Journal of Cases on Information Technology
Volume 23 • Issue 4 • October-December 2021

Translation of Activities in a Global Virtual Teams Software Development: Agile vs. Waterfall

Olayele Adelakun, DePaul University, USA
Tiko Iyamu, Cape Peninsula University of Technology, South Africa
https://orcid.org/0000-0002-4949-094X

ABSTRACT

The waterfall software development approach and traditional project management have been the dominant approaches used for software development for many years among collocated teams and global teams alike. In recent years, agile development approaches such as Scrum, XP, Kanban, and Scrumban have increasingly gained popularity in practice, and they are the dominant software development approach among start-ups and Silicon Valley companies. While waterfall is still being used among very large traditional companies, there is a gradual move toward agile by many of these companies. This is often referred to as agile transformation. Global software teams have to constantly struggle with the appropriate software development approach for global projects. More often than not, the choice is left to the teams, and sometimes the dominant software development practice in the organizations will be adopted by the global teams. This study explores global virtual software team development practices and tries to demystify some of the misconceptions about global software development practices based on findings from the global virtual software team experiment that was carried out at DePaul University from 2011-2018. The moments of translation from the perspective of actor-network theory (ANT) was employed in the data analysis to examine how the development approach was selected by the global virtual teams. One of the key findings from the research is that the success of a global software development project does not have a strong dependency on the development approach. While one agrees that it is one of the key influencing factors, there are other equally strong factors for global virtual software team success.

KEYWORDS

Actor-Network Theory, Agile Methodology, Global Virtual Teams, Software Development, Translation, Waterfall Methodology

1. INTRODUCTION

Companies employ project management approach in the development and implementation of software in their environments. In recent years, companies change from the traditional waterfall to what is considered a more dynamic approach, agile practices for various reasons. Some of the key benefits include flexibility to make changes to project requirements, less documentation, earlier defect detection, and customers' feedback (Laanti, Salo, & Abrahamsson, 2011). In addition, agile practices seem to provide more transparency than waterfall approach, which assist some organizations

DOI: 10.4018/JCIT.20211001.oa11

to validate their requirements. Perhaps the most important benefit of agile practice is the ability to get a working software product or features to market faster than waterfall approach.

Despite the numerous benefits and advantages of agile approach for software development, its adoption has been challenging (Boehm & Turner, 2005; Nerur, Mahapatra, & Mangalaraj, 2005). The challenge gets worse when the approach had to be adopted at an enterprise level. In recents years, researchers note that organizations that have stable requirements can develop successful projects by using waterfall (McHugh, McCaffery, & Casey, 2012). In attempts to find a more suitable approach, some organizations have resulted to hybrid agile, such as Scrum-Fall, a combination of waterfall and Scrum development methods (Binder, Aillaud, & Schilli, 2014; Rahmanian, 2014; Adelakun, Garcia, Tabaka, & Ismail, 2017).

Agile scrum stipulates that the scrum team should be co-located in order to increase the level of interaction and collaboration among team members. This is well documented in agile guide at scrum.org. Despite the documentation, it remains a challenge for global teams. Also, scrum stipulated that agile scrum teams should meet on daily bases for about 10 or15 minutes for a ceremony called daily scrum. The daily scrum ceremony is a major challenge for virtual teams that are geographically separated across the globe. The major challenges are differences in time zone and language. This makes some argue that scrum should be used only for local teams, and waterfall for global teams.

This paper shows evidence that both software teams separated by geographical locations and those in the same location can choose either agile or traditional (waterfall), to successfully develop a software application. This paper reports an experimental research involving global virtual teams across seven countries in four continents, North America, South America, Europe, and Africa. The experiment was conducted from 2011 to 2018. The data for this paper was collected during the 7 years of the experiment.

The actor-network theory (ANT) was selected to underpin this study. This means that the theory is used as a lens in analysing and gaining an understanding of human activities in the selection and use of agile or waterfall method, for software development. The theory was selected primarily because the theory is concerned about actors' interaction, relationship, and how networks (teams) are created including the activities. The main tenets of ANT are actor, network, and translation. Both human and nonhuman are regarded as actors, and a network is an alliance, formation of common interest by actors (Callon, 1986). From the perspective of ANT, the moments of translation provide a conceptual framework for analysing and understanding human activity. Increasingly, theories are used as lenses in IS studies through which researchers perform analysis of data and interpretation of findings. In Walsham's (2006) explanation, a sociotechnical theory can be used as an initial guide to a study design and as part of an iterative process in the collection and analysis of data.

The paper is structure into seven main sections, it began with introduction, followed by problematisation and motivation for the study. In the third section, an overview of managing traditional and Agile software development is presented. The fourth section presents a discussion about the theory, ANT that underpins the study. The methodology employed in this study, and the findings from analysis are discussed in sections five and six, respectively. The conclusion is drawn in the final section.

2. PROBLEMATISATION AND MOTIVATION

Agile and waterfall are two different methodologies, however, there are similarities such as common goal, to deliver quality software (Kusters et al., 2017). The differences and similarities can range from approach to skill and culture. Andrei et al. (2019) identify and argue two characteristics of similarities between the Agile and Waterfall methods. The existence of this area of concern is readily shown, both in theory and in practice. However, no current overview of the issues that play a role could be identified in literature. The waterfall is static whereas Agile is dynamic (Kettunen & Lejeune, 2020).

Is that enough to select or adopt one over the other? Anecdotal evidence seems to shift preference towards Agile methodology, but what does the empirical evidence says?

Each individual or team including an organisation often have preference between the methodologies. According to Andrei et al. (2019), Waterfall is clearly different from two angles, highly structured, and it does not allow disruption by emerging requirements during project. Almeida and Simões (2019) implicitly positioned an argument on how difficult it can be for organisations to select one methodology over the other, and that there were only a few case studies as at the time of their study. Also, little is known about the extent to which any of the methodologies truly influences software design and its outcomes (Thummadi & Lyytinen, 2020). A question that that came out of Heeager and Nielsen (2018) that doesn't seems to have been empirically answered is whether certain software are better developed using a specific methodology rather than the other.

Literature has tries to address the differences between Agile and Waterfall methodologies but from cost, quality, and productivity perspectives (Thummadi & Lyytinen, 2020). Even though there are few areas of agreement among authors, the debate continues. This could be attributed to the fact that the authors or researches viewed the subject from different perspectives and objectives. Irrespective of the underpinning reasoning or goal, the subject remain plausible for further research and such pursuit is supported by literature. Sandberg and Alvesson (2011) explain in their argument how and why gap-spotting is the best and most common way of constructing of problematising research. Thus, we formulated the research question: what are the factors that influence and determine whether both global software teams and co-located can choose either agile or traditional (waterfall) to successfully develop a software application?

