Issues in Informing Science + Information Technology

An Official Publication of the Informing Science Institute
InformingScience.org

IISIT.org

Volume 14, 2017

THE USE OF KANBAN TO ALLEVIATE COLLABORATION AND COMMUNICATION CHALLENGES OF GLOBAL SOFTWARE DEVELOPMENT

Maureen Tanner* University of Cape Town, South Africa mc.tanner@uct.ac.za
Marcelo Dauane University of Cape Town, South Africa DNXMAR004@myuct.ac.za
* Corresponding author

ABSTRACT

Aim/Purpose	This paper aims to describe how various Kanban elements can help alleviate two prominent types of challenges, communication and collaboration in Global Software Development (GSD).
Background	Iterative and Lean development methodologies like Kanban have gained significance in the software development industry, both in the co-located and globally distributed contexts. However, little is known on how such methodologies can help mitigate various challenges in that occur in a globally distributed software development context.
Methodology	The study was conducted using a single-case study based on a general inductive approach to analysis and theory development. Through the literature review, collaboration and communication challenges that GSD teams face were identified. Data collected through semi-structured interviews was then inductively analyzed to describe how the case-study teams employed various Kanban elements to mitigate communication and collaboration challenges they face during GSD.
Findings	The study found that some Kanban elements, when properly employed, can help alleviate collaboration and communication challenges that occur within GSD teams. These relate to Inclusion Criteria, Reverse Items, Kanban Board, Policies, Avatars, and Backlog.
Contribution	The paper contributes to knowledge by proposing two simple concept maps that detail the specific types of communication and collaboration challenges which can be alleviated by the aforementioned Kanban elements in GSD.
Recommendations for Practitioners	This paper is relevant to GSD teams who are seeking ways to enhance their team collaboration and communication as these are the most important ele-

Accepting Editor: Eli Cohen | Received: November 30, 2016 | Revised: January 23, January 26, March 24, 2017 | Accepted: March 28.
Cite as: Tanner, M., * Dauane, M. (2017). The use of Kanban to alleviate collaboration and communication challenges of global software development. *Issues in Informing Science and Information Technology Education, 14,* 177-197. Retrieved from http://www.informingscience.org/Publications/3716

ments that contribute to GSD project success. It is recommended that relevant Kanban elements be used to that effect, depending on the challenges that they aim to alleviate.

Future Research — Future research can investigate the same research questions (or similar ones) using a quantitative approach.

Keywords — Kanban, lean software development, global software development, communication challenges, collaboration challenges

INTRODUCTION

The development of software projects through interactions of organisations, people and technology across geographical boundaries, organisational and national cultures, languages and working styles is known as Global Software Development (GSD) (Tanner, 2009). GSD team members are generally from diverse cultures and work together while located in different locations and time zones (Jarvenpaa & Leidner, 1998). This form of work setup is implemented through IT outsourcing (Mohagheghi, 2004) and is enabled through computer-mediated technologies (Jarvenpaa & Leidner, 1998).

GSD is widely used in the software industry due to the numerous benefits that it offers to businesses (Scharff, 2011). These benefits range from reduced cost to improved software quality, resulting in a competitive advantage that business organisations are constantly seeking (Niazi et al., 2013; Scharff, 2011). Mergers and acquisitions are also influencing the shift towards GSD, resulting in the production of innovative products, new markets, and access to a more diverse pool of software developers (Sudhakar, Farooq, & Patnaik, 2011). Recently, the outsourcing of software development experienced exceptional growth especially within European companies outsourcing to Indian IT companies (Søderberg, Krishna, & Bjørn, 2013). Søderberg et al. (2013) supported the reasons for such growth are not only the labour cost reduction and high quality of software produced by Indian IT companies, but also includes the long-term partnerships and the knowledge that these offshore companies have to offer. However, teams working in GSD face numerous collaboration and communication challenges (Bannerman, Hossain, & Jeffery, 2012; Bjarnason, Wnuk, & Regnell, 2011; Herbsleb, 2007). These categories of challenges are very important subjects of study because human relations are crucial for the success of GSD projects. In fact, even more important than technical skills (Sudhakar et al., 2011).

A practice that aids in mitigating the aforementioned challenges, both in co-located and in GSD teams, is the use Agile methodologies (Bannerman et al., 2012; Wang, Conboy, & Cawley, 2012). In GSD, Agile methodologies are being used in response to the fast-paced changes that occur in software development projects, with added focus on collaboration and communication challenges occurring within that context (Kaur & Sharma, 2014).

Lean approaches form a subset of Agile approaches that focus on eliminating waste (Ebert, Abrahamsson, & Oza, 2012). Lean software development starts with value orientation, then reducing unnecessary features, improving the interfaces, empowering the software developers and continuously improving the solutions (Ebert et al., 2012).

Kanban is a software development methodology which applies Lean principles (Ahmad, Markkula, & Oivo, 2013; Ikonen, Pirinen, Fagerholm, Kettunen, & Abrahamsson, 2011). Kanban is becoming increasingly popular (Ahmad et al., 2013) and is being used to enhance Scrum and other Agile methodologies (Ahmad, Markkula, Oivo, & Kuvaja, 2014). Although Kanban's popularity is increasing, many questions with regard to its adoption in software development are still not answered. Practitioners face serious challenges while implementing Kanban as clear definitions of its practices, principles, techniques, and tools are lacking (Mahnic, 2014).

Studies have shown that in co-located settings, Kanban can help promote communication and collaboration especially when the teams come together and do not yet know each other (Oza, Fager-

holm & Münch, 2013). However, little is known how Kanban can assist in the mitigation of communication and collaboration challenges in GSD. Literature suggests that Kanban is used to enhance Scrum and other existing agile methods (Ahmad et al., 2014), but it has not yet been established how such enhancements are experienced within the GSD context. This study seeks to address this gap by not only describing how Kanban elements are used in GSD but also by shedding light on the use of these elements to mitigate communication and collaboration challenges.

The findings might be useful to GSD and Lean practitioners who are looking for ways of leveraging off Kanban as a methodology and improve the outcome of their GSD projects by reducing communication and collaboration challenges. By providing a deeper insight on existing Kanban elements and the benefits that they might bring to GSD, practitioners might have a better understanding of how to leverage off that methodology and the practices that it proposes.

The paper is organised as follows. First an overview on literature on GSD, Lean and Kanban is proposed. The methodology employed for the study is then described followed by a detailed description of the findings. The paper is then concludes with an overview of the findings, contributions, and recommendations for future research.

LITERATURE REVIEW

In the literature review, an overview of GSD and GSD teams as well as the communication and collaboration challenges experienced in GSD will be provided. Lean software development, with particular emphasis on Kanban elements is then described.

GSD & GSD TEAMS

GSD is swiftly growing with explicit interest from academia and industry (Bannerman et al., 2012; Hanssen, Šmite, & Moe, 2011; Portillo-Rodríguez, Vizcaíno, Piattini, & Beecham, 2012). Bannerman et al. (2012) defined GSD as software development that is scattered across numerous locations and separated by national borders. A study from Richardson, Casey, McCaffery, Burton, and Beecham (2012) characterised this environment as having been plagued by geographical, temporal, and cultural distances. The fact that this practice is growing leads to an increased number of software developers expected to engage in a dispersed environment (Richardson et al., 2012) leading to the emergence of GSD teams or Global Virtual Teams.

A team can be defined as a group of at least two people who engage in a dynamic, interdependent, and adaptive manner towards a common objective, where each person of the group has a specific role (Costa, Passos, & Bakker, 2014). Teams in the software development environment are unique since the projects they engage with are complex, invisible, and very flexible (Sudhakar et al., 2011).

