

Challenges of Agile Development and Implementation in a Developing Country: A Zambia Case Study

Douglas Kunda, Mwenge Mulenga, Muwanei Sinyinda and Victoria Chama

Mulungushi University, School of Science, Engineering and Technology, Kabwe, Zambia

Article history

Received: 28-06-2017

Revised: 21-07-2017

Accepted: 29-08-2017

Corresponding author:

Douglas Kunda

Mulungushi University, School of Science, Engineering and Technology, Kabwe, Zambia

Email: dkunda@mu.edu.zm

Abstract: Agile development is a software development process that advocates adaptive planning, early delivery, evolutionary development and continuous betterment and supports rapid and flexible response to change. The purpose of Agile development is minimize project failure through customer interactions and responding to change. However, Agile development is vulnerable to failure because of a number of factors and these factors can be categorized under four dimensions, namely; organizational, people, process and technical. This paper reports the result of a study aimed at identifying factors that influence success and/or failure of Agile development in a developing country, Zambia. A multiple case study approach and grounded theory approach was used for this case study. The study shows that there are challenges that are unique to developing countries and therefore measures should be developed to address these unique problems when implementing Agile projects in developing countries.

Keywords: Agile Development, Challenges of Agile Development, Developing Countries

Introduction

While software is so important for all aspects of the modern world, software development itself has many challenges. Agile software development methods have recently emerged as a new and different way of developing software to address the challenges faced by traditional software development methodologies. Agile software development is a set of principles for software development under which requirements and solutions progress through the combined effort of self-organizing and cross-functional teams (Collier, 2011). It promotes adaptive planning, evolutionary development, early delivery and continuous improvement and it encourages rapid and flexible response to change. However, their achievement has mostly been anecdotal and research in this subject is still limited in the academic circles (Chow and Cao, 2008).

Experienced software practitioners created a set of practices or methods for Agile software development. These methods can be seen as a reaction to plan-based or traditional methods, which emphasize “a rationalized, engineering-based approach” in which it is claimed that

problems are fully specifiable and that optimal and predictable solutions exist for every problem (Nerur *et al.*, 2005; Dyba, 2000). According to Boehm (2002) the “traditionalists” are said to advocate extensive planning, codified processes and rigorous reuse to make development an efficient and predictable activity. On the other hand by contrast, Agile processes address the challenge of an unpredictable world by relying on “people and their creativity rather than on processes” (Nerur *et al.*, 2005; Dyba, 2000).

Williams and Cockburn (2003) state that Agile development is “about feedback and change”, that Agile methodologies are established to “embrace, rather than reject, higher rates of change”. The “Agile manifesto” written by the practitioners states that Agile development should concentrate on four core values (Dingsøyr *et al.*, 2012):

- Customer collaboration over contract negotiation
- Working software over comprehensive documentation
- Individuals and interactions over processes and tools
- Responding to change over following a plan

Agile methods apply a human-centered approach to software production and aims to deliver high-quality products faster, producing satisfied customers (Ceschi *et al.*, 2005). Ohno (1988) states that “lean production is a set of practices focused on the continuous improvement of the production process, by identifying and removing anything that does not add value to the customer”. Agile methods apply the principles of lean production borrowed from manufacturing environment to the overall software life cycle. So, these methods focus on providing value for the customer and support requirements variability, but they don’t fit every application domain (Poppendieck and Poppendieck, 2003).

Highsmith and Cockburn (2001) argue that Agile approach is more people-oriented rather than process-oriented. This means that it depends heavily on individual skills. Agile methods claim that people make projects successful and that no procedure will ever make up for the lack of the development team’s ability; so, a procedure’s role is to support the development team. Additionally Highsmith and Cockburn (2001) argued that these methods promote the cohesion of team members and developer and customer interaction. According to Boehm and Turner (2005), Agile methods are lightweight processes that employ short iterative cycles, actively involve users to create, prioritize and validate requirements and rely on a team’s tacit knowledge as opposed to documentation. A truly Agile

method must be iterative (take several cycles to complete), incremental (not deliver the entire product at once), self-organizing (teams determine the best way to handle work) and emergent (processes, principles and work structures are recognized during the project rather than predetermined) (Boehm and Turner, 2005). Figure 1 shows an example of an Agile process flow.

It promotes adaptive planning, evolutionary development, early delivery and continuous improvement and it encourages rapid and flexible response to change. However, their achievement has mostly been anecdotal and research in this subject is still limited in the academic circles (Chow and Cao, 2008).

Experienced software practitioners created a set of practices or methods for Agile software development. These methods can be seen as a reaction to plan-based or traditional methods, which emphasize “a rationalized, engineering-based approach” in which it is claimed that problems are fully specifiable and that optimal and predictable solutions exist for every problem (Nerur *et al.*, 2005; Dyba, 2000). According to Boehm (2002) the “traditionalists” are said to advocate extensive planning, codified processes and rigorous reuse to make development an efficient and predictable activity. On the other hand by contrast, Agile processes address the challenge of an unpredictable world by relying on “people and their creativity rather than on processes” (Nerur *et al.*, 2005; Dyba, 2000).

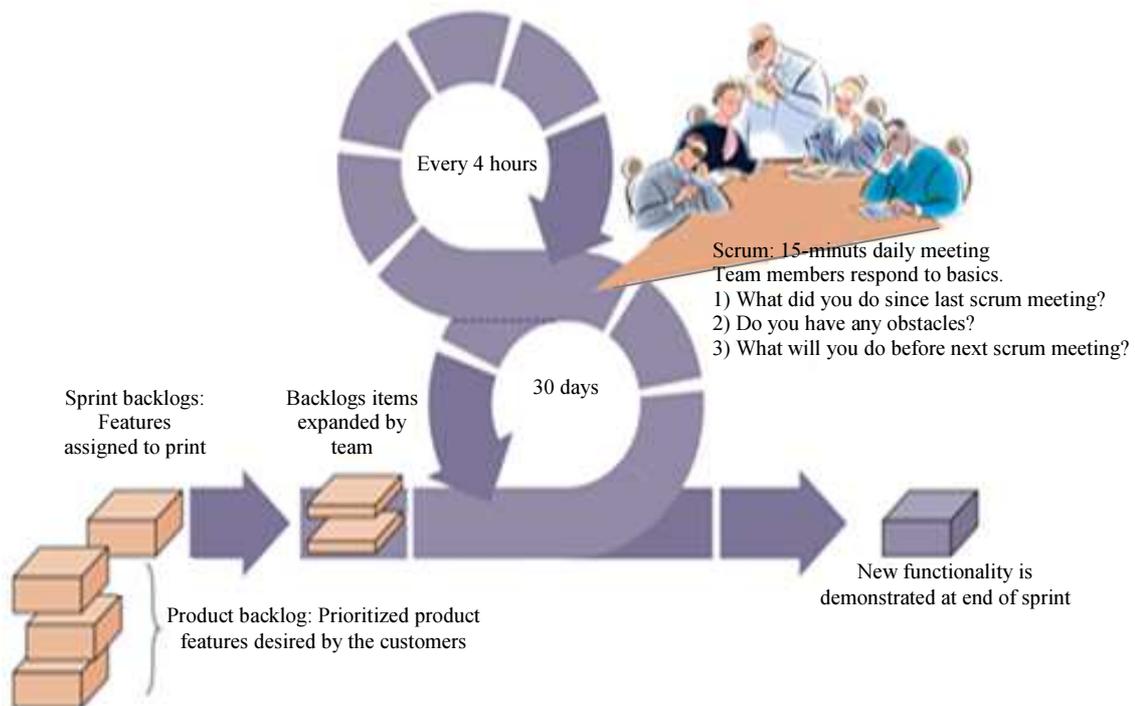


Fig. 1: The Scrum process an example of an agile process flow (Boehm and Turner, 2005)