3. MANAGING TRADITIONAL AND AGILE SOFTWARE DEVELOPMENT

Software development project either agile or traditional, at its core, is about delivering successful project or quality software product on time, on budget, and according to stakeholders' expectation (Seymour & Hussein, 2014). Key stakeholders are the product owners in scrum agile, and customers in traditional approach. Managing waterfall project is generally referred to as traditional. It is a linear process with heavy upfront planning of resources. Traditional project management therefore emphasis proper allocation of task and tight control of project resources (Sixsmith, Freeburn, & Mooney, 2014; Fernandez & Fernandez, 2008; Rehman, Rauf, & Shahid, 2010). A good documentation of project activities is considered an important measure of project progress and thereby successful execution.

From success perspective, Khoza and Marnewick (2020) explored whether there is a difference between Agile and Waterfall in software development projects. Tam et al. (2020) argues that "team capability" and "customer involvement" are the main factors to the selection of the Agile methodology and was silent on the part of Waterfall. Also, different meanings and connotation are associated with the same thing, which increases the confusion for team members who are set-out to select either of the methodology. Consequently, the combination of Agile and Waterfall in a project can only be validated through empirical studies and practice (Gemino, Horner Reich & Serrador, 2021).

One of the major problems with traditional projects is that conforming to plan is extremely difficult because of rapid changes that happen in business environments (Highsmith & Cockburn, 2001). It is actually expected that the project requirements will change before the end of the project. Therefore, project managers try to reduce changes to traditional projects. Changes to project increases both time and cost, which are two of the key measures of project success. Agile presents a potential solution to some of the challenges raised by traditional project management (Abrahamsson, Salo, Ronkainen & Warsta, 2017).

Agile is a broad umbrella that consist of many practices such as Scrum, Kanban, Extreme programming (XP) and a number of hybrid approaches (Adelakun, Garcia, Tabaka, & Ismail, 2017). All agile practices have similar principles such that they favour rapid incremental development and collaboration among team members. Agile scrum especially, request that team members should be

Figure 1. Source (Agile Manifesto at agilemenifesto.org)

self-organized and cross functional (Lindvall et al., 2002). One of the most important benefits of agile is the ability to accept changes to software development requirements at any time during the development process, even at the very last minute before deployment. This is a radical departure from traditional approach, which makes it difficult for those with extensive experience with traditional project management and waterfall to change to agile. Hence the need for agile transformation in many traditional and hierarchal organizations.

Figure 1 shows the difference between agile and traditional (Stoica, Mircea & Ghilic-Micu, 2013; Agilemanifesto.org). Figure 1 is based on agile manifesto that highlighted that while items on the right side of the figure (e.g. a focus on process and tools) are important to project success, agile team prefers items on the left side better (e.g. a focus on individual and interactions). Adopters believe that items on the left side help agile teams to deliver a more relevant and useful product, on time. Such products are also believed fulfil customers' satisfaction at a high level rather than traditional projects.

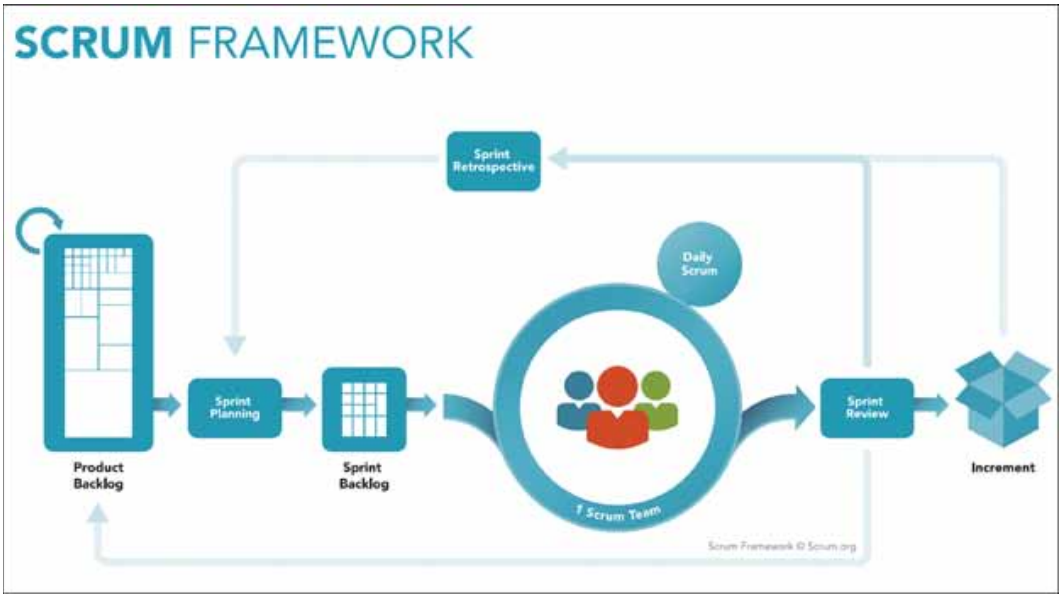
Out of many agile practice (Scrum, Kanban, eXtreme Programming, Scrumban etc.) the most popular and most adopted is Scrum (Moniruzzaman & Hossain, 2013; Schwaber & Beedle, 2002). Scrum is perhaps the most adopted because it is relatively easy to implement, and easy to combine with waterfall, to form Scrum-fall. Like other agile methods, scrum works well in situation where requirements change constantly, while deliverable time is fixed (Augustine, Payne, Sencindiver, & Woodcock, 2005). Figure 2 shows the scrum framework copied from Scrum.org.

Scrum has clearly defined roles such as product owner, scrum master, and product team. Also, there are clearly define responsibilities for each role. In addition, Scrum stipulate process ceremonies, which include daily scrum, sprint review, and sprint retrospective. Some have argued that the scrum master replaces the traditional project manager. While some of the responsibilities of the scrum master and project manager may overlap, overall, the two roles have completely difference responsibilities.

4. ACTOR-NETWORK THEORY

Actor-network theory (ANT) was employed to underpin the study. Some justification for selecting ANT for this study are its unique strengths and concepts of heterogeneity, translations and relationship between human and non-human including classification of actors' actions and activities. These factors are critical factors in examining and understanding virtual software development teams that are in different locations across the world.