GSD teams are software development teams distributed across national boundaries (Layman, Williams, Damian, & Bures, 2006). Several concepts, such as offshore team or global virtual team, are related to the idea of a GSD team (Anh, Cruzes, & Conradi, 2012). These teams find themselves in numerous scenarios where communication between them is often electronic, asynchronous, with limited face-to-face and informal interactions (Anh et al., 2012). GSD teams are affected by geographical, temporal, and cultural distances. For example, geographical distance indirectly affects the performance of a GSD team due to the coordination problems that it creates (Anh et al., 2012).

COLLABORATION AND COMMUNICATION CHALLENGES

As previously mentioned, GSD teams face cultural, geographical, and temporal distances that introduce barriers and complexity to software development and that negatively impact coordination, visibility, communication, and cooperation between the teams (Richardson et al., 2012). These distances were also mentioned in numerous other studies (Bannerman et al., 2012; Colomo-Palacios, Soto-Acosta, García-Peñalvo, & García-Crespo, 2012; Hanssen et al., 2011; Niazi et al., 2013). According to Herbsleb, (2007), as the temporal and geographical distances increase, the effectiveness of collabo-

ration and communication decreases. Collaboration and communication challenges, which are fundamental to good teamwork (Hsu, Shih, Chiang, & Liu, 2012), are further elaborated in the following sub-sections.

Collaboration challenges

In a study done by Herbsleb (2007), it was posited that the vital problem of GSD is the absence or disruption of numerous mechanisms that are responsible for coordinating the work in a co-located environment. In fact, Hersbleb mentioned that even very small physical distances can have sizable repercussions. Portillo-Rodríguez et al. (2012) shared the same view and mentioned that collaboration with teams in dispersed locations is an enormous challenge for organizations. Table 1 provides an overview of the different types of collaboration challenges that GSD teams may face from the perspective of the geographic, cultural, and temporal distances as defined by Hashmi et al. (2011).

Table 1. Collaboration challenges faced by GSD teams (Hashmi et al., 2011).

Distance	Challenges
Cultural	Unequal work distribution
	Lack of Trust
	Increase in project cost
	Reporting problems
	Different working styles
Geographical	Project Delays
	Ambiguity on technical aspects
	Unequal quality levels across the software development sites
Temporal	Lack of Motivation
	Less visibility
	Poor project management
	Chances of loss of project artefact

Communication challenges

Communication is regarded by numerous researchers (Bjarnason et al., 2011; Sudhakar et al., 2011) as being a crucial factor for the success of software development projects. Furthermore, communication is a challenge that is faced by every software development team regardless of where they are located (Niinimäki, Piri, Lassenius, & Paasivaara, 2012). Taking into consideration the fact that GSD teams are physically located in different places, this challenge is acerbated for them (Niinimäki et al., 2012). The cultural, geographical, and temporal distances faced by GSD teams lead to the following communication challenges identified by Korkala and Maurer (2014) as presented in Table 2.

Table 2. Communication Challenges of GSD (Korkala & Maurer, 2014).

Distance	Challenges
Cultural	Misunderstandings in communication stemming from cultural differences. Cultural differences may result in situations situations such as disagreements that are not willingly expressed and negative issues that are shared reluctantly. Language barriers can significantly hinder communication.
Geographical	Face-to-face meetings are difficult to arrange and informal communication is lacking. This inhibits idea sharing.
Temporal	Opportunities for synchronous communication are reduced. Communication needs to take place in unconventional times due to the lack of overlapping working hours and leads to overtime work. This is consuming and leads to communication overhead. Possible unavailability of remote colleagues when help is needed can lead to delays. Using interactive media for efficient communication can be very difficult due to temporal distance.

Lean Principles

Lean thinking has principles that support value creation for customers (Ikonen et al., 2011). The core of lean is the removal of waste, that is, eliminate non-value added activities (Abdulmalek & Rajgopal, 2007; Jailia, Sujata, Jailia, & Agarwal, 2011). Lean software development is based on the application of principles from Lean manufacturing (Ebert et al., 2012). Lean software development starts from value orientation, then reducing unnecessary features, improving the interfaces, empowering the software developers and continuously improving the solutions.

The impact of lean methods in software development are still not completely understood, even though these are commonly adopted with a combination of other agile methodologies (Ebert et al., 2012). According to Dingsøyr, Nerur, Balijepally, and Moe (2012), although leanness focuses on cost reduction by eliminating waste, agility treats leanness in the sense that the elimination of waste is a manner of creating valuable and effective results. Hence, leanness can be perceived as efficiency oriented, while agility involves embracing lean processes with an emphasis on realising effective results (Dingsøyr et al. 2012). Kanban is a lean approach that demonstrates how lean practices are beneficial to software development, (Petersen & Wohlin, 2011).

Kanban Elements

Kanban is a Japanese word which literally means "signboard". The Kanban methodology was introduced by the manager at Toyota Motors with the aim of limiting the inventory level at each stage of the production process through the use of cards (Gaury, Pierreval, & Kleijnen, 2000).

Kanban is popular within the software development industry because of its ease of implementation, use of visual controls, efficient management of work in progress, and relentless focus on continuous process improvement (Mahnic, 2014). The success of Kanban is also dependent on the implementation of explicit policies and feedback loops (Al-Baik & Miller, 2014). Overall, Kanban is known as the best method for executing lean thinking in practice (Chai, 2008).

In contrast to other Agile methodologies like the Rational Unified Process (RUP), XP, and Scrum, Kanban is more adaptive as opposed to being prescriptive. RUP is very prescriptive and recommends over 30 roles, 20 practices, and 70 artifacts. As a result, the methodology is not easily implemented in

practice. XP is less prescriptive than RUP but more prescriptive than Scrum. Scrum is considered to be less prescriptive as it does not recommend any specific software engineering practices. Finally Kanban is the least prescriptive agile methodology and only introduces constraints related to the visualization of the workflow and the limiting of Work in Progress. Interestingly, this lack of prescribed practices is seen as a powerful feature of Kanban (Kniberg, 2009). The following paragraphs explain how Kanban allows for the visualization of the workflow and controls the work in progress.

The ***Kanban board*** is a tool used to visualise the workflow of the Kanban methodology (Ikonen et al., 2011). This tool supports the lean principle of "optimise the whole" by demonstrating what needs to be done, the work in progress (WIP), and what has been completed (Ikonen et al., 2011). The Kanban board has columns that represent workflow stages of the development process and the number of user stories in each column is limited in order to manage the workflow (Khan, 2014). In this context, a user story is a simple and natural language explanation of a system feature (Layman et al., 2006). The Kanban board could have columns for the tasks to be completed, tasks in progress, tasks in testing, and done tasks. Each user story has one or more tasks related to it. The developers solely focus on a user story that is in progress and ensures that the work items are completed before starting working on a new user story (Mahnic, 2014). The user story moves to the next column when completed, thus "pulling" other stories from the previous column. This is known as the **Pull System**. An example of a Kanban board is provided in Figure 1.