Williams and Cockburn (2003) state that Agile development is “about feedback and change”, that Agile methodologies are established to “embrace, rather than reject, higher rates of change”. The “Agile manifesto” written by the practitioners states that Agile development should concentrate on four core values (Dingsøyr *et al.*, 2012):

- Customer collaboration over contract negotiation
- Working software over comprehensive documentation
- Individuals and interactions over processes and tools
- Responding to change over following a plan

Agile methods apply a human-centered approach to software production and aims to deliver high-quality products faster, producing satisfied customers (Ceschi *et al.*, 2005). Ohno (1988) states that “lean production is a set of practices focused on the continuous improvement of the production process, by identifying and removing anything that does not add value to the customer”. Agile methods apply the principles of lean production borrowed from manufacturing environment to the overall software life cycle. So, these methods focus on providing value for the customer and support requirements variability, but they don’t fit every application domain (Poppendieck and Poppendieck, 2003).

Highsmith and Cockburn (2001) argue that Agile approach is more people-oriented rather than process-oriented. This means that it depends heavily on individual skills. Agile methods claim that people make projects successful and that no procedure will ever make up for the lack of the development team’s ability; so, a procedure’s role is to support the development team. Additionally Highsmith and Cockburn (2001) argued that these methods promote the cohesion of team members and developer and customer interaction. According to Boehm and Turner (2005), Agile methods are lightweight processes that employ short iterative cycles, actively involve users to create, prioritize and validate requirements and rely on a team’s tacit knowledge as opposed to documentation. A truly Agile method must be iterative (take several cycles to complete), incremental (not deliver the entire product at once), self-organizing (teams determine the best way to handle work) and emergent (processes, principles and work structures are recognized during the project rather than predetermined) (Boehm and Turner, 2005). Figure 1 shows an example of an Agile process flow.

Two popular Agile techniques are Extreme Programming (XP) and Scrum. XP aims at enabling successful software development despite vague or

constantly changing requirements in small- and medium-size teams while Scrum on the other hand aims at managing the development process through an empirical approach that applies the ideas of industrial process control theory to software development (Abrahasson *et al.*, 2001). XP contain activities that occur with each iteration. Code is developed by pairs of programmers, tested and integrated in very small increments. Requirements are elicited from the user *stories* (such as use cases) as well as intimately involve the customer with what and when code is implemented based on the progress of the development and the planning results. XP has rules that govern what *small increment* really means. Planning also includes what many would call project estimating, tracking and controls, as well as changes to how future XP cycles will apply the experience gained from the previous iterations. Designing and coding are distinct activities in XP. However, they occur along with testing in very tight yet simple formation. XP does not have a coding standard, except to specify that there must be one (Glazer, 2001). Figure 2 above depicts a typical XP project with iterations.

Background and Related Work

According to the study done by Ceschi *et al.* (2005) introducing Agile methods also proposals improvements in requirements management, quality, team satisfaction and customer satisfaction. However the main problems caused introducing Agile methods are the lack of a detailed preliminary cost evaluation and the troubles new concepts such as pair programming, test first and customer’s on-site cause. The primary difficulty seems to be cultural: people both customers and developers don’t easily accept drastic changes in traditional environments.

In a Scrum environment, it’s often difficult to measure success against a predetermined plan because Scrum lets business owners adapt and change their plan every sprint (Schatz and Abdelshafi, 2005). Karlstrom and Runeson, (2005) argue that developers focused on past and present releases, while the managers focused increasingly on present and upcoming releases. This is problematic when technical issues are raised too early for management and there are disagreements between developers and management. Lack of attention to design and architectural issues by developers has repeatedly mentioned in the literature as limitation of Agile development methods (Mcavoy and Butler, 2007; Stephens and Rosenberg, 2003). In the study conducted by Tessem (2003), the planning game practice of XP was used to estimate the size of work and estimates made at the beginning of the project were about one third of what turned out to be correct, which is explained by both the team’s lack of estimation experience and coarse user stories. This may also suggest that although Agile development methods promises early delivery of the software in reality it has its own limitations.

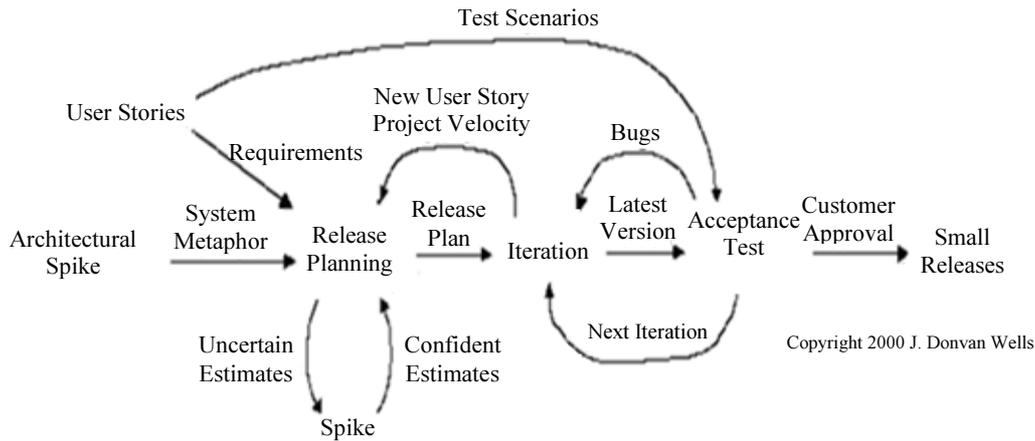


Fig. 2: Extreme programming project (Glazer, 2001)

Agrawal *et al.* (2016) argue that the major limitations of Agile methods are lack of up front planning and this makes it difficult for both developers and managers to monitor the progress of the development process. Some other limitations are budget constraints, lack of sufficient documents; don't go with SDLC steps, lack of predictability, a lot of meetings, regular compliance, requirement of training and not for small organization (Agrawal *et al.*, 2016). This is because Agile method favours the changing of requirements even very late. Kaijo-Mattsson (2008) conducted a study which examined problems that were encountered within 18 Agile organizations. The examined problems were grouped into two categories which are product related problems and process related problems. The following is a summary of some of the problem which was identified in the study:

- Oral communication takes time
- It is difficult to find a person who has the right answer
- Engineers do not remember what changes they made and why
- Engineers repeat many of the former mistakes
- Organizations may undergo major initial pain in their change process after key engineers quit. The pain is defined in terms of loss of time and profit
- Engineers do not know the whole system. Knowledge of the system is spread in the heads of several engineers.
- Architecture deteriorates substantially
- Many product and process issues get misunderstood
- When scaling up, teams have become more difficult to coordinate, systems have become more difficult to understand and change and engineers have started losing track of the company's vision

Svensson and Host (2005) conducted a study aimed at introducing a process based on XP to a large software development company and the process was introduced to a pilot team that worked for eight months. Svensson and Host (2005) concluded that the introduction of the process proved difficult, due to the complexity of the organization. They advise companies that want to introduce Agile development methods to assess existing processes, with the following goals in mind: determining what to introduce which should introduction to new concepts; clarifying terminology to simplify communication with the rest of the company; avoiding underestimating the effort needed to introduce and adapt XP which will help in proper planning and cost evaluation ; and introducing the practice of continuous testing early, because it takes time and effort to introduce this properly. Hilkka *et al.* (2005) emphasize the importance of skilled team members with solid domain knowledge: “without these kinds of persons, the chosen approach would probably have little possibility to succeed”. This solution can be used to solve the introduction to new concepts explaining technical issues to management if and when they are introduced. In addition, they found that it is also possible to combine Agile project management with overall traditional principles, such as the stage-gate project management model. A limitation that was brought out in this study is that team members are less interchangeable in Agile teams, which has an impact on how a project is managed and implemented.

Karlstrom and Runeson (2005) studied how traditional stage-gate project management could be combined with Agile methods. In a case study of three large companies, they found that Agile methods provide the stage-gate model powerful tools for micro planning, day-to-day work control and reporting on progress. By using this tool it will be easier to measure success against a predetermined project plan. Vijayasathya and

Turk (2008) indicate that some of the factors that lead to Agile project failure include lack of training and peer support, ignorance of Agile approaches, lack of facilities for pair programming, individuals' resistance and relying only on economic evaluation criteria. Another concern raised is managerial apathy and organizational resistance to change. Kaijo-Mattsson (2008) proposes some measures to address the above-mentioned problems. These measures are:

- To remedy the architectural ad hoc expansion, the organizations have created an *Architect* role assuring the architectural quality. This measure is nothing new and is commonly practiced in traditional environments
- To remedy the communication problem, the organizations have created a *Communication Owner* role, a role responsible for communicating the overall system and process information. The existence of this role however does not guarantee that the valuable information gets documented. This measure is new with respect to traditional environment
- To remedy the scalability problems, the organizations have utilized the practices from traditional evolution and maintenance by creating system/component ownership role. They have also improved their system and process documentation

In literature challenges and failure factors for Agile methods are categorized in several ways. For example, Chow and Cao (2008) presents the failure factors in four dimensions, namely; organizational, people, process and technical. Conboy *et al.* (2011) identified nine categories or themes for challenges experienced by 17 large multinational organisations using Agile methods. The identified categories were: developer fear as a result of the transparency of skill deficiencies; the need for developers to be "master of all trades"; dependency on social skills; the need to understand and learn values and principles of Agile, not just the practices; deficiency of developers' business knowledge; lack of developer motivation to use Agile methods; implications of devolved decision-making; the need for Agile compliant performance evaluation; and absence of specific recruitment policies and absence of trained IT graduates for Agile. Using literature analysis Gandomani and Zulzalil (2013) categorized the challenges faced by organisations when moving to Agile development into four main categories: organisation and management related; people related; process related; and technology and tools related challenges. Based on grounded theory approach Waardenburg and Vliet, (2013) investigated challenges caused by the co-existence of Agile methods

and plan-driven development from two large enterprises in the Netherlands and categorized the challenges under two categories: increased landscape complexity and lack of business involvement.

Gregory *et al.* (2015) used thematic analysis approach to identify 193 challenges and categorized them into seven themes namely: claims and limitations (misconceptions, shortcomings, hype, failure); organisation; culture (organisational, changing mindsets, national, distributed teams, trust); teams; sustainability; scaling (large projects, governance); and value (business value and measurement). Dikert *et al.* (2016) conducted a review on how Agile methods and lean software development has been adopted at scale, focusing on reported challenges and success factors in the transformation. The review identified 35 reported challenges grouped into nine categories and 29 success factors, grouped into eleven categories. The nine categories for the reported challenges are summarized as follows: change resistance; lack of investment; Agile difficult to implement; coordination challenges in multi-team environment; different approaches emerge in a multi-team environment; hierarchical management and organizational boundaries; requirements engineering challenges; quality assurance challenges; and integrating non-development functions. The most salient success factor categories were management support, choosing and customizing the Agile model, training and coaching and mindset and alignment.

This study used the classification model proposed by Chow and Cao (2008) Table 1 presents some of the challenges or failure factors identified in literature.

Case Study Objective and Method

The overall goal of this study was to identify factors that influence success or failure of Agile development in Zambia, a developing country. The immediate objectives of this study are:

- Identify failure factors and challenges experienced by developers and implementer of Agile methodology in Zambia
- Assess techniques and tools for Agile methods and learn how they can promote success of Agile development and implementation in Zambia
- Review the technical and non-technical issues that influence success or failure of Agile development in Zambia
- Investigate whether identified failure factors and challenges of Agile methodology from Zambia can be classified using the four dimensions proposed by (Chow and Cao, 2008), namely; organizational, people, process and technical