Figure 2. Source (Scrum.org)



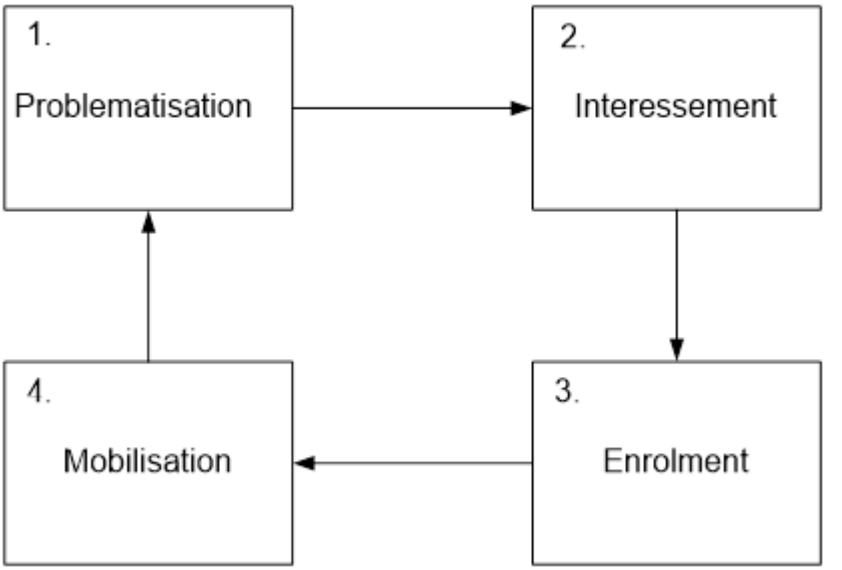
Another rational for selecting ANT is its strength in examining associations that exist, to reflect on how a network becomes durable (Latour, 1999). Bilodeau and Potvin (2018) argue translation is employed to resolve controversies and disparate situations between actors through their associations, as they influence each other to gain power. Through translation of the activities, knowledge and viewpoints were strengthened (Bilodeau & Potvin, 2018), which assisted in stabilising the teams in their locations. From technology perspective, this could be attributed to the fact that ANT explains proliferation of discrete technological objects (Sage, Vitry & Dainty, 2020).

The concept of heterogeneity from ANT perspective is used to study networks (Islam, Mäntymäki & Turunen, 2019), which we explored and leveraged to examine relationship between teams (networks) that are in different locations. This helps us to create linkages among the entities in the various networks, which is one of ANT's strengths (Markus & Rowe, 2018). ANT's unique ways of describing clarifying processes that occurs when actors form connections and ultimately create links help to understand why things happen in certain manners (Fenwick & Edwards, 2019). Through such links, the use of technologies are shaped within a social context (Cresswell, 2019).

From ANT perspective, the moment of translation is a process where activities are transformed or given a new meaning through negotiation. The moments of translation consist of four stages as shown in Figure 3: problematisation, interessement, enrolment, and mobilisation (Callon, 1986). The first stage, problematisation is where the focal actor identifies a problem that needs to be addressed ((Nehemia-Maletzky, Iyamu & Shaanika, 2018)). This is followed by interessement stage in which actors declare interest in the problematised activity (Thapa, 2011). Enrolment is the third stage, where some interested actors actually participate in the activities that the focal actor allocated to them (Heeks & Stanforth, 2015). Finally, mobilisation takes place, a stage where appointed or volunteered spokesperson ensures that actors carry out their roles and responsibilities in order to achieve the objectives of the network (Callon, 1986).

In ANT, actors and network are inseparable, which helps to solidify negotiation between them through the four moments of translation (Iyamu & Mgudlwa, 2018). The main purpose of having a network is to allow actors from various areas to collaborate in order to solve a problem or create something new, such as the development, implementation, and maintenance of software. Iyamu and

Figure 3. Moments of translation (Callon, 1986)



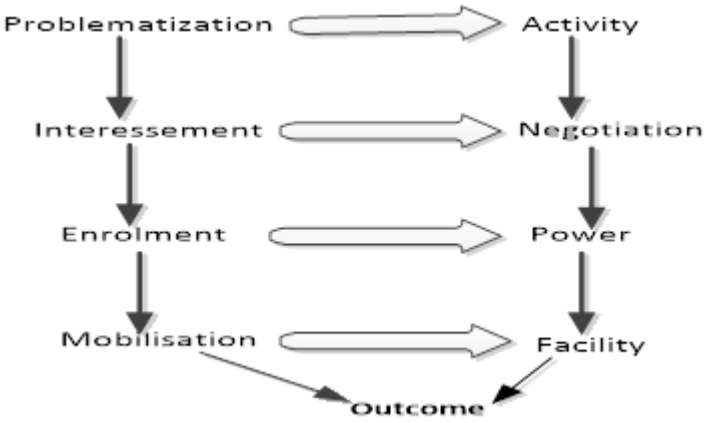
Roode (2010) argued that a network is a formation of a group of actors who work together towards a common interest. Networks consist of both human and nonhuman actors.

The theory is applied in this study primarily because virtual software development teams consist of sociotechnical realities, in which human actors make use of nonhuman actors (software, agile, waterfall) through problematisation, based on interest and participation, and this consequentially made known through mobilisation. The social activities such as interaction and culture which humans employ as they develop, implement, maintain and interact with the systems are complex in nature. Mkhomazi and Iyamu (2013) suggest that the social nature is more than technical challenges in seeking IT solutions. According to Callon (1986), ANT is a social technical theory that focuses on actors' negotiation and interaction within networks. This makes the use of ANT to underpin this study most appropriate. The theory was therefore used as a lens to gain an in-depth understanding of how networks (virtual teams) were formed, and how the interaction and relationship that happens between actors manifest in the selection and use of the agile or waterfall method in the development of software. This include the negotiation that happens in the process, and how the outcome was often translated.

Figure 4 is the conceptual model that we developed purposely, to follow the actors, and help gain better understanding during the analysis and interpretation of the data and activities that come from our experiment. In ANT, problem is not something that is broken, but initiation of an activity that require a solution, or different approach.

An activity is problematised, which influences the interest (interessement) of different actors and draws on negotiate scheme (negotiation). Negotiation results to actors' participation (enrolment) in an activity. Individually, actors consciously or unconsciously possess power, which they employ during negotiation. This defines actors' interaction and relationship, and enact the use of facilities, such as skills to deter actions. According to Callon (1986), actors' interaction creates associations, and defines one another through the intermediaries that pass between them. The selection and use of agile or waterfall was initiated, decisions were made. In the process, relationship and interaction happen between humans, and with nonhuman actors within teams during the development of software. The interested actors including those who participated in an activity do so either voluntarily or based on contractual agreement enacted by power bestowed on authority or position (Iyamu & Roode, 2010). Available facilities are used to facilitate an outcome from each team.

Figure 4. Conceptual model



Power in itself is not a thing or an act, but rather a facility (or resource) that is used to make a difference. for example, how power and values are embedded in the use of technology. Law (1992) suggests that how size, or power is generated should be studied in an unprejudiced manner. As shown in Figure 4, ANT can be used in an iterative process. According to Heeks and Stanforth (2015), through the four moments of translation, the theory can be seen as a process that is recursive in practice.