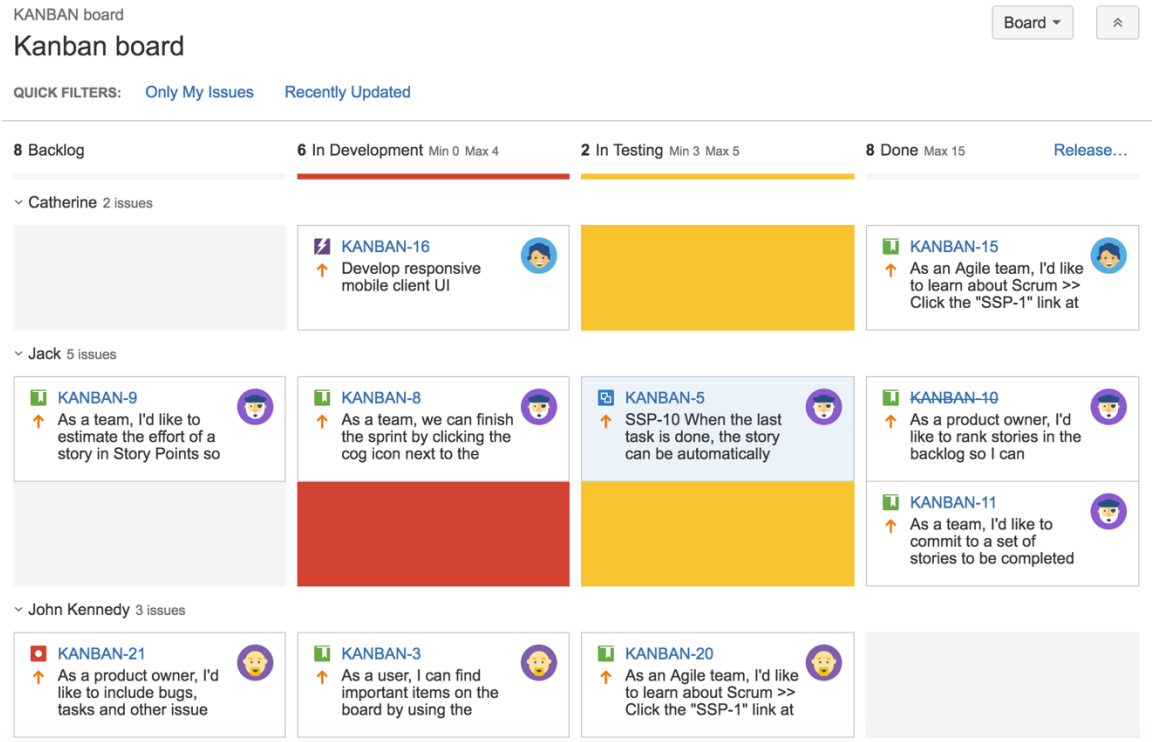

Figure 1. Example of a Kanban board used in a software development project (Li, 2016)

In the Pull System, the development process is only initiated following a customer request (Al-Baik & Miller, 2014). One of the requirements of this approach is the setting up the ***Work-in-Progress (WIP)*** limit by constraining the number of tasks in each prioritised queue. Work in progress refers to the number of work items that may be in progress at each workflow state (Kniberg & Skarin, 2010). It is advisable to pull work items with the highest priority and have a WIP limit since a task is only pulled from one column to the next if the WIP limit is not reached (Al-Baik & Miller, 2014; Polk, 2011).

The ***Backlog*** or prioritised queue is a list that contains work items that are yet to be processed (Al-Baik & Miller, 2014; Turner, Madachy, Ingold, & Lane, 2012). Different criteria may be applied to prioritise these queues (Al-Baik & Miller, 2014). Al-Baik and Miller (2014) defend that these may be prioritised by importance, urgency, or value. Moreover, the ***Inclusion Criteria***, one of the most vital elements of Kanban, ensures that every work item added to the backlog and ultimately to the Kanban board creates value to the customer (Al-Baik & Miller, 2014).

Done Item refers to a work item that is considered completed. Al-Baik and Miller (2014) considered this element to be a dominant contributor to the continuous workflow of Kanban. On the other hand, a ***Reverse Item*** is a work item that is moved to any previous state of the workflow rather than forward (Al-Baik & Miller, 2014). There can be a point where no further work items can be pulled from the previous state. This is known as a ***Bottleneck*** (Al-Baik & Miller, 2014). Bottlenecks are easily identifiable when visualising the Kanban board (Middleton & Joyce, 2012) and can be resolved by breaking the work items into smaller items, thus realising their value incrementally (Al-Baik & Miller, 2014). Bottlenecks can also be reduced through the use of ***Slack or Buffer*** (Ericsson & Granlöf, 2011). A Slack or Buffer is a list of work items that cannot be controlled by the team and are consequently not yet actionable (Ericsson & Granlöf, 2011).

Cycle Time or ***Lead Time*** is a term whose definition researchers have not yet reached an agreement on. Kniberg and Skarin (2010) defined this as the average time it takes to complete a task. Some researchers define it as the time taken to start and complete a feature and others as the time between deliveries of tasks (Al-Baik & Miller, 2014). Regardless of what the exact definition is, this element is argued by Al-Baik and Miller (2014) as being the motivation for process effectiveness and efficiency because it is used to measure overall performance. The ***Performance Measurement Tools*** are used to ascertain the performance of the Kanban project (Al-Baik & Miller, 2014). Al-Baik and Miller (2014) found in their study that performance can be measured daily through burndown charts or with cumulative flow diagrams based on the WIP and lead time.

Validated Learning is a process used to measure the value of a feature that was completed from the perspective of the business (Al-Baik & Miller, 2014; Reis, 2011). This should be supported by data collected from real customers. Researchers suggest that in assessing the success of a project, learning and monitoring are key performance indicators (Al-Baik & Miller, 2014). ***Waste*** is defined as every element in the project that does not produce value to the customer (Ahmad et al., 2013; Ericsson & Granlöf, 2011). Extra processes, features, defects, and partially completed work are also considered as waste (Ahmad et al., 2013).

Similarly to other agile methodologies, Kanban recommends the implementation of ***Stand-Up Meetings***. However, there are divergent interpretations regarding the importance of this element. Some researchers believe that meetings are a source of waste while others recommend daily stand-up meetings to discuss the status of the project (Al-Baik & Miller, 2014; Dybå & Dingsøyr, 2008). The daily stand-up meetings are usually 10 to 15 minutes long (Middleton & Joyce, 2012). ***Planning Meetings*** can also be held and also generate divergent views from researchers. For example, some researchers state that they are a source of waste as they can take up to two-thirds of the production process cycle time. Others perceive this element as important because of the improvement that can be achieved from the feedback of team members during the meetings (Al-Baik & Miller, 2014).

Kanban also advocates the use of ***Feedback Loops***. Kniberg and Skarin (2010) explained the feedback loop process as follows: change something, find out how it went, learn from it, and change it again. Researchers identified that the importance of feedback loops stems from gathering information on customers' experience with the product. It is also important to have a short feedback loop to adapt the processes quickly (Kniberg & Skarin, 2010).

An ***Avatar*** is a visual representation of a team member on the Kanban board. An example could be an image or a magnetic button with the initials of a team member (Ericsson & Granlöf, 2011; Filho & Toledo, 2015). Its purpose is to highlight the work item that each member is working on at a spe-

cific moment (Ericsson & Granlöf, 2011). This enables management to stay informed about resource availability and capacity, thus helping in the assignment and scheduling of tasks (Al-Baik & Miller, 2014).

Lastly, all of the above Kanban elements should be documented as *Policies*. Policies are rules that outline what should be completed and how it should be done. It is important that every team member follows the policies in place (Ericsson & Granlöf, 2011).

Benefits and Challenges of Kanban

Ahmad et al. (2013) report on several benefits of using Kanban during software development. These include the following: Improved understating of entire processes, Refined software quality, Increased focus on customer needs and satisfactions, Increased motivation of developers, Enriched communication and coordination between team and stakeholders, Quicker bug fixes, Improved software productivity, Increased problem-solving ability (easy detection and removal of bugs), Decreased batch size, Reduced time to delivery, Increased release frequency, Efficiently controlled software development projects, Increased ability to manage changes to requirements, and Quicker feedback on features, amongst others.