Table 1: Challenges/factors

Dimension	Challenges/Factors
Organizational	<ul style="list-style-type: none"> • Lack of executive sponsorship (Chow and Cao, 2008). • Lack of management/stakeholder commitment (Gregory <i>et al.</i>, 2015). It only works if all stakeholders get involved and support the Agile process • Organizational culture being too political (Gregory <i>et al.</i>, 2015). Traditional management may see Agile as just another IT method that can be implemented and structured to “fit” existing organizational norms. • Cultural issues (Gandomani and Nafchi, 2016). Organizations usually dictate their own culture to people and sometimes, the real problem isn't people's culture and habits by its organizational culture • Lack of Agile logistical arrangements (Chow and Cao, 2008). • Existing organizational structure including the company's mind-set that needs to be opened-up for Agile principles (Berger and Eklund, 2015).
People	<ul style="list-style-type: none"> • Lack of necessary skill set (Cordeiro <i>et al.</i>, 2007). • Lack of project management competence (Chow and Cao, 2008). • Ill-defined cost and schedule estimation (Ramesh <i>et al.</i>, 2010). Initial estimates of time and cost are changed substantially by a change in requirements in subsequent stages (Inayat <i>et al.</i>, 2015) • Resistance from groups or individuals (Gandomani and Nafchi, 2016). People are not willing to change unless there are good reasons that they understand and the change is perceived easy enough (Dikert <i>et al.</i>, 2016). Most people are accustomed to their current roles and activities and naturally resist change especially if it involves decreasing their power (Gandomani and Nafchi, 2016). • Bad customer relationship (Baskerville <i>et al.</i>, 2011). • Wrong mindset (Gandomani and Nafchi, 2016). This challenge, which can be seen regarding all stakeholders, mainly arises from perceptions and beliefs about the development process, required roles and responsibilities and their fear of change
Process	<ul style="list-style-type: none"> • Ill-defined project planning (Chow and Cao, 2008); (Conforto and Amaral, 2016). Difficulty in planning and controlling the project due to several uncertainties regarding the technology, systems and sub-systems of the product and the absence of an adapted methodology to their project environment (Conforto and Amaral, 2016). The importance of having frequent project planning sessions in a collaborative and participative approach, using an iterative development technique was identified (Conforto and Amaral, 2016). • Ill-defined management and customer role (Dikert <i>et al.</i>, 2016). The need for additional management positions in a larger organization may pose problems to Agile processes that emphasize self-organization. Especially the role of middle management was unclear in Agile methods • Lack of Agile progress tracking mechanisms (Chow and Cao, 2008). • Lack of customer presence (Chow and Cao, 2008). • Lack of effective collaboration (Gandomani and Nafchi, 2016). Effective collaboration and involvement is necessary for success of Agile development and therefore lack of effective collaboration and involvement is a failure factor for Agile development. Many people have problem in collaboration and communication as they do not have enough confidence to participate in group works and group decision making resulting poor communication. • Inflexible test environment that inhibits fast feedback to changed or added features (Berger and Eklund, 2015) • Multi-team environment coordination difficult (Dikert <i>et al.</i>, 2016). For example problem of achieving technical consistency and interfacing between teams difficult.
Technical	<ul style="list-style-type: none"> • Agile practices customized poorly (Dikert <i>et al.</i>, 2016). Difficulty of and misunderstandings related to Agile were evident in cases where the methods were customized poorly. • Inappropriate technology and tools (Chow and Cao, 2008). • Inappropriate architecture (Ramesh <i>et al.</i>, 2010). • Neglecting non-functional requirements (Inayat <i>et al.</i>, 2015) (Chow and Cao, 2008). • Lack of documentation (Ramesh <i>et al.</i>, 2010). User stories and product backlogs are the only documents in Agile methods. • Lack of knowledge (Gandomani and Nafchi, 2016). Lack of sufficient knowledge about Agile, its principles and its values by team members and other stakeholders such as managers and customers • Requirements engineering challenges (Inayat <i>et al.</i>, 2015; Dikert <i>et al.</i>, 2016). For example, complex interactions and dependencies between requirements, such as user interfaces and quality requirements, are a challenge to capture with acceptance test cases (Bjarnason <i>et al.</i>, 2016).

Multiple case study approach was used to identify the factors that influence the success or failures of Agile development. Three phased approach was used in this

case study comprising preparation phase, data collection and data analysis phase. This is in line with the design principles and procedures described by a number of

researchers (Runeson and Höst, 2008; Pagano and Bruegge, 2013). Data preparation phase involved selecting the data collection method and case selection strategy as well as development of a data collection instrument. The development of the data collection instrument involved extensive analysis of literature on Agile Development, review of the instrument and pilot testing in an organisation.

Data collection phase involved in-depth interviews using open-ended question. It also involved review of documentation thus invoking a process of triangulation. With triangulation, the potential problems of construct validity was addressed, because the multiple sources of evidence essentially provide multiple measures of the same phenomenon (Yin, 1994). Developers and implementers of the Agile projects were interviewed to share their experience in the development and implementation of the Agile projects included in the study. The advantage with interviews is it focuses directly on the case study topic and provides perceived causal inferences.

Data analysis phase involved use of the grounded theory approach. Grounded theory is where the analysis is inductively derived from the study of the phenomenon it represents (Strauss and Corbin, 1990). A grounded theory approach was used because Agile software development offers a people-intensive approach and grounded theory is best fitted for studying people related issues and has been extensively and successfully used in Agile context studies (Coleman and O'Connor, 2007; Baskerville *et al.*, 2011; Hoda *et al.*, 2011; Waardenburg and Vliet, 2013; Gandomani and Nafchi, 2016). The classification model proposed by (Chow and Cao, 2008) was used to classify and analyse data from interviews and then collaborated using documentation. Two additional dimensions (project management and developing countries specific problems) were included. Thus the case study sought to describe the important identified failure factors for Agile development and implementation in line with recommendation that case studies should expose descriptive or explanatory character (Runeson and Höst, 2008). The data analysis was performed by the Corresponding author and was independently validated by the second and third authors.

Background to Organizations

In-depth interviews were conducted in 6 organizations in Zambia where Agile development was used to elicit the failure factors and challenges experienced by developers and implementer. In addition, a deliberate effort was made in selection of organizations to ensure that a mixture of organizations from different sectors and of various sizes were included in this study (Table 2).

Table 2: Case study organizations

Case #	Main business area	Size	No of Developer	Agile Project
1	Health	Small	6	Development of Nursing Information System
2	Government	Medium	5	Development of prison management system
3	Higher Learning Institution	Medium	8	Development of Student Management Information System
4	Agriculture Sector	Large	5	Development of Agriculture Management Information System
5	Local authority	Large	5	Development of PLGO database system
6	Professional Organization	Small	6	Development of Profession Registration and Monitoring Information System

Case 1

This organization provides health services to its clients. The objective of the Agile project was to develop the Nursing Information System to help the organization fulfill her mandate of ensuring public protection through the provision of quality education, training and practice of nurses and midwives. The system implemented was a web application with the following modules: (1) registration module to register nurses and nursing training institutions and to issue annual licenses; (2) accounts receivable module to produce invoices and receipts of clients; (3) monitoring and evaluation module to produce monitoring report of inspections (initial inspections, re-inspections, compliance monitoring/routine inspections) relating to a training institution/programme; (4) system administration module to manage administrative functions such as users, drop down menus and system security (database views and audit trails).

Case 2

This is a government department providing correction services to prisoners. The purpose of the Agile project was to develop the prison sentence management system that would help in management of various categories of inmate sentences nationally. The system implemented was a web application using Java Server Pages and Servlets Technology with the following modules which had all the relevant functions relating to their inmate categories: (1) initial, concurrent, consecutive and subsequent concurrent calculation module (2) aggregated and detailed reporting module (3) prohibited Immigrant module (4) remand module (5) Convicts module (6) lifer module (7) death roll module (8) System administration module to manage administrative functions.