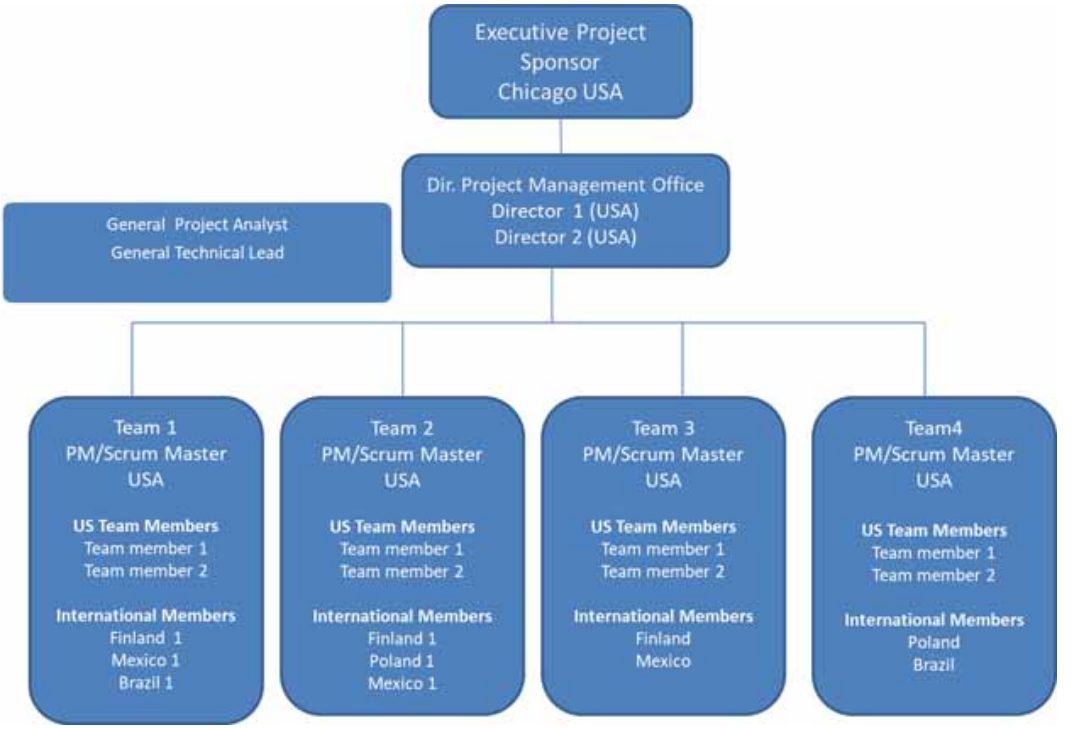
5. RESEARCH METHOD

In 2011 a group of researchers from DePaul University launched the global virtual software team project. The research project started with five teams. Each team had a project manager from DePaul University, that reports to the office of the project manager, as shown in Figure 5. All the team members were students, at graduate and undergraduate levels. Only students at graduate level (PhD or Master) were qualified to occupy the position of a director, and/or a project manager in DePaul University, Chicago. In addition, for a student to become a project manager or director of the project or program office, they have to meet some academic requirements, such as successful completion of core project management courses at their university or at DePaul University. Each team is designed to have two software developers from Chicago, and two to three development team members from one of the following countries, Brazil, Finland, Poland, Ghana, and Mexico. See Figure 5, for the structure of the global team that was constituted from 2011-2014.

Although Figure show 5 countries, there were three participating members (universities) in this aspect of the project, from which this paper is produced. These are Germany, USA, and South Africa. A total of 17 people: software developers (x13), scrum masters (x2), and project manager (x2) were involved in the project. Software developers: Germany (x4), USA (x7), and South Africa (x2); scrum masters: Germany (x1); USA (1); and the 2 project managers were from USA. The universities selected participants using the same criterial. A minimum of 2 and 1 year experience of software development and methodology (either agile or waterfall), respectively. These were the basic criteria for software developers. In selecting scrum masters and project managers, it was the same criteria for software developer, in addition, a minimum of 3 years' experience of project management.

In 2011, the development approach was waterfall and traditional software project development approaches. Each team has a clearly defined requirements in the beginning of the project. Core project documentations were maintained by each team. At the end of the ten weeks' project in Spring of 2011 to 2014, each team presents the final outcome of their projects. Each project was not code heavy and

Figure 5. DePaul University's Global Virtual Software Team Structure in 2011.



each team has the choice of using web-authoring tools like Joomla, Word Press, or any other tool that the team was familiar with for a web-based project.

From 2011 to 2014, all project managers were from the DePaul University, Chicago, USA and the projects also were predetermined by the project sponsors in DePaul University. By 2016, more countries namely Germany and South Africa were added, and the project structure changed to agile. As a result, each team no longer has a project manager. As the development approach changed to Scrum, each team had a scrum master, a product owner, and development project team. Product owners still resides in Chicago, but scrum masters reside in any location. The global virtual teams were required to follow all scrum ceremonies using collaborative tools like Skype, Zoom or any other tool they felt comfortable with.

Each team from 2011 to 2018 experienced the same type of challenges, namely, time zone, distance, cultural, and language differences. See Iyamu, Nehemia-Maletzky & Shaanika (2016) on global software teams' challenges. The virtual global teams between 2011 to 2014 focused on web-based applications primarily. Some of the web-based application developed then include patients vital record management system and others. From 2016, the global teams developed both web based and mobile applications. The focus of this paper in not about the type of application developed nor the complication of the application but the final outcome, deliverable of the global teams using different development approaches. Data were collected through observation in an action research setting. The lead author of this paper was the director of the program since 2011, and the co-author has been involved with this project since 2016.

6. ANALYSIS OF THE DATA

In achieving the objective of this study, which was to determine whether both global software teams and co-located can choose either agile or traditional (waterfall) to successfully develop a software application, the moments of translation was employed. We therefore use the four stages of translation to examine and gain an understanding of the interaction that happens among team members (software developers, scrum masters, and project managers). This includes how tasks were allocated. Primarily, a network is aimed at allowing human actors to collaborate using the available tools (nonhuman actors) in solving or finding solution to problems (Nehemia-Maletzky et al., 2018). According to Latour (2005), ANT addresses the elements of science and technology in the making as opposed to ready-made science and technology. ANT's approach is thus to unpack and analyse the elementary components such as actors, processes, environment, policies and networks, which are often involved in the development of IT solutions (Nehemia-Maletzky, Iyamu & Shaanika, 2018), such as the selection and use of agile or waterfall method.

6.1. Problematisation

The question as problematised by the focal actor was, can both agile and waterfall be used as methods in the development of software by virtual teams? The project manager problematised the question. Members of the teams including the scrum masters agreed that the question was worth an answer for collaboration purposes. Virtual teams exist and will continue to do so. Issues were problematised considering the different culture, locations and time zones, languages, levels of expertise and aligned with responsibilities (Islam et al., 2019). Also, teams' members require methods, agile or waterfall to guide their software development. These were realties that were problematised in creating the project in 2011, and position remained as at the year 2018. According to Iyamu and Mgudlwa (2018), ANT is significantly a useful tool in that it recognises that there are many realities which are experienced and enacted by different actors, without neglecting their inter-elatedness even in the worst of complexity. Some actors expressed their preferences between agile and waterfall, based on personal or the project interest, or both.

6.2. Interessement

Although the actors were interested in determining whether both agile and waterfall can be used as methods in the development of software by virtual teams, there were challenges such as differences in time-zones and spoken languages. Some of the members were heterogeneous in their actions and within the structure, being a part of both scrum and waterfall methods' teams. From language perspective, some team members spoken in more than one language, which gave them domineering power. Human action is composed of structures of meaning, power and moral frameworks, to choose or express preference. The idea of deciding between the agile and waterfall methods was to find a common ground, and foster relationships within the teams. This increases the interest based, and ease allocation of tasks, which constitute empowerment among participatory individuals. Interessement occurs, when other actors align their interests to that problematised activity (Dwiartama & Rosin, 2014), gaining empathetic view between agile and waterfall methods.

The various teams that were created in the various countries comprised of diverse types of actors. In each of these teams, there was heterogeneous actor, who shown interest in the individual members and the success of the teams including the type of tools that had to be employed. This was primarily because the interest of the human actors and the selected tools (agile or waterfall) will influence the outcome of the project.

6.3. Enrolment

The ANT approach allows actors to ask questions of how rather than why. In the context of DePaul Virtual Software project, members of a team enquired about other teams' preferences of methodology,

between agile and waterfall. Some of them did so because they wanted to align with the global teams' interest. Also, the sizes of each team were not a determining factor in the methodology preference. In addition, the question was to understand the allocation of roles and responsibilities in the project. This include how decisions were made, and how team members and coordinators (scrum masters and project manager) were locked in negotiation, to a make decision based on the preferences and choices from the various teams. In the process, there was no dispute, interests of the teams were well aligned, which made it easier for the project manager to execute his roles and responsibilities. In ANT, this is a situation where actors become established as powerful through the stability of the networks that pass through them.

The key roles played by the various actors, including the focal actor, to consolidate and perform tasks were possible through heterogeneous networks (Islam et al., 2019). The project manager was aware that the success of the teams depends on actors' participation in the discourse, and ultimately the activities of the project. In ANT, this is described as stability, cordial relationship where actors participate and influence the shape of the heterogeneous networks (Dwiartama & Rosin, 2014).

6.4. Mobilization

At this stage the virtual teams were considered successful in that more actors voluntarily took responsibilities in sharing their experiences and making decision in the selection of the method for software development. At this stage the outcome of the question, whether global virtual software teams are more successful with traditional development approach compare to agile practices was facilitated by scrum masters and project manager through interaction and relationship within the teams. Both scrum masters and project manager exercised their power to mobilize the teams to a concession. According to Law (2009), ANT is rather used to view or assess the impact of power on a successfully assembled network.

This helps to identify why some team members were reluctant to certain technology and approaches (Cresswell, 2019). Both project manager and scrum masters were by default designated spokespersons within and on behalf the teams. The action of the project manager and scrum masters help to transform socio-technical processes within the teams. Motivation of the teams' members including the project manager and scrum masters was completing the project, irrespective of the method, agile or waterfall applied. This made it easier for the spokespersons. According to Walsham (1997), ANT is used to examine the motivations and actions of groups of actors, which are part of a heterogeneous network of aligned interest.

7. DISCUSSION OF THE FINDINGS

In conducting the analysis, fundamental questions were posed: Giving the fact that all the virtual teams have the same challenges, which include time zone differences, cultural differences, language difference, does using a difference software development approach matter? That is, is there a significant result to the outcome of the software development project among virtual global teams using a different software development approach? As mention in the introduction, ANT was used to underpin the study, which means that the theory was employed in answering the research question, including examining the insights and trends of global virtual teams in the selection and use of agile or waterfall method for software development.

The four moments translation were employed to guide the analysis. This includes gaining an understanding of the interaction between team members and methodology (agile or waterfall) consisting of virtual teams. Based on the analysis, our observation and experience show that there are no significant differences. This paper demystifies the notion that traditional project management and waterfall are appropriate for global teams. The same argument does not apply to agile practices particularly scrum, which is likely inappropriate for global virtual teams. Each team from 2011 to 2018 have been successful at various degree, and there was no team with a catastrophic failure. All

the application developed by each team throughout this experiment works with various degree of functionality.

The teams that experienced most challenges between 2011 to 2018reported that software development practice was not the reason their team struggles. Through the use of the four moments of translation from the actor-network perspective, in the data analysis, and interpretation of findings, this paper evidently shows how both global software teams and co-located can choose either agile or traditional (waterfall) to successfully develop a software application. Based on our analysis and experiment following the translation process, six main factors or challenges were identified, they are: 1. size of virtual teams; 2. structural formation of the teams; 3. complexity of software; 4. project documentation, communication and collaboration; 5. requirements' clarification; and 6. gaps in cultural and spoken languages. The challenges are discussed as follows:

7.1. Team Size

In the DePaul Virtual Software Team experiment, each team was made of a minimum of 5 people, and maximum of 7 people. Although scrum recommends that scrum team size can be 7 +/- 2 = 9 or 5. We find that smaller team size helps to build relationship and trust regardless to the development method. The closer the relationship, the easier it is to build trust among members. This was vital in that in the process we experienced that building trust in a distance is more difficult compare to building trust among co-located teams. We also noticed that teams that were made up of 5 people tend to have higher level of communication than the other teams of larger numbers. The reasons for this was not clear because the program director and program management team didn't feel that there was a significant difference between five and seven people in a team. The program management team includes people from Chicago and other participating sites like South Africa.

Based on the different cultural affiliation of the sites, it was thus critical to interact and maintain cordial relationship, for a common goal. Laasch, Moosmayer & Arp (2020) explain that enrolment of different human actors into a network is based on competence. Size was important so as to contentedly manage the interaction and relationship within the team, which were the primary factors that influence productivity. The size of the team also helps to follow the actors in their activities toward productivity. From ANT perspective, following the actors helps to understand the traces of the minds through which motive and actions can be examined (Iyamu & Roode, 2012). Based on the virtual nature of the project, it was important to maintain a team size that is manageable. From 2011 to 2014 while the global teams adopt the traditional approach to software development, we noticed that the smaller the team size the better is the quality of communication and cordial relationship among team members. This observation was more pronounce when the global teams uses scrum agile primarily from 2016.

7.2. Team Structure

Each team had a Project Manager who was responsible for managing the team, or a Scrum Master who was responsible for project's processes. Roles across all the teams included a core development team, which was mostly technical people, and often distributed globally. Other roles were a project manager, a scrum master, a product owner, and there was general technology support team for all the teams. While the roles and responsibilities within the teams have changed over time, we maintained very similar structure. Change began to occur based on the importation of heterogeneity of new culture and approaches (Markus & Rowe, 2018) in the virtual teams. That is, the technical development team members were global in nature, while the less technical team members which were traditional project manager, or a scrum master, or product owner were often centralized in Chicago. However, we've experimented with scrum masters from Germany, but all product owners were in Chicago. This structure has consistently delivered products or projects with various degree of success. In the minimum, at the end of 10 weeks (time was fixed) each team had delivered a working software prototype.

The team structure worked because of the ability to translate the roles, responsibilities, processes, and activities to the various teams and individual members. Translation is one of ANT strengths, here it helps to address challenges within global virtual teams by better accounting for technology property and social reality. Translation of the episode makes participation look easy, as team members embrace negotiation in various deliverables while using agile or waterfall for software development. This strength of ANT can be associated with Walsham and Sahay (1999) explanation that: "aligned interests are created through the enrolment of sufficient body of allies and the translation of their interests so that they are willing to participate in particular ways of thinking and acting that maintain the network" (p. 42). The structure was simple, and aligns to software development practice, from project management and scrum maters perspectives.

7.3. Software Complexity

Intentionally, the complexity of each application was very minimum because of the time and resource required to organize the teams. Each application has about five features. In one case, we had two teams working on a large application. The tasks were split between the teams, to avoid duplication, overlap, gap, or conflict. One team was responsible for the front-end of one application, and the other focused on the backend of the same application. In this case both team collaboration was critical in delivering the final software product. The project was part of a larger program, which made it even more crucial, in order to make the program a success. The program director emphasis on the need for both teams to collaborate and have regular software-code integration. The two teams in this case were considered high performing teams. At program level, new or inexperienced teams' members were not recommended to undertake complex tasks. However high performing teams with highly skilled members may be able take on a program level global project regardless of the software development methodology that was selected.

Problematisation helps to articulate and formulate a problem by a focal actor within a network (Callon, 1986), without which the complexity of software persists. Articulation draws interest and encourages actors to partake, thereby contributing their knowhow, which simplifies the complexity. Translation is a relational approach, which employs interconnectedness of actors toward a common goal (Law, 2009). However, the project manager and scrum masters understand that interactions are not fixed, meaning it can dynamically change as conditions change and other factors come to play. In such circumstances, ANT prescribes following the actors (Latour, 2005). This helps to gain deeper understanding of the involvement of social context, which makes the activities and processes of developing, and implementing software more complex in nature. The social context in technology makes it more challenging, whether in the development or implementation stages, hence, the use of a lens such as ANT helps to unpack and understand complexities within an environment (Mkhomazi & Iyamu, 2013).

7.4. Project Documentation, Communication and Collaboration

During the traditional projects from 2011 – 2014, clear project documentations were required. This was primarily because documentation was critical to change management, application development and maintenance, and user friendliness. As a result, decision was made that final project demonstration must be accompanied by the appropriate documentations. From 2016, the only key documentation was a working software. The team from 2016 still maintain project documentation in Trello (the collaboration tool) but the emphasis shifted to working software. During the experiment, we found out that although having a good documentation helped in 2011-2014, but a working prototype was more relevant as we progressed in 2016 projects. In general, we observed that the team from 2016 experienced more of the typical global challenges than the teams that existed between 2011 – 2014. This is primarily attributed to a more intensive communication and collaboration. It also shows that teams from 2016 tend to deliver more software features. The teams from 2016's general output had more features than the previous teams did. However, all the teams met the project objectives. This means,

from our experiment, controlling the amount of changes that is made to requirements for traditional global teams was very helpful for successful software delivery. While changes to requirements for scrum teams make no difference to project success.

Although human actors are not distinguished in the beginning, they do differ in purpose, particularly in their roles (Iyamu & Roode, 2012). This is primarily because actors are unpredictable in a network (team), and translation is not deterministic for their actions. It is therefore critical to examine actors' motivations and actions particularly those that are part of heterogeneous network of aligned interests (Walsham, 1997). Once the project manager was convinced and clear of various actors' interest in the use of agile or waterfall approach, it became easier to allocate tasks, in the areas of documentation of events, communication content and channel, and collaborative processes. From ANT perspective, sociotechnical objects (software development) that are created by humans, using different skills, through translation and negotiation including the use of power is synonymous to change. To avoid a change in the middle of the project, it was inevitable to make the decision, agile or waterfall, in the beginning.

7.5. Requirement Clarification

Clarification of requirements was a critical factor for both agile approach and the traditional approach to global software development. We observed that regardless of the method, agile or traditional, the clearer the requirements the easier it was for the teams to execute their tasks. While the scrum teams were open to changes and reprioritization of features from sprint to sprint. The clearer the goal of the sprint or the project the more successful the teams were. Our take from this experiment is that the clearer the objective of the project the easier it is to select the development approach. The project goal and clarity of scope and requirements should be the determinate of the choice of development approach, traditional or agile.

Problematisation of the requirements defines a system of alliances, or associations, between entities, thereby revealing actors' preferences between agile or waterfall approaches. This therefore requires a detailed examination of the interactions between the actors, which consciously or unconsciously manifest from and their relationship. According to Law (2007), ANT is good in conducting analysis of actors' relationship in the social world in which they are natural settings. Collaboratively, Avgerou (2010) argues that ANT is needed to identify the relevance and consequences of interactions influenced by "local culture." Based on the translation of the requirements, we were able to negotiate and clarify the objectives and factors of influence to a point of stability. Without clarification of the requirements, the project would have been severely challenged or derailed, irrespective of the approach, agile or waterfall that was selected.

7.6. Cultural and Language Gap

There were differences in cultural affiliations and languages spoken by members of the teams. These differences influenced the teamwork and collaboration. Regardless of the development method that was selected, reducing cultural and emotional gap were more of key success factors. The linkages in the operations that connects entities (Bilodeau & Potvin, 2018), such culture, language and methodological approaches were revealed. We found out that in teams where they build more social and emotional relationship they tend to deliver better quality software than in teams where there was a lot of frictions among team members, which often is either due to language difference or cultural differences. We noticed that when teams are struggling due to socio-cultural and emotional gaps, the software development approach adopted in those teams does not matter so much.

Actors and their networks are not independent of each other (Thapa, 2011). By their association tired by cultural affiliation and spoken languages, actors stimulate each other's interest, and together stabilise the team (network). In such situation, actors have the ability to act and also alter each other's actions (Durepo & Mills, 2012). Based on a stable network, actors consciously or unconsciously create links which are difficult to break by 'outsiders. Through a state of stability, the teams were

uniform and collective in their decision to employ agile or waterfall, or both approaches (hybrid). The program director makes sure the teams have a good understanding of both agile and traditional very early before the global project begins. Through an understanding of the roles of culture and language, translation and negotiation of processes, allocation of tasks, and participation in teamwork were easier to manage.

8. CONCLUSION: CHOOSING A GLOBAL DEVELOPMENT APPROACH

One of the questions we try to demystify in the paper is if global virtual software teams are more successful with traditional development approach compare to agile practices. That is, does the choice of agile approach or traditional approach matter for global software teams. The answer based on this experiment is that the development approach doesn't matter. Four factors matter the most: (1) the structure of the team, which defines roles and responsibilities; (2) the nature of the project, which helps to determine and select team members; and (3) reducing the socio-cultural and emotional gap among team members and (4) clarity of the project requirements either from sprint to sprint or as the requirement changes and control in the traditional environment. The findings from this study is intended to benefits software developers, project managers, scrum managers, and collaborative teams.

From theoretical viewpoint, this study made two significant contributions. First, through ANT, we further theorize how the formation and activities of global virtual teams are carried out from software development perspective. Secondly, the study contributes to ANT pedagogy for data analysis and interpretation in IS research through its comprehensive use to gain deeper understanding in the conceptualization of human and nonhuman heterogeneity from competence viewpoint.

We recognize that more data is needed to reach a more generalizable conclusion. This is planned to be a longitudinal study and preliminary results are just being summarized for publication purposes. The study lays foundation for further researches.

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). *Agile software development methods: Review and analysis*. arXiv preprint arXiv:1709.08439.

Adelakun, O., Garcia, R., Tabaka, T., & Ismail, R. (2017). Hybrid Project Management: Agile with Discipline. In CONF-IRM (p. 14). Academic Press.

Almeida, F., & Simões, J. (2019). Moving from waterfall to agile: Perspectives from IT Portuguese companies. *International Journal of Service Science, Management, Engineering, and Technology*, *10*(1), 30–43. doi:10.4018/IJSSMET.2019010103

Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boiangiu, C. A. (2019). A study on using waterfall and agile methods in software project management. *Journal of Information Systems & Operations Management*, 125-135.

Augustine, S., Payne, B., Sencindiver, F., & Woodcock, S. (2005). Agile project management: Steering from the edges. *Communications of the ACM*, *48*(12), 85–89. doi:10.1145/1101779.1101781

Avgerou, C. (2010). Comments on Allen Lee's article 'Retrospect and prospect': Information systems research in the last and next 25 years' from a socio-theoretical IS research perspective. *Journal of Information Technology*, *25*(4), 355–357. doi:10.1057/jit.2009.29

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), 26–30.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). *Manifesto for agile software development*. Academic Press.

Bilodeau, A., & Potvin, L. (2018). Unpacking complexity in public health interventions with the Actor–Network Theory. *Health Promotion International*, *33*(1), 173–181. PMID:27492825

Binder, J., Aillaud, L. I., & Schilli, L. (2014). The project management cocktail model: An approach for balancing agile and ISO 21500. *Procedia: Social and Behavioral Sciences*, *119*, 182–191. doi:10.1016/j.sbspro.2014.03.022

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, *22*(5), 30–39. doi:10.1109/MS.2005.129

Callon, M. (1986). Some elements of a sociology of translation: Domestication of the scallops and the fishermen of St Brieuc Bay. *The Sociological Review*, *32*(1, suppl), 196–233.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133. doi:10.1109/2.963450

Cresswell, K. (2019). Using Actor-Network Theory to Study Health Information Technology Interventions. *Studies in Health Technology and Informatics*, *263*, 87–97. PMID:31411155

Durepos, G., & Mills, A. J. (2012). Actor-network theory, ANTi-history and critical organizational historiography. *Organization*, *19*(6), 703–721. doi:10.1177/1350508411420196

Dwiartama, A., & Rosin, C. (2014). Exploring agency beyond humans: The compatibility of Actor-Network Theory (ANT) and resilience thinking. *Ecology and Society*, *19*(3), 28–38. doi:10.5751/ES-06805-190328

Fenwick, T., & Edwards, R. (2019). *How is actor-network theory contributing to educational research? A critical revisitation. In Revisiting Actor-Network Theory in Education*. Routledge.

Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management—Agilism versus traditional approaches. *Journal of Computer Information Systems*, *49*(2), 10–17.

Gemino, A., Horner Reich, B., & Serrador, P. M. (2021). Agile, traditional, and hybrid approaches to project success: Is hybrid a poor second choice? *Project Management Journal*, *52*(2), 161–175. doi:10.1177/8756972820973082

Heeager, L. T., & Nielsen, P. A. (2018). A conceptual model of agile software development in a safety-critical context: A systematic literature review. *Information and Software Technology*, *103*, 22–39. doi:10.1016/j.infsof.2018.06.004

Heeks, R., & Stanforth, C. (2015). Technological change in developing countries: Opening the black box of process using actor–network theory. *Development Studies Research*, *2*(1), 33–50. doi:10.1080/21665095.2015.1026610

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, *34*(9), 120–127. doi:10.1109/2.947100

Islam, A. N., Mäntymäki, M., & Turunen, M. (2019). Why do blockchains split? An actor-network perspective on Bitcoin splits. *Technological Forecasting and Social Change*, *148*, 119743. doi:10.1016/j.techfore.2019.119743

Iyamu, T., & Mgudlwa, S. (2018). Transformation of healthcare big data through the lens of actor network theory. *International Journal of Healthcare Management*, *11*(3), 182–192. doi:10.1080/20479700.2017.1397340

Iyamu, T., Nehemia-Maletzky, M., & Shaanika, I. (2016). The overlapping nature of Business Analysis and Business Architecture: What we need to know. *The Electronic Journal of Information Systems Evaluation*, *19*(3), 169–179.

Iyamu, T., & Roode, D. (2010). Theoretical analysis of strategic implementation of enterprise architecture. *International Journal of Actor-Network Theory and Technological Innovation*, *2*(3), 17–32. doi:10.4018/jantti.2010070102

Kettunen, J., & Lejeune, M. A. (2020). Waterfall and Agile Product Development Approaches: Disjunctive Stochastic Programming Formulations. *Operations Research*, *68*(5), 1356–1363. doi:10.1287/opre.2019.1977

Khoza, L., & Marnewick, C. (2020). Waterfall and Agile information system project success rates-A South African perspective. *South African Computer Journal*, *32*(1), 43–73. doi:10.18489/sacj.v32i1.683

Kusters, R. J., van de Leur, Y., Rutten, W. G., & Trienekens, J. J. (2017, April). When agile meets waterfall-investigating risks and problems on the interface between agile and traditional software development in a hybrid development organization. In *International Conference on Enterprise Information Systems* (Vol. 2, pp. 271-278). SCITEPRESS. doi:10.5220/0006292502710278

Laasch, O., Moosmayer, D. C., & Arp, F. (2020). Responsible practices in the wild: An actor-network perspective on mobile apps in learning as translation (s). *Journal of Business Ethics*, *161*(2), 253–277. doi:10.1007/s10551-019-04214-8

Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, *53*(3), 276–290. doi:10.1016/j.infsof.2010.11.010

Latour, B. (1999). On recalling ANT. *The Sociological Review*, *47*(1, suppl), 15–25. doi:10.1111/j.1467-954X.1999.tb03480.x

Latour, B. (2005). *Reassembling the social: An introduction to actor-network-theory*. Oxford University Press.

Law, J. (2009). Actor network theory and material semiotics. *The New Blackwell Companion to Social Theory*, *3*, 141–158. doi:10.1002/9781444304992.ch7

Law, J. (1992). Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. *Systems Practice*, *5*(4), 379–393. doi:10.1007/BF01059830

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., & Zelkowitz, M. (2002). *Empirical findings in agile methods.* In *Proceedings of the Conference on Extreme Programming and Agile Methods* (pp. 197-207). Springer. doi:10.1007/3-540-45672-4_19

Markus, M. L., & Rowe, F. (2018). Is IT changing the world? Conceptions of causality for information systems theorizing. *Management Information Systems Quarterly*, *42*(4), 1255–1280.

McHugh, M., McCaffery, F., & Casey, V. (2012). Barriers to adopting agile practices when developing medical device software. In *Proceedings of the International Conference on Software Process Improvement and Capability Determination* (pp. 141-147). Springer. doi:10.1007/978-3-642-30439-2_13

Mkhomazi, S. S., & Iyamu, T. (2013). A guide to selecting theory to underpin information systems studies. In *International Working Conference on Transfer and Diffusion of IT*. Springer. doi:10.1007/978-3-642-38862-0_33

Moniruzzaman, A., & Hossain, D. S. A. (2013). *Comparative Study on Agile software development methodologies.* arXiv preprint arXiv:1307.3356.

Nehemia-Maletzky, M., Iyamu, T., & Shaanika, I. (2018). The use of activity theory and actor network theory as lenses to underpin information systems studies. *Journal of Systems and Information Technology*, *20*(2), 191–206. doi:10.1108/JSIT-10-2017-0098

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78. doi:10.1145/1060710.1060712

Rahmanian, M. (2014). A Comparative Study on Hybrid IT Project Management. *International Journal of Computer and Information Technology*, *3*(05), 1096–1099.

Rehman, I. U., Rauf, A., & Shahid, A. A. (2010). Scope management in agile versus traditional software New York, NY. doi:10.1145/1890810.1890820

Rose, K. H. (2013). A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. Project Management Journal, 44(3), e1.

Royce, W. W. (1970). Managing the development of large software systems. In *Proceedings of the International Conference on Software Engineering*. IEEE WESCON.

Sage, D., Vitry, C., & Dainty, A. (2020). Exploring the organizational proliferation of new technologies: An affective actor-network theory. *Organization Studies*, *41*(3), 345–363. doi:10.1177/0170840618815524

Sandberg, J., & Alvesson, M. (2011). Ways of constructing research questions: Gap-spotting or problematization? *Organization*, *18*(1), 23–44. doi:10.1177/1350508410372151

Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Prentice hall.

Seymour, T., & Hussein, S. (2014). The history of project management. *International Journal of Management & Information Systems*, *18*(4), 233–240.

Shankarmani, R., Pawar, R., Mantha, S. S., & Babu, V. (2012). Agile methodology adoption: Benefits and constraints. *International Journal of Computers and Applications*, *58*(15), 31–37. doi:10.5120/9361-3698

Sixsmith, A., Freeburn, C., & Mooney, G. (2014). *Project Management in Practice: Views from the Trenches. In Proceedings of* the 24th International Business. *Information & Management*.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informações Econômicas*, *17*(4), 64–76. doi:10.12948/issn14531305/17.4.2013.06

Tam, C., da Costa Moura, E. J., Oliveira, T., & Varajão, J. (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, *38*(3), 165–176. doi:10.1016/j.ijproman.2020.02.001

Thapa, D. (2011). The role of ICT actors and networks in development: The case study of a wireless project in Nepal. *The Electronic Journal on Information Systems in Developing Countries*, *49*(1), 1–16. doi:10.1002/j.1681-4835.2011.tb00345.x

Thummadi, B. V., & Lyytinen, K. (2020). How Much Method-in-Use Matters? A Case Study of Agile and Waterfall Software Projects and Their Design Routine Variation. *Journal of the Association for Information Systems*, *21*(4), 864–900. doi:10.17705/1jais.00623

Walsham, G. (1997). Actor Network Theory and IS Research: Current Status and Future Prospects. In A. S. Lee, J. Liebenau, & J. I. DeGross (Eds.), Information Systems and Qualitative Research. London: Chapman & Hall.

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, *15*(3), 320–330. doi:10.1057/palgrave.ejis.3000589

Walsham, G., & Sahay, S. (1999). GIS for district-level administration in India: Problems and opportunities. *Management Information Systems Quarterly*, *23*(1), 39–65. doi:10.2307/249409

Olayele Adelakun is a professor of IS. Research area include outsourcing, Agile, Virtual Teams and Digital Innovation.

Tiko Iyamu holds a PhD in information systems from the University of Cape Town, South Africa. Currently, Iyamu is Research Professor at the Faculty of Informatics and Design, Cape Peninsula University of Technology (CPUT), Cape Town, South Africa. Previously, he was with the Tshwane University of Technology, South Africa, and Namibia University of Science and Technology, Namibia. Iyamu served as a Professor Extraordinaire at the Department of Computer Science, University of the Western Cape, from 2006 to 2015. In 2014, he was a visiting professor at the Flensburg University of Applied Sciences, Germany. Professor Iyamu's areas of focus include enterprise architecture, health informatics, big data analytics, and mobile computing. In addition, he has keen interest in the application of sociotechnical theories, such as actor-network theory, structuration theory, and diffusion of innovation in information systems research. He has authored 6 books, and over 140 peer-reviewed research articles in journals, conference proceedings and book chapters.