In contrast, the most Kanban-related challenge is the fact that most people do not have enough experience with the technology, making it difficult for them to select and prioritise tasks as well as manage Work in Progress (Ahmad, et al., 2014). Organisational culture can also be an issue. In essence, the implementation of Lean practices often requires a shift in organizational culture and processes, which is not easily achieved (Al-Baik & Miller, 2014). Another challenge is the lack of guidelines on how Kanban can be employed (Ahmad et al., 2013).

Research Questions

Kanban can alleviate collaboration and communication challenges of software development teams as observed in existing literature. However, this is not yet well understood in GSD teams. Good research questions are the starting point to understand the phenomenon (Agee, 2009). The questions that this research addressed are:

- What are the Kanban elements that can be employed by GSD teams to overcome communication and collaboration challenges?
- How do Kanban elements help alleviate the collaboration challenges that GSD teams face?
- How do Kanban elements help alleviate the communication challenges that GSD teams face?

METHODOLOGY

The techniques used to complete this study are explained in this section. The approach taken to perform the research as well as the strategy, sampling, data collection process, and the data analysis techniques are described.

Research Approach

The study was interpretive, qualitative, and followed an inductive approach. An interpretive philosophy is based on the assumption that social reality is shaped by social contexts and human experiences (Bhattacherjee, 2012). The study was interpretive on the basis that this philosophy is relevant to explore and study context-specific processes, which is the case for this research. Moreover, qualitative research is aimed at understanding phenomena which aligns with the research objectives (Bhattacherjee, 2012). Lastly, the study followed an inductive approach as there is no existing framework that fitted the objectives of the study given that limited research has been conducted on that topic.

Research Strategy

In line with Benbasat, Goldstein, and Mead (1987), there are three reasons why the case study approach was chosen for this study. Firstly, a case study approach enables the researcher to study how Kanban is employed in the GSD setting. Secondly, a case study is well suited for "How" questions as is the case for this study. Lastly, a case study is appropriate when researching an area that is relatively under-researched. Indeed, as shown in the literature review, few studies have focused on the use of Kanban in GSD teams.

The person, collective, or object that will be investigated is referred to as a unit of analysis (Bhattacherjee, 2012). Multiple units of analysis may be used in a single case study to offer extensive analysis and enhance the observations (Yin, 2003). In this research, the unit of analysis was the GSD teams. The case study was conducted in a South African organisation that manages a GSD project using Kanban. The case is described in later in this paper.

Sampling

The study employed purposeful sampling, whereby the case was selected on the knowledge, experience and information that could be provided (Marshall, 1996). It was important to select an organisation that was involved in GSD and thus employed a GSD team setup, and also used Kanban as a methodology for their software projects. Convenience sampling was also applied because the study was dependent on the willingness and availability of the participants (Marshall, 1996). Forums, meetups, search engines, as well as virtual communities that shared information about software development and project management were reviewed.

Data Collection

For this study, data was collected using semi-structure qualitative interviews. Qualitative interviews can provide a profound understanding of the nuanced, contextual, and authentic representation of participants' experiences and how it is interpreted by them (Schultze & Avital, 2011). The interviews were conducted to obtain an understanding of team members' views on the use of Kanban to overcome communication and collaboration challenges they experience throughout their projects. Open-ended questions were posed to allow the participant to share as much knowledge as possible. The researcher used incomplete scripts and left space for improvisation, which are common practices in semi-structured interviews (Myers & Newman, 2007). Therefore, additional questions were asked based on the interviewee's responses. The interviews were voice recorded to ensure that all information provided by the participant was saved and to avoid information loss. The participants' profile is described in the section describing the case description.

The questionnaire was split into three sections. The first and second sections focused on understanding how Kanban helps to alleviate the communication and collaboration challenges which the teams experienced. The questions were based on challenges identified in the literature review. The third section focused on gathering information on the organisation, the team, and the projects they worked on. This information was used to compile the case description.

Data Analysis

Data was analysed using the general inductive approach defined by Thomas (2006). Thomas specifies a simple approach for deriving results aligned with the context of the research questions According to Thomas, the purpose of developing a general inductive analysis approach is to (1) summarise extensive and raw data, (2) link the summary generated with the gathered data and the research objectives to ensure that these links are transparent, and (3) create a theory or a model about the experiences that are conspicuous in the data.

CASE DESCRIPTION

The selected case study is described in this section. Details about the company, GSD Teams, and their respective projects are provided.

THE COMPANY

The case study was conducted in an IT company with a background in financial services, based in Cape Town, South Africa (SA). The company was founded in 2014 and employed 10-15 workers at the time of the study. Albeit being small, the company was composed of a software development team, a project management team, and a team of consultant specialising in the support of financial applications. The company mainly built web applications for clients situated in the investments/finance industry in the United Kingdom (UK). At the time of the study, they had two branches, one in Cape Town and another in London, which is where their main clients were located.

THE GSD TEAM

The GSD Team was split across London and Cape Town. The team members in London also included staff from their client and were responsible for requirements gathering and development. The team members located in Cape Town were responsible for planning, development, and management of all the software development projects. The requirements gathered by the team in London were transmitted to the team members in Cape Town, who then built (in collaboration with the London Team Members) and delivered the applications. In addition to that, the team members in Cape Town also maintained the existing applications of the client.

THE PARTICIPANTS

Table 3 summarises the profile of the participants in relation to where they are located, their position in the team, how long they have been in the company, their level of GSD experience, and where the interview was conducted. Most Participants were selected based on their experience and knowledge of GSD projects managed using Kanban. However, in some cases (i.e., A3) the respondents were selected for convenience. It is acknowledged that the sample is small. However, to compensate for this issue, in-depth interviews were conducted.

Table 3. Respondents' Profile

Identifier	Location	Position	Time in Organisation	GSD Experience	Place of Interview
A1	Cape Town	Team Lead	2 years	2+ years	Interviewee's office
A2	Cape Town	Software Developer	1.5 years	1.5 years	Interviewee's office
A3	Cape Town	Software Developer	6 months	2 months	Skype
A4	Cape Town	Project Manager	2.5+ years	2.5 years	Interviewee's office
A5	London	Business Analyst	2 years	2 years	FaceTime

FINDINGS

RESOLVING COLLABORATION CHALLENGES USING KANBAN ELEMENTS

This section describes how various Kanban elements helped in alleviating various collaboration challenges which the GSD team experienced throughout the project.

Poor project management

Project management of changes in requirements and requirements' priorities are difficult to manage in the GSD setting and respondents felt that the Kanban approach was useful in alleviating this issue. A1 explained that Kanban enabled the team to react quickly and proactively to changing requirements and even to newly prioritized requirements which might occur in the middle of the sprint. This relates to Kanban's elements of ***Inclusion Criteria*** and ***Reverse Items.*** The inclusion criteria allows items which might bring the highest value to the customers to be included in the queue, while others might be reversed back to a previous state to make room for the newly prioritized item.

> *"The nice thing about the Kanban approach is that you find that priorities change as you go along and things need to be moved around [….] and you can react quickly to changing dynamics. Maybe tomorrow we find that there is a huge bug in whatever system. We can say '[…] we can't wait for the next sprint. We have to do it now'. So, that is the nice thing about the Kanban methodology. Is that it allows you to be very responsive" (A1).*

In GSD, project managers might also find it difficult to keep track of who is doing what, especially at dispersed sites. The ***Kanban Board*** can bring that visibility by enabling the project manager to visualize who is working on what and who is being delayed on certain tasks. The board also provides relevant information to the project manager enabling them to better coordinate and allocate tasks if need be.

> *"It helps to know who is working on what […]. Say you have been here for a while. You can see how someone is working and you can measure their work style and see whether he is taking a little bit longer on this task. Or you can see someone going through work really quickly. I imagine it would be helpful for a project manager to know someone's work style is or how quickly someone goes through a particular kind of work. Maybe this one was a bit harder but you went through it quite quickly. Just from a project manager perspective, it would help to get to know your developers a little bit better" (A2).*

Project delays

The use of the ***Kanban board*** can assist in reducing project delays that occur due to geographical distances. Moreover, the Kanban board even allows for work completion to be speeded up. The respondents explained that they did not always face a situation where urgent items had to be completed and when team members from one geographical location (e.g., London) were about to finish work and could not attend to those items. But when this did happen, the ***Kanban board*** enabled the Cape Town based team to attend to these items sooner (since they get to the office before the London team) and more efficiently as these items would have been listed there.

> *"There are some cases. When it does happen then when we come in, we see those issues and we can start working on it before they even open shop. [..]. We can have a quality issue, code review, double check, test it out and things can continue from there" (A1).*

The respondents also stated that given the fact that the team worked on multiple projects at the same time, without the use of the ***Backlog*** and the ***Kanban Board***, they might not have been able to keep track of these various projects and quickly identify issues related to these projects whenever they occurred. This would have resulted in project delays. The backlog enabled them to easily keep track of the various requirements that needed to be completed in the various projects that they were running in parallel.

"The communication problems that we face are based around the fact that since people are distributed around different projects, it was hard to put focus in one project. So, if you say you have an issue or you are, it took some time for people to jump on to your issue to give you feedback. It would take time for them to respond. […] If Kanban wasn't there it be a lot harder to keep track of what is on the backlog. What has been done or what is in progress and stuff" (A3)

Unequal work distribution

Unequal work distribution, whereby the team members at one site might feel that they are doing more work that those at the other site, can be an issue in GSD. However, the respondents felt that "fairness" in relation to work distribution does not necessarily relate to the number of items being handled by individual team members, but rather to the level of complexity of that item. One team member might be required to complete one complex task while someone else might handle five easy tasks and this was perceived as fair. They felt that the ***Kanban board*** was useful in achieving that level of fairness, given the fact that team members could see who is working on what and also what were the outstanding tasks that they could handle, given their level of expertise. Without the Kanban board, such visibility regarding the work contribution of each team member would not have been possible, thus creating collaboration challenges related to perceived unequal work distribution within the GSD context.

"Because of the nature of Kanban and how there are always things filling up your Kanban project. As your resources are free they can tackle the next issue. You find that one resource might be struggling with one issue for the whole day so they can only take on one issue whereas another resource can be chugging it out per second. The fairness is not in terms of number but rather in terms of how much can each person can achieve in relation to their abilities or to their own capacities" (A1).

Poor visibility

The ***Kanban board*** also brings visibility on who is working on what, what changes have been implemented, and what needs to be done at any point in time. Such visibility in a GSD context is not as easily achieved as it is the case in a co-located setting and can lead to frustration and resentment if not managed properly.

"If you didn't have Kanban, you wouldn't have visibility of the project. Who is working on what, what needs to be done, etc. So, if you have that in place, I would assume it would eliminate the need to have face to face meetings every day" (A2).

Poor visibility would also have been felt during the sprint planning meeting in GSD without the use of an easily accessible and visible Kanban Board. The accessibility of the board not only increases visibility on various aspects of the project but also allows for meetings to be held at any location, irrespective of how dispersed the team members are.

"Because the most important information of the project is not on their face, it's on the board. So, I don't need to be there physically but we all just need to be looking at the same board. That just makes it easier to have meetings in any location. Because now, for example the sprint planning meetings, which is where I think the majority of the input from a project management perspective, from a developer perspective, from a business requirements perspective, that can now be completely done where the development team is in one place, the project management is in one place and the client can be in another place" (A4)

Different working styles

GSD teams face challenges related to different work styles because of cultural distance, as shown in literature. Some respondents felt that the ***Kanban Board*** could also be useful in that regard given the fact that it enables team members to, for example, work from home if need be. However, it was also noted that the tool alone is not enough and measures should be put in place to keep track of work progress of individual team members.

"Definitely. Again, Kanban is an approach and then you have tools to manage Kanban like JIRA for example. So, if you are using good tools to manage like a person championing the actual approach. You also have tools championing your approach. Very good tools that allow you to access you work from your house and all that. Then in that way, it helps the process, it helps to get things moving but there is still the human element" (A1).

The difference in workstyles might lead to frustrations in GSD if there is a lack of visibility on who is doing what and how much work is being accomplished by dispersed team members. Some people might prefer to work 5 hours straight while others might work in short periods but still get the work done. As explained by A5, the ***Kanban Board*** and the ***policies*** in place on how to use the board eliminates the likelihood of this occurring. For instance, team members are required to log the number of hours they have spent on a task allowing everyone to have a clear representation of the amount of time that any team member has invested on a task, irrespective of where they are located and how they chose to work.

"In terms of work style it does because Kanban allows you to do time logging. You can log how many time you spent on an issue and people manage their time differently and work differently. Some people prefer dividing the issues and work for few hours on one thing and few on the other, and some prefer working straight long hours. So it does have the capability to tracking and accommodate different styles. But it also depends on people actively using that specific functionality to log hours they work on an issue" (A5).

Ambiguity on technical aspects

Because of geographical distance, ambiguity related to various technical aspects might arise during software development. It might be difficult for a team member to keep track of what has been done on a particular piece of code, why, and by whom. Kanban can help by encouraging the implementation of ***feedback loops*** mechanisms. In the team, this was established by encouraging team members to explicitly describe the changes that they have implemented to any piece of code.

"That is not nice when you are not using Kanban properly. But when it is done properly, it makes a difference and shows you exactly how far they are with comments saying I implemented this functionality. That is remaining. Then Kanban is great for communication" (A2)

Resolving Communication Challenges using Kanban Elements

This section describes how various Kanban elements helped in alleviating various communication challenges which the GSD team experienced throughout their project.

Language differences

Language differences (e.g., inability to understand people's accents) can be a major hindrance to smooth project completion during GSD, as shown in literature. In the study, team members were based in London and Cape Town and there were no major challenges relating to English per se. However, some team members originated from different countries (e.g., Zimbabwe, Sudan, Italy). The inability to understand people's accent has been an issue at times. Respondents explained that since Kanban requires that tasks' and issues' descriptions are written down, the accent barrier was eliminated and team members could even Google the definition of words if need be. The need to write down the items and issues description using the right tool (e.g., Jira) is a ***Policy*** that has been adopted by the team. Given that the team members followed that policy adequately, it helped resolve language difference issue when it occurred.

"Especially, again, Kanban being the approach, we use a tool called JIRA, to manage the approach and in the products that we use. We can right down in plain text our problems. So, it does not matter what accent you have, you will always be able to easily understand. If you don't understand the language you can take your time, google the words. I would assume maybe certain people that struggled a little bit probably did that a bit" (A1).

"So, sometimes someone might not be able to understand exactly the words someone uses or obviously the construct of the words while someone describes a requirement. But, obviously, if something is described in writing on the

Kanban board where everyone can collaborate on the specific issue and that can help alleviate any misunderstanding based on someone expresses themselves" (A5)

Reduced opportunities for synchronous communication

In GSD, due to temporal distances, opportunities for synchronous communication are reduced. Kanban can be useful in that instance because it allows for ***policies*** to be put in place by the team to promote a medium of communication that suits their needs. In the case study, the team made use of tools like HipChat and Link to send quick messages and share information. It was also important for them to keep records of what team members have been saying and these tools allowed for this.

"So, there is obviously this thing called HipChat. A plugin that you can integrate with Atlassian tools to allow you to chat to someone. Link as well to chat and send quick messages. So, we do use nice communication tools to share information and also have that paper trail of what people have been saying. You can as a developer look back and figure for yourself. Instead of just trying to master communication" (A1).

The ***Kanban Board*** also helped by reducing the need for synchronous communication by making readily available whatever information on the project that might be needed urgently. Respondents felt that without the Kanban board, there would be a lot of follow up and phone calls to determine what the item is about, when and who is working on it, and what the priority of that item is. The kind of information that might be relevant can relate to who put the issue on the backlog or the board and what is required for that item. Having a visible Kanban board and having team members regularly contributing to the information that was on the board therefore helped the lack of synchronous communication challenge.

"The Kanban board is visible to everyone. When you have an issue you can always write down on the Kanban board and at the time we congregate again, the details are already there" (A2)

"The communication would probably be harder without Kanban. Most of the issues I have been working on were populated a long time ago and they have already been pre-defined, everything was described properly. So all you do is just pick it up. The fact that everything was well-defined, it kind of limited the back and forth communication between who raised the issue and who put the issue on the backlog. It was much easier to just work on something and get it done. The minute you have a problem you can easily link that person or call them. You already see how everything fits together because the board is continuous. From start to finish, I would say" (A3)

"Absolutely because projects that don't have the visibility of Kanban, there is a lot of following up. Things are raised in the morning, there is a lot of calling to understand exactly what happened, when, who is working on it, what has priority in the day. Outside of that, there's a lot of being on the phone chatting to the developer, sending emails to get status updates and that ends up being very distractive. Whereas with Kanban the developer is in charge of updating each point and it's up in a global location. So, after they completed the task, they update it and the put it on the JIRA Kanban board, and all I had to do is log in and check the status" (A4)

Lack of face-to-face meetings

Given the geographical distance between team members, it is difficult to organize face-to-face meetings in GSD. And yet, these face-to-face meetings are important during software development projects. In GSD, the most common way of conducting meetings is via video conference or conference calls. In the case study, respondents felt that Kanban allowed for fewer meetings to be conducted because of the fact that important information was already readily available on the ***Kanban Board***, allowing for shorter and more focused meetings to be held. The need for meetings was also reduced because relevant information for decision making was also readily available on the ***Kanban Board***.

"What you find is that now because of the right tools and the Kanban approach, your meetings are shorter but more focused on what actually makes sense and what actually adds value." (A1)

"The nice thing about Kanban and asynchronous communication is that kit allows people to make the decisions there. It eliminates the meetings." (A1)

However, although Kanban decreases the time needed to meet, meetings are still necessary to ensure a common understanding of the requirements as explained by A2 and A5:

"Kanban decreases the need for face to face meetings but face to face is something that is supplementary" (A2)

"I would say that Kanban decreases the need for face to face meetings but Kanban supplements the face to face meetings. Depending on whoever the parties are because it's subjective. Understanding is subjective. People comprehend things differently" (A5)

Time difference

In GSD, time difference can be a challenge when communication needs to take place at unconventional times due to the lack of overlapping working hours and leads to overtime work. Although there was only a maximum of 2-hour difference between London and Cape Town, some respondents regarded time difference as being somewhat an issue in the case study. Those respondents felt that the ***Kanban Board*** helped alleviate the time difference problem because even though the members might be working at different times, it is always possible to have access to any updates on the project through the board. Any updates are clearly visible on the Kanban board. In addition to that, having the team collaborating and proactively updating the Kanban board allows every team member to know what the exact state of the project is.

"Because all the other users can do is pretty much reading what the other participants have created and because it's always available and online regardless of your location or time" (A2)

"It does to a very large extent because if used properly people adhere to populating issues adequately I think it can be a powerful tool. All the other users can do is pretty much reading what the other participants have created and because it's always available and online regardless of your location or time. You can always access it and get the notifications regardless of time zone. I think it does help" (A5)

SUMMARY OF THE FINDINGS

The study reveals that several Kanban elements can be used to alleviate communication and collaboration challenges as they occur in GSD. In this study, not all communication and collaboration reported in literature were observed in the case study. In addition, not all Kanban elements reported in the literature were employed by the case organisation. The specific Kanban tools and how they alleviate communication and collaboration challenges are summarised below.

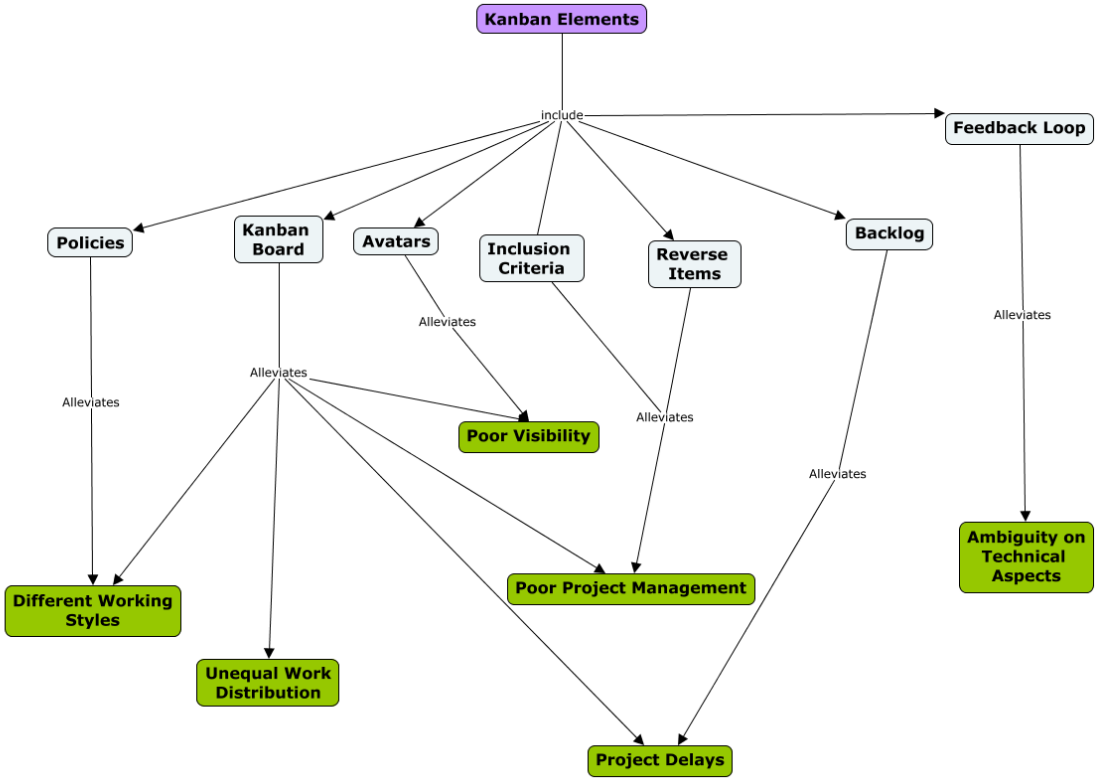

Figure 2. Kanban Elements & Collaboration challenges

As previously mentioned, there are different types of Kanban elements and this study indicates that some of them, when employed might be useful in alleviating certain types of collaboration challenges. As can be seen in Figure 2:

- When ***policies*** to regularly log the time spent on items are put in place, collaboration challenges emerging from different working styles in the GSD context are alleviated
- The use of the ***Kanban board*** in the GSD context helps alleviate collaboration challenges related to different working styles, unequal work distribution, project delays, poor project management, and poor visibility
- The use of ***Avatars*** helps alleviate visibility issues and promote transparency on who is doing what in the GSD team
- The use of ***Inclusion Criteria*** and ***Reverse Items*** help alleviate project management related issues, especially when requirements change and new requirements emerge in the middle of an iteration in GSD
- The use of up to date and easily accessible ***Backlogs*** alleviates the occurrence of project delays especially when team members are working on multiple projects simultaneously in GSD
- The use of ***Feedback Loops*** alleviates collaboration challenges related to the ambiguity on technical aspects of a requirements

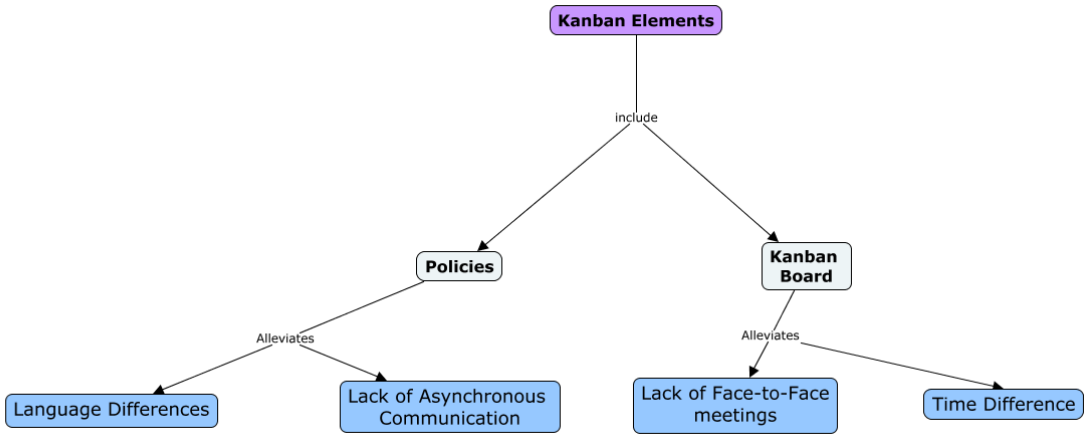

Figure 3. Kanban Elements & Communication Challenges

Only two Kanban elements were found useful in alleviating certain types of communication challenges. As can be seen in Figure 3:

- When ***policies*** are implemented that promote the use of communication tools like Link and HipChat, communication challenges related to the lack of asynchronous communication can be alleviated
- When ***policies*** are implemented that promote the need to write down descriptions of items and issues on the Kanban Board, communication challenges related to language differences and accents can be alleviated
- The use of the ***Kanban Board*** can be used to alleviate communication challenges related to the lack of face-to-face meetings and time difference

CONCLUSION

Cultural, geographical and temporal distances create a number of communication and collaboration challenges in GSD teams. These include a lack of synchronous communication and face-to-face meetings, as well as difficulty in distributing work equally and creating mutual understanding in the teams. The study employed a qualitative case study to inductively determine how Kanban elements can alleviate these challenges in the GSD context. Data was analysed using the general inductive approach from Thomas (2006).

The study found that some Kanban elements, when properly employed, can help alleviate some collaboration and communication challenges that occur within GSD teams. These relate to Inclusion Criteria, Reverse Items, Kanban Board, Policies, Avatars, and Backlog. Figures 2 and 3 summarise how each Kanban element employed by the GSD team which was studied helped alleviate specific communication and collaboration challenges. Not all Kanban elements are relevant to all communication and collaboration challenges. The study found that given their nature and how they are employed, some elements are more suited to some specific forms of challenges. Therefore when employed in a coordinated and consolidated manner, these Kanban elements can be useful in the GSD context.

It must be noted that the case organisation did not employ all Kanban elements, and this might be considered as a limitation of the study. Further research is therefore required, with a bigger sample that uses all the Kanban elements to better understand how Kanban can be used in GSD. Indeed, there is a still a great deal to be researched on the topic of Kanban and its impact on GSD projects.

There is a lack of detailed guidelines as well as a lack of awareness on the Kanban elements. The following are some suggestions for possible future research projects.

- An extended study following a similar approach to this one but with a bigger sample that uses more Kanban elements.
- A similar study with teams located in countries with different first languages and bigger time differences.

This study makes an important contribution to theory and practice as there is very few studies that investigated how Kanban can be used to alleviate collaboration and communication challenges of GSD teams. The contribution of this paper is a detailed description on how some Kanban elements have been successfully applied to a dispersed software development environment. These experiences are useful for organisations planning to use Kanban in a similar setting. In particular, GSD and Lean Practitioners might benefit from this study, especially if they are looking for ways of mitigating communication and collaboration challenges through a novel software development method like Kanban. Since specific insight has been provided on the relevance of some Kanban elements in relation to specific communication and collaboration challenges, practitioners might adapt their practices to better reap the benefit that such elements can provide in GSD.

Future research can investigate the same research questions (or similar ones) using a quantitative approach. This would allow for a broader sample to be reached and the results could then be compared with what has been found in this qualitative study.

REFERENCES

Abdulmalek, F. A., & Rajgopal, J. (2007). Analyzing the benefits of lean manufacturing and value stream mapping via simulation: A process sector case study. *International Journal of Production Economics*, *107*(1), 223–236. http://doi.org/10.1016/j.ijpe.2006.09.009

Agee, J. (2009). Developing qualitative research questions: a reflective process. *International Journal of Qualitative Studies in Education*, *22*(4), 431–447. http://doi.org/10.1080/09518390902736512

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review (pp. 9–16). IEEE. http://doi.org/10.1109/SEAA.2013.28

Ahmad, M. O., Markkula, J., Oivo, M., & Kuvaja, P. (2014). Usage of Kanban in Software Companies. *In 9th International Conference on Software Engineering Advances*. Retrieved from https://www.researchgate.net/profile/Muhammad_Ovais_Ahmad/publication/267515017_Usage_of_Kanban_in_Software_Companies_An_empirical_study_on_motivation_benefits_and_challenges/links/54522f080cf285a067c72291.pdf

Al-Baik, O., & Miller, J. (2014). The kanban approach, between agility and leanness: a systematic review. *Empirical Software Engineering*, *20*(6), 1861–1897. http://doi.org/10.1007/s10664-014-9340-x

Anh, N.-D., Cruzes, D. S., & Conradi, R. (2012). Dispersion, coordination and performance in global software teams: a systematic review. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement* (pp. 129–138). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2372274

Bannerman, P. L., Hossain, E., & Jeffery, R. (2012). Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage? (pp. 5309–5318). IEEE. http://doi.org/10.1109/HICSS.2012.512

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS quarterly*, 369-386.

Bhattacherjee, A. (2012). *Social science research: principles, methods, and practices*.

Bjarnason, E., Wnuk, K., & Regnell, B. (2011). Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development. In *Requirements Engineering Conference (RE), 2011 19th IEEE International* (pp. 37–46). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6051639

Chai, L. L. (2008, September). E-based inter-enterprise supply chain Kanban for demand and order fulfilment management. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on* (pp. 33-35). IEEE.

Colomo-Palacios, R., Soto-Acosta, P., García-Peñalvo, F. J., & García-Crespo, Á. (2012). A study of the impact of global software development in packaged software release planning. *Journal of Universal Computer Science*, *18*(19), 2646–2668.

Costa, P. L., Passos, A. M., & Bakker, A. B. (2014). Team work engagement: A model of emergence. *Journal of Occupational and Organizational Psychology*, *87*(2), 414–436. http://doi.org/10.1111/joop.12057

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), 1213–1221. http://doi.org/10.1016/j.jss.2012.02.033

Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, *50*(9–10), 833–859. http://doi.org/10.1016/j.infsof.2008.01.006

Ebert, C., Abrahamsson, P., & Oza, N. (2012). Lean software development. *IEEE Software*, (5), 22–25.

Ericsson, R., & Granlöf, A. (2011). The effects of Kanban in software development teams: a study of the implementation at Sandvik. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2:519094

Filho, A. F. G., & Toledo, R. de. (2015). Visual Management and Blind Software Developers (pp. 31–39). IEEE. http://doi.org/10.1109/Agile.2015.14

Gaury, E. G. A., Pierreval, H., & Kleijnen, J. P. (2000). An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid. *Journal of Intelligent Manufacturing*, *11*(2), 157-167.

Hanssen, G. K., Šmite, D., & Moe, N. B. (2011). Signs of Agile Trends in Global Software Engineering Research: A Tertiary Study (pp. 17–23). IEEE. http://doi.org/10.1109/ICGSE-W.2011.12

Hashmi, S. I., Clerc, V., Razavian, M., Manteli, C., Tamburri, D. A., Lago, P., … Richardson, I. (2011). Using the Cloud to Facilitate Global Software Development Challenges (pp. 70–77). IEEE. http://doi.org/10.1109/ICGSE-W.2011.19

Herbsleb, J. D. (2007). Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering* (pp. 188–198). IEEE Computer Society. Retrieved from http://dl.acm.org/citation.cfm?id=1254718

Hsu, J. S.-C., Shih, S.-P., Chiang, J. C., & Liu, J. Y.-C. (2012). The impact of transactive memory systems on IS development teams' coordination, communication, and performance. *International Journal of Project Management*, *30*(3), 329–340. http://doi.org/10.1016/j.ijproman.2011.08.003

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation (pp. 305–314). IEEE. http://doi.org/10.1109/ICECCS.2011.37

Jailia, E. K., Sujata M., Jailia M. M., & Agarwal, M.M. (2011). Lean software development ("as a survival tool in recession"). *International Journal of Software Engineering and its Applications*, *5*(3), 63-70.

Jarvenpaa, S., & Leidner, D. E. (1998). Communication and trust in global virtual teams. Journal of Computer Mediated Communication, 3(4).

Kaur, P. & Sharma, S. (2014). Agile software development in global software engineering, *International Journal of Computer Applications*, 97(4)

Khan, Z. (2014). *Scrumban-Adaptive agile development process: Using scrumban to improve software development process.* (Master's thesis.) Helsinki Metropolia University of Applied Sciences. Retrieved from https://publications.theseus.fi/bitstream/handle/10024/77014/Khan_Zahoor.pdf?sequence=1

Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum: making the most of both*. s. l.: C4Media.

Kniberg, H. (2009). *Kanban vs. Scrum*, available at https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf

Korkala, M., & Maurer, F. (2014). Waste identification as the means for improving communication in globally distributed agile software development. *Journal of Systems and Software*, *95*, 122–140. http://doi.org/10.1016/j.jss.2014.03.080

Layman, L., Williams, L., Damian, D., & Bures, H. (2006). Essential communication practices for Extreme Programming in a global software development team. *Information and Software Technology*, *48*(9), 781–794. http://doi.org/10.1016/j.infsof.2006.01.004

Li, P. (2016, November). *Running your project using Kanban*. Retrieved May 04, 2017, from https://www.packtpub.com/books/content/running-your-project-using-kanban

Mahnic, V. (2014). Improving Software Development through Combination of Scrum and Kanban. *Recent Advances in Computer Engineering, Communications and Information Technology, Espanha*. Retrieved from http://www.wseas.us/e-library/conferences/2014/Tenerife/INFORM/INFORM-40.pdf

Marshall, M. N. (1996). Sampling for qualitative research. *Family Practice*, *13*(6), 522–526.

Middleton, P., & Joyce, D. (2012). Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management*, *59*(1), 20–32. http://doi.org/10.1109/TEM.2010.2081675

Mohagheghi, P. (2004). Global software development: Issues, solutions, and challenges, Retrieved April 07, 2009 from http://www.idi.ntnu.no/grupper/su/publ/parastoo/gsd-presentation-slides.pdf

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, *17*(1), 2–26. http://doi.org/10.1016/j.infoandorg.2006.11.001

Niazi, M., Mahmood, S., Alshayeb, M., Rehan Riaz, M., Faisal, K., & Cerpa, N. (2013). Challenges of project management in Global Software Development: Initial results. In *Science and Information Conference (SAI), 2013* (pp. 202–206). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6661739

Niinimäki, T., Piri, A., Lassenius, C., & Paasivaara, M. (2012). Reflecting the choice and usage of communication tools in global software development projects with media synchronicity theory: CHOICE AND USAGE OF COMMUNICATION TOOLS IN GSD PROJECTS. *Journal of Software: Evolution and Process*, *24*(6), 677–692. http://doi.org/10.1002/smr.566

Oza, N., Fagerholm, F., & Munch, J. (2013). How does Kanban impact communication and collaboration in software engineering teams? In *Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on* (pp. 125–128). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6614747

Petersen, K., & Wohlin, C. (2011). Measuring the flow in lean software development. *Software: Practice and Experience*, *41*(9), 975–996. http://doi.org/10.1002/spe.975

Polk, R. (2011, August). Agile and Kanban in coordination. In *Agile Conference (AGILE), 2011* (pp. 263-268). IEEE.

Portillo-Rodríguez, J., Vizcaíno, A., Piattini, M., & Beecham, S. (2012). Tools used in Global Software Engineering: A systematic mapping review. *Information and Software Technology*, *54*(7), 663–685. http://doi.org/10.1016/j.infsof.2012.02.006

Reis, E. (2011). *The lean startup.* New York: Crown Business.

Richardson, I., Casey, V., McCaffery, F., Burton, J., & Beecham, S. (2012). A Process Framework for Global Software Engineering Teams. *Information and Software Technology*, *54*(11), 1175–1191. http://doi.org/10.1016/j.infsof.2012.05.002

Scharff, C. (2011). Guiding global software development projects using Scrum and Agile with quality assurance. In *Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on* (pp. 274–283). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5876097

Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. *Information and Organization*, *21*(1), 1–16. http://doi.org/10.1016/j.infoandorg.2010.11.001

Sudhakar, G., P., Farooq, A., & Patnaik, S. (2011). Soft factors affecting the performance of software development teams. *Team Performance Management: An International Journal*, *17*(3/4), 187–205. http://doi.org/10.1108/13527591111143718

Tanner, M. (2009). Communication and Culture in Global Software Development: The Case of Mauritius and South Africa, *Journal of Information, Information Technology, and Organizations, 4*.

Thomas, D. R. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data.

Turner, R., Madachy, R., Ingold, D., & Lane, J. A. (2012). Modeling kanban processes in systems engineering. In *Proceedings of the International Conference on Software and System Process* (pp. 23–27). IEEE Press. Retrieved from http://dl.acm.org/citation.cfm?id=2664365

Søderberg, A.-M., Krishna, S., & Bjørn, P. (2013). Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships. *Journal of International Management*, *19*(4), 347–361. http://doi.org/10.1016/j.intman.2013.04.004

Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, *85*(6), 1287–1299. http://doi.org/10.1016/j.jss.2012.01.061

Yin, R. K. (2013). *Case study research: Design and methods*. Sage Publications.

BIOGRAPHIES

A/Prof Maureen Tanner has been teaching systems analysis and design at the Department of Information Systems of the University of Cape Town since 2009. Her research interests lie in Social Media, Agile software development related issues (for both collocated and distributed teams), UML, software engineering and social aspects of social engineering, global software development, virtual teams, and team collaboration.

Marcelo Dauane completed a BSc in Computer Science and a BComm (Honors) in Information Systems at the University of Cape Town. He currently works as a Mobile Software Developer in the Fintech industry. His research interests evolve around Blockchain, Distributed Ledgers, Digital assets, Global Software Development, Agile and Lean Software Development.