Case 3

This organization is a public university with over 5000 students. The Agile project was aimed at developing a system that would help the University support the management of student information through

key business activities, such as admission, registration, invoicing, accommodation, progression and graduation. The system was required to be linked to financial accounting systems of University and the financial accounting systems needs to linked to banks so that student deposits are automatically reflect on the student financial statement. The system to be implemented was a web application with the following modules: (1) online student application and processing of admission of students; (2) online student course registration linked to moodle elearning software; (3) student examination results module (4) Interface with banking and financial accounting system; (5) reporting module

Case 4

This is a large organization that provides agricultural services to the public. The purpose of the Agile project was to design and pre-test an Agriculture Management Information System (AMIS) database that will hoard all indicators that will be tracked by the sector from the camp up to the central level. The database was to be designed with flexibility built in to allow for an increase in the number of indicators that would be uploaded onto the database and tracked by the sector over time. The scope of the project included: (1) design an AMIS database which will receive, archive, process and generate reports based on the 33 M&E Indicators contained in the M&E manual; (2) Identify an appropriate software which will operate the AMIS database; (3) develop a User's Manual with clear steps and guidelines on data entry, archiving, processing and generation of various M&E reports; (4) pre-test the M&E database to assess its suitability; (5) review current M&E databases in the MAL including the Crop Forecast and Post-Harvest Survey, Livestock Information Management System (LIMS), National Livestock Epidemiology and Information Centre (NALEIC) and Agriculture Marketing Information System to identify common design features, software and elements of the databases which can be integrated into the new database; (6) develop user manual and train the M&E Implementation Core Team.

Case 5

The core function of this organization is to supervise, monitor and evaluate councils in the southern part of Zambia and provide guidance in implementing government policy. The objective of the Agile project was to provide technical expertise to the Provincial Local Government Officer (PLGO) in the establishment of a simplified data bank that stores relevant data, analyse it regularly for reporting, information and decision making. Once the assignment is completed, it is anticipated that PLGO will have a developed and usable data bank and enhanced skills and systems for up-dating the data bank. The scope of the project included: (1) to

review all data collection and processes at PLG; (2) to undertake a data analysis using a formalised methodology to create a database (3) design and develop the PLGO databank; (4) develop guidelines or procedures in maintaining the data bank that will be used as a reference point for the PLGO; (5) To provide training to PLGO in the physical implementation of the so designed data bank; 6)To document the assignment and recommendations in a report.

Case 6

This organization deals with registration and issuance of annual licenses to Health practitioners, Health facilities, Accredited health services, Training institutions, CPD centers, Internship hospitals and sites. The scope of the Agile project involved the automation of the core functions of the organisation as a way of improving its operations. It included the following modules: (1) the registration database module that facilitates the registration and issuance of annual licenses. The system produces a number of monitoring and evaluation reports such as number of registrants on Full Registration; Temporary Registration; Provisional Registration; Specialist Registration; Limited Registration and inactive member. (2) the compliance monitoring and inspection database module that facilitates the inspection, compliance monitoring and issuance of licenses to Health facilities, Health services, CPD centres, Training institutions. The system produces a number of monitoring and evaluation reports that includes the Scoring/Grading used in the inspections tool, the number of health facilities opened/closed/charged. Demographic data is produced according to ownership (Zambian/ non-Zambian), Town/township/area/street, District, Province, Type (private/public /mission), Class (A, B, C, D, E), etc. (3) the procurement and inventory module that facilitates the procurement of goods and services; inventory management and Stock taking. The system produces a number of monitoring and evaluation reports that includes number of items purchased and user by item, user, week, month, quarter and yearly. (4) The accounting module that facilitates accounting and finance. The accounting module included the general ledger, cashbook, accounts receivable, accounts payable, invoicing, sales order processing and report writer.

Results and Discussion

Organizational Factors

Many studies show that the major cause of most software failures including those developed using Agile methods is the non-technical issues such as organisation factors (Kunda and Brooks, 2000; Chow and Cao,

2008; Gandomani and Zulzalil, 2013; Gregory *et al.*, 2016). This is because software systems do not exist in isolation they are used in social and organisational contexts (Kunda and Brooks, 2000). The respondents in four cases indicated that lack of executive support, lack of Agile logistical arrangements and organisational issues are the major organisational factors that impact on the success of Agile development and implementation (Table 3). Respondents pointed out that lack of executive support was demonstrated by budgetary constraints in these projects and the unwillingness of the executive sponsor to provide adequate funds for the projects. This suggests that Agile methods should not be used to implement projects with highly constrained budgets. This methodology is highly dependent on skilled programmers whose services can only be sustained by good remuneration packages. In all the cases there was high dependency on junior programmers so as to meet budgetary shortfalls. This proved costly due to the long learning curve incurred when incorporating junior programmers into the project team as well rectifying bugs generated by these junior programmers.

Gandomani and Nafchi (2016) found that organizational culture rather than people's culture was a challenge in transition to Agile methods and advised that focusing on organizational behaviors and improving them would be the only solution. Respondents from three cases (Case 3, 4 and 5) brought out the challenge of organizational culture and indicated that there was shared assumptions and belief that Agile methods may not bring about the desired benefits. Other organizational failure factors brought out by the case study were lack of management commitment; and lack of reward system to support Agile method.

People Factors

People factor is another challenge for Agile methods. The following people factors were discovered during

data analysis: lack of necessary skill set; lack of commitment from customers; resistance from groups or individuals; and bad customer relationship. Respondents from four cases indicated that there is resistance from groups or employees with ill motives against the implementation of a transparent business workflows (Table 4). They pointed out that employees resisted implementation of transparent business workflows because there was personal financial gain from inefficient system. For example, a system that takes a long to time to provide service to customer provides corrupt employees opportunities to receive bribes in order to provide quick service to those customers who are willing to pay extra fees to these corrupt individuals. It is believed that some of such employees take advantage of the mess for their personal gains and as such will resist any attempt to improve the system. In such cases Agile methods are likely to fail due to the constant change of requirements inspired by hidden motives of resistance to change. Resistance to change is one major challenge identified by Dikert *et al.* (2016) in large scale Agile transformations which includes general resistance to change, skepticism towards the new way of working, top down mandate creates resistance and management unwilling to change.

Respondents argued that most institutions have IT staff whose responsibility includes creating a link between users and the developers. These IT staff are supposed to translate institutional domain knowledge to the developers as well as help the institutions with basic knowledge of the new system. In most cases the IT staff from the client's side was quite insufficient and fail to perform basic system configurations. This created a very big gap in the projects especially where the developers were working remotely. This in turn frustrated users as they could not wait for developers for two or more weeks to come and fix bugs.

Table 3: Organizational factors

Identified factors	Case #					
	1	2	3	4	5	6
Lack of executive sponsorship	X	X		X		X
Lack of management commitment	X		X		X	
Organizational culture			X	X	X	
Lack of Agile logistical arrangements	X	X	X			X
Organization Issues	X	X			X	X
Lack of reward system to support Agile method		X	X		X	
Teams located in different places		X		X	X	X

Table 4: People factors

Identified factors	Case #					
	1	2	3	4	5	6
Lack of necessary skill set	X	X	X			X
Lack of commitment from customers		X		X		
Resistance from groups or individuals	X		X	X		X
Bad customer relationship					X	X

The lack of sufficient IT skill sets from client IT Personnel was brought out by Cases 1, 2, 3 and 6. Customers play an important role in success of Agile methods and they should be responsive, collaborative, authorized, committed and knowledgeable (Conboy *et al.*, 2011; Gandomani and Zulzalil, 2013). Lack of commitment from customers was brought out by respondents from Case 2 and 4. Respondents from Cases 5 and 6 indicated the project was almost abandoned because of bad customer relationship as some users in one of the key component of the project did not want to provide all the requirements and kept changing the requirements in order to frustrate the project. In order to address the problem of bad customer relationship, the developers engaged project managers and the project sponsors to resolve the personal relationships problems.

Process Factors

Process factors are another challenge for Agile methods. The process factors identified by four cases in this study were ill-defined project scope and project requirements; ill-defined management and customer role; and lack of Agile progress tracking mechanisms (Table 5). Respondents indicated that lack of Agile progress tracking mechanisms contributed to the delay in implementation of the projects. Respondents from cases 1, 2 and 6 reported that the project scope was not well defined and users kept on changing requirements and therefore the projects could not finish on time. Users from these projects kept on adding important requirements and yet when the project was delayed they blamed the development team. Respondents from cases 3, 4 and 6 argued that customer roles was not clear from the documentation provided by the organizations and therefore they had to identify the customers and define the roles themselves. However, even when roles were

defined it was difficult to implement as the customers were not available to the projects. Lack of customer presence have been identified in literature as a critical success factor in Agile software projects (Chow and Cao, 2008). Respondents from Cases 1, 4 and 6 pointed out that the issue lack of customer presence during the project implementation. It was noted for example in Cases 1 and 6 that some critical customers were not even informed about the projects and therefore they were reluctant to fully participate as they thought these were projects for the IT departments instead of the user departments.

The other process factors identified during the case study were lack of effective collaboration and multi-team coordination problems. Effective collaboration and involvement is necessary for success of Agile Methods therefore lack of effective collaboration can lead to project failure. Cases 1, 2 and 6 brought out lack of effective collaboration as a challenge because some users did not want participate due to lack of involvement at the scoping stage of the project and some IT staff did not want to be involved because of lack of confidence. Respondents from Cases 1, 2 and 6 brought out the issue multi-team coordination as the critical team necessary for integration of components of the project were struggling with other failure factors such as hostile customers and lack of executive sponsors.

Technical Factors

Technical factors were identified as leading challenges for Agile methods. The following technical factors were identified: lack of knowledge of correct Agile practices; inappropriate technology and tools; complex design; lack of documentation; lack of technical training; lack of software re-use; lack of or delayed integration testing (Table 6).

Table 5: Process factors

Identified factors	Case #					
	1	2	3	4	5	6
Ill-defined project scope and project requirements	X	X			X	X
Lack of effective collaboration	X	X				X
Ill-defined management and customer role	X		X	X		X
Lack of Agile progress tracking mechanisms	X	X	X	X		
Lack of customer presence	X			X		X
Multi-team coordination problems	X	X				X

Table 6: Technical factors

Identified factors	Case #					
	1	2	3	4	5	6
Lack of knowledge (correct Agile practices)		X	X			
Inappropriate technology and tools		X	X			
Complex design	X			X		X
Lack of documentation	X			X		X
Lack of technical training		X		X	X	
Lack of software re-use			X			
Lack of or delayed integration testing	X		X	X		X

Dikert *et al.* (2016) states that another common challenge of implementing Agile methods turned out to be technically difficult as an experienced software team may do well in training, but when the time comes to apply Agile techniques in practice, the team may get lost. Respondents from Cases 1, 4 and 6 argued that complex design was a major factor especially that object oriented design was used and the design was complex because of too many requirements. For example Case 1 and 6 required a registration database module that facilitates the registration and issuance of annual licenses to its clients and that this should be integrated to the compliance monitoring and inspection database module as well as accounting module (that is general ledger, cashbook, accounts receivable, accounts payable, invoicing, sales order processing and report writer). Furthermore, the accounting module needed to be integrated with the procurement and inventory module. Design of system meeting requirement of different professional users was complex. In addition, communicating such a design to the developers and users was taunting task.

Lack of documentation was brought out by respondents from Cases 1, 2, 4 and 6. They indicated that lack of or minimum documentation was vital challenge they faced as development teams as this led to communication lapses and misunderstandings. Respondents from Case 2 and 4 pointed out that lack of documentation had more adverse impact on large projects where traceability was an important project requirements i.e. validating whether the delivered products met the initial requirements. The case study findings are supported by Gregory *et al.* (2016) who argue that documentation overhead demanded by management takes additional time for the development team and undermines the Agile approach and indicates a lack of trust.

The challenge of complex design explained the lack of or delayed integration testing in Cases 1 and 6. They indicated that lack of integration testing was discovered during training sessions, for example an accounting system in case 1 and 6 depended on the registration module and delayed implementation of the registration module had significant impact on the success of the accounting system. Lack of or delayed integration testing was brought out by respondents from four cases (Cases 1, 3, 4 and 6). This challenge was also identified by Bjarnason *et al.* (2016) who reported that complex interactions and dependencies between requirements, such as user interfaces and quality requirements, are a challenge to capture with acceptance test cases.

Project Management Factors

The following project management factors were identified by five cases lack of functional Project Board

and support by the sponsors; lack of Project Quality Assurance; Project with fixed cost; and project cost under budgeted (Table 7). The other process factors identified during the case study were ill-defined project planning. Cases 2 and 5 indicated that in Agile methodology the project planning is important but the process did not include customers and it was discovered that some important customer went on leave at critical time of project implementation and therefore little progress was made on the projects. It was noted for example in Cases 3 and 4 that some critical customers were not even informed about the projects during scoping and therefore they were reluctant to fully participate as they thought these were IT projects rather than for user departments.

Resistance to change has been identified in literature as a challenge to Agile projects because people are not willing to change unless there are good reasons that they understand and the change is perceived easy enough (Gandomani and Nafchi, 2016; Dikert *et al.*, 2016). In order to address resistance to change most projects will constitute project management team to manage change. Lack of project management team was brought out by Cases 1, 3 and 6. They indicated that some customer did not fully accept and recognize the Agile principles and even the IT project itself as they did not see individual benefit and feared that the project would disturb the balance of power within the organisations. Other factors identified by this case study were project with rapid changing requirements and project with multiple independent teams.

Factors Unique to Zambia, a Developing Country

Zambia is a landlocked country and faces a number of challenges faced by landlocked countries such as (1) dependence on transit infrastructure; (2) dependence on political relations with neighbours; (3) dependence on peace and stability within transit neighbours; and (4) dependence on administrative processes in transit (Faye *et al.*, 2004). The Zambian economy has historically been heavily dependent on copper mining and from early 1970s there has been a significant decline in copper prices (Shafika, 2007). This led to the closure of mines and slow down on the economy. Slow progress in diversifying the economy and high levels of borrowing are also contributing factors to the country's economic malaise. Zambia is still one of the low income countries as it can be seen from the socio-economic indicators in the Table 8. According the 2013 statistics, Zambia suffers a high rate of HIV/AIDS, with 12.5% of Zambians age 15 to 49 years being HIV positive (UNAIDS, 2015). This has a negative impact on the availability of national ICT skills because people with ICT expertise either die during the prime age or spend most of their time at the hospital thus robbing the country of the required technical skills.

Table 7: Project factors

Identified factors	Case #					
	1	2	3	4	5	6
Lack of functional Project Board and support from sponsors	X	X	X	X		X
Ill-defined project planning		X	X	X	X	
Lack of project Quality Assurance	X	X		X	X	X
Project with rapid changing requirements		X				X
Project with fixed cost	X	X		X	X	X
Project cost under budgeted	X	X		X	X	X
Project with multiple independent teams			X			X
Lack of Change Management team	X		X	X		X

Table 8: Zambia Economic Indicators (World Bank, 2015)

Indicators	
Population	14.54 million (2013)
Human Development Index (HDI)	0.561 (2013)
GDP	\$26.82 billion (2013) Annual GDP growth rate 6.4% (2014)
Poverty Headcount ratio (% of population)	60.5%
Per capita Gross National Income (GNI)	\$1,810 (2013)

Table 9: Developing countries factors

Identified factors	Case #					
	1	2	3	4	5	6
Delay in the approval process	X	X	X	X		
Lack of resources and/or delay in payment	X	X		X		X
Lack of technical skills	X	X		X		X
Lack of latest ICT tools and use of obsolete technology	X	X		X	X	X

It was discovered within data analysis process Zambia being a developing countries had the following challenges: delay in the approval process; lack of resources and/or delay in payment; lack of technical skills; lack of latest tools and use of obsolete technology (Table 9). Four cases (Cases 1, 2, 4 and 6) indicated that lack of technical skills was a major failure factor and this was demonstrated by IT staff from these institutions failing to do basic installation and configuration of Windows web servers. Furthermore, respondents indicated that these organisations have not been able to recruit qualified staff either because they are not available and when they are available they are expensive to maintain. Lack of latest ICT tools and use of obsolete technology was brought out by five cases, for example respondents from case 1 and 6 indicated that use of obsolete Windows operating system was a major security threat to the success of the projects and use of old servers had a negative impact on the performance of the web application.

Cases 1, 2, 4 and 6 indicated that because of lack of resources in these institutions there were delays in approval of project deliverables. Furthermore they indicated that there delay in payment after project deliverable sign off by the customers because of lack of resources. Respondents proposed that organizations in Zambia should consider using open source software and cloud computing to reduce on the cost of investing in IT infrastructure and then the savings can be channeled to projects utilizing Agile methods.

Measures that Promote Success of Agile Methods

Measures to promote success of Agile methods have been proposed in literature. For example, Kaijo-Mattsson (2008) proposes that to remedy the communication problem, the organizations should create a Communication Owner role, a role responsible for communicating the overall system and process information. Ensure management support and make management support visible will also promote success of Agile methods (Dikert *et al.*, 2016). The existence of this role however does not guarantee that the valuable information gets documented. This measure is new with respect to traditional environment. This case study brought the importance of having a project board comprising of representative from customer and developers (see Table 10). Other measures identified by the case study were involvement of collaborative customers and use of communication tools e.g. Skype, team viewer.

Respondents from Cases 1, 2, 4, 5 and 6 indicated that whenever a meeting involving the Chief Executive Officer of the customer institution and Director of the developer institution was convened a lot of progress was made in terms of unlocking things that stalled during project implementation. Therefore this suggests that having a project board comprising of representative from customer and developers at senior level promotes success of Agile methods.

Table 10: Measure of success factors

Identified factors	Case #					
	1	2	3	4	5	6
Involvement of collaborative customers	X	X				
Having a project board comprising of representative from customer and developers	X	X		X	X	X
Use of communication tools e.g. Skype, team viewer, whatsapp	X					X

Conclusion

Lessons Learnt

Although Agile software development provides potential for evolutionary software development, early software delivery and addresses resistance to change yet it has a number of challenges. This study reports the failure factors and challenges experienced by developers and implementer of Agile methodology in Zambia, a developing country (Table 3 to Table 9). The four dimension failure factors proposed by (Chow and Cao, 2008) proved useful in identifying and classifying the important failure factors for Agile methodology. Two additional dimensions namely project management dimension and developing country unique problems dimension were helpful in understanding project management issues and eliciting failure factors that are specific to developing countries like Zambia. This study used grounded theory and therefore its findings are grounded in the substantive data and conclusion can be drawn from these findings. Grounded theory approach is useful for wide range of research studies in context of software engineering, information system and in Agile context in particular (Hoda *et al.*, 2011; Gandomani and Zulzalil, 2013; Gandomani and Nafchi, 2016). However, the findings from this study are not applicable to all developing countries because this study was limited Zambia and to those who voluntarily participated in this research. Therefore, it cannot be directly generalized to other developing countries because of the diversity of these countries in terms of size of country, whether landlocked or not, endowments of natural resources, nature of their incomes and levels of per capita national incomes.

The theory building or conceptualization led to learning many lessons, for example, the study shows that there are challenges that are unique to developing countries and therefore measures should be developed to address these unique problems when implementing Agile projects in developing countries. The other lessons learned from this case study include:

- Some challenges are project management issues and therefore are not unique to Agile methods, for

example lack of functional Project Board. However these project management issues appear to adversely affect Agile methods compared to traditional methods because Agile methods advocates for quick delivery of the products while applying a human-centered approach

- Some challenges are specific to developing countries, for example lack of resources. In developing countries like Zambia where resources are scarce contract should provide penalties for delayed payment and organisation should consider using open source software and cloud computing to minimize costs of IT infrastructure
- Agile methods are not suitable for projects with constrained budgetary allocations, uninformed end users as well as for institutions with business processes that are determined externally. This can be overcome by increasing communication ties within and across stakeholder groups
- Agile methods are suitable for projects where all parties have a cordial working relationship otherwise disgruntled users may frustrate developers during implementation or lead a good product not be used by users. Therefore project manager must pay a lot attention to people issues when using Agile methods for example they can assign experienced and professional coaches in their teams

Future work will involve undertaking a comparative studies involving Zambia and other developing countries in the Sub-Saharan Africa in order to share experiences and challenges in the adoption of the Agile software development methodologies. It is proposed that the Technology Acceptance Model (TAM) be used for future research to undertake the comparative study. TAM Model, has been widely accepted and recommended by researchers as one of the models that helps to predict the behaviour of users with regard to the intention and subsequent adoption of information systems (Al-Sharafi *et al.*, 2016). Therefore the use of the TAM would prove useful in eliciting experiences and perceptions towards adoption of Agile software development methodologies.

Acknowledgement

The authors would like to thank Mulungushi University Management for providing financial support to this research through consultancy funds. The authors also wish to thank the management of the organisations where this study were conducted.

Author's Contributions

Douglas Kunda: Proposed the research framework and implemented it to get the result. Analyzed the results and writing this paper for publication purpose.

Mwenge Mulenga: Co-writing, validation of Agile software development failure factors and editing the literature review.

Muwanei Sinyinda: Co-writing, validation of Agile software development failure factors and editing the literature review.

Victoria Chama: Co-writing and editing the literature review.

Ethics

The authors confirm that this manuscript has not been published elsewhere and that no ethical issues are involved.

References

- Abrahamson, P., N. Oza and M.T. Siponen, 2001. Agile software development methods. VTT Publications.
- Agrawal, A., M. Aurangzeb Atiq and L.S. Maurya, 2016. A current study on the limitations of Agile methods in industry using secure Google forms. *Procedia Comput. Sci.*, 78: 291-297. DOI: 10.1016/j.procs.2016.02.056
- Al-Sharafi, M., R. Arshah, E. Abu-Shanab, M. Fakhreldin and N. Elayah, 2016. The effect of security and privacy perceptions on customers' trust to accept internet banking services: An extension of TAM. *COMSCET 2016*, pp.1-9.
- Baskerville, R., Pries-Heje, J., & Madsen, S. 2011. Post-agility: What follows a decade of agility? *Inform. Software Technol.*, 53: 543-555. DOI: 10.1016/j.infsof.2010.10.010
- Berger, C. and U. Eklund, 2015. Expectations and challenges from scaling Agile in mechatronics-driven companies-a comparative case study. *Proceedings of the International Conference on Agile Software Development Agile Processes*, in *Software Engineering and Extreme Programming*, pp: 15-26. DOI: 10.1007/978-3-319-18612-2_2
- Bjarnason, E., M. Unterkalmsteiner, M. Borg and E. Engström, 2016. A multi-case study of Agile requirements engineering and the use of test cases as requirements. *Inform. Software Technol.*, 77: 61-79. DOI: 10.1016/j.infsof.2016.03.008
- Boehm, B. and R. Turner, 2005. Management challenges to implementing Agile processes in traditional development organizations. *IEEE Software*, 5: 30-39. DOI: 10.1109/MS.2005.129
- Boehm, B., 2002. Get ready for Agile methods, with care. *IEEE Comput.*, 35: 64-69. DOI: 10.1109/2.976920
- Ceschi, M., A. Sillitti and G. Succi, 2005. Project management in plan-based and Agile companies. *IEEE Software*, 22: 21-27. DOI: 10.1109/MS.2005.75
- Chow, T. and D. Cao, 2008. A survey study of critical success factors in Agile software. *J. Systems Software*, 81: 961-971. DOI: 10.1016/j.jss.2007.08.020
- Coleman, G. and R. O'Connor, 2007. Using grounded theory to understand software process improvement: A study of Irish software product companies. *Inform. Software Technol.*, 49: 654-667. DOI: 10.1016/j.infsof.2007.02.011
- Collier, K.W., 2011. *Agile analytics: A value-driven approach to business intelligence and data warehousing*. Addison-Wesley Professional.
- Conboy, K., S. Coyle, X. Wang and M. Pikkarainen, 2011. People over process: Key challenges in Agile development. *IEEE Software*, 28: 48-57. DOI: 10.1109/MS.2010.132
- Conforto, E.L. and D.C. Amaral, 2016. Agile project management and stage-gate model-A hybrid framework for technology-based companies. *J. Eng. Technology Management*, 40: 1-14. DOI: 10.1016/j.jengtecman.2016.02.003
- Cordeiro, L., R. Barreto, R. Barcelos, M. Oliveira and V. Lucena *et al.*, 2007. Agile development methodology for embedded systems: a platform-based design approach. *14th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems*, March, 26-29, IEEE Xplore press, Tucson. DOI: 10.1109/ECBS.2007.16
- Dikert, K., M. Paasivaara and C. Lassenius, 2016. Challenges and success factors for large-scale Agile transformations: A systematic literature review. *J. Systems Software*, 119: 87-108. DOI: 10.1016/j.jss.2016.06.013
- Dingsøy, T., S. Nerur, V. Baliyepally and N.B. Moe, 2012. A decade of Agile methodologies: Towards explaining Agile software development. *J. Systems Software*, 85: 1213-1221. DOI: 10.1016/j.jss.2012.02.033

- Dyba, T., 2000. Improvisation in small software organizations. *IEEE Software*, 17: 82–87.
DOI: 10.1109/52.877872
- Faye, M.L., J.W. McArthur, J.D. Sachse and T. Snowd, 2004. The challenges facing landlocked developing countries. *J. Human Development*, 5: 31-68.
DOI: 10.1080/14649880310001660201
- Gandomani, T.J. and M.Z. Nafchi, 2016. Gile transition and adoption human-related challenges and issues: A Grounded Theory approach. *Comput. Hum. Behav.*, 62: 257-266.
DOI: 10.1016/j.chb.2016.04.009
- Gandomani, T.J., G. Zulzalil, A.A. Ghani, A.B. Sultan and M.Z. Nafchi, 2013. Obstacles in moving to Agile software development methods: At a glance. *J. Comput. Sci.* 9: 620–625.
DOI: 10.3844/jcssp.2013.620.625
- Glazer, H., 2001. Dispelling the process myth: Having a process does not mean sacrificing agility or creativity. Entinex, Inc.
- Gregory, P., L. Barroca, H. Sharp, A. Deshpande and K. Taylor, 2016. The challenges that challenge: Engaging with Agile practitioners' concerns. *Information Software Technol.*, 77: 92-104.
DOI: 10.1016/j.infsof.2016.04.006
- Gregory, P., L. Barroca, K. Taylor and D. Salah, 2015. Agile challenges in practice: A thematic analysis. *Proceedings of the International Conference on Agile Software Development XP 2015: Agile Processes in Software Engineering and Extreme Programming* pp: 64-80.
DOI: 10.1007/978-3-319-18612-2_6
- Highsmith, J. and A. Cockburn, 2001. Agile software development: The people factor. *IEEE Comput.*, 34: 131-133. DOI: 10.1109/2.963450
- Hilkka, M.R., T. Tuure and R. Matti, 2005. Is extreme programming just old wine in new bottles: A comparison of two cases. *J. Database Management*, 16: 41-61. DOI: 10.4018/jdm.2005100103
- Hoda, R., J. Noble and S. Marshall, 2011. Developing a grounded theory to explain the practices of self-organizing Agile teams. *Empirical Software Eng.*, 17: 609-639. DOI: 10.1007/s10664-011-9161-0
- Inayat, I., S.S. Salim, S. Marczak, M. Daneva and S. Shamshirband, 2015. A systematic literature review on Agile requirements engineering practices and challenges. *Comput. Hum. Behav.*, 51: 915-929.
DOI: 10.1016/j.chb.2014.10.046
- Kajko-Mattsson, M., 2008. Problems in Agile trenches. *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Oct. 9-10, ACM, Kaiserslautern.
- Karlstrom, D. and P. Runeson, 2005. Combining Agile methods with stage-gate project management. *IEEE Software*, 22: 43-49. DOI: 10.1109/MS.2005.59
- Kