiner, A., 1994, *The fifth discipline fieldwork: Strategies and tools for building a learning organisation*, Doubleday Inc. Publishers, New York.
- Spohrer, K., Kude, T., Schmidt, C.T. & Heinzl, A., 2013, 'Knowledge creation in information systems development teams: The role of pair programming and peer code review', *Proceedings of the 21st European Conference on Information Systems*, Utrecht, Netherlands, 6–8 June 2013, pp. 1–12.
- Strode, D.E. & Clark, J., 2007, 'Methodology in software development capstone projects', *Proceedings of the 20th Annual National Advisory Committee on Computing Qualifications*, Nelson, New Zealand, 8–11 July 2007, pp. 243–251.
- Styhre, A., Roth, J. & Ingelgard, A., 2002, 'Care of the other: Knowledge creation through care in professional teams', *Scandinavian Journal of Management* 18(4), 503–520. [http://dx.doi.org/10.1016/S0956-5221\(01\)00022-7](http://dx.doi.org/10.1016/S0956-5221(01)00022-7)
- Suchman, L., 1995, 'Making work visible', *Communications of the Association for Computing Machinery* 38(9), 56–64. <http://dx.doi.org/10.1145/223248.223263>
- Wang, A.I., 2009, 'Post-mortem analysis of student game projects in a software architecture course: Successes and challenges in student software architecture game projects', Paper presented at the International Consumer Electronic Games Innovations Conference, London, England, United Kingdom, 25–28 August.
- Wan, J., Zhang, H., Wan, D. & Huang, D., 2010, 'Research on knowledge creation in software requirement development', *Journal of Software Engineering & Applications* 3(5), 487–494. <http://dx.doi.org/10.4236/jsea.2010.35055>
- Wasko, M. & Faraj, S., 2000, 'It is what one does: Why people participate and help others in electronic communities of practice', *The Journal of Strategic Information Systems* 9(2/3), 155–173. [http://dx.doi.org/10.1016/S0963-8687\(00\)00045-7](http://dx.doi.org/10.1016/S0963-8687(00)00045-7)
- Wenger, E., 1998, *Communities of practice: Learning, meaning, and identity*, Cambridge University Press, Cambridge. <http://dx.doi.org/10.1017/CBO9780511803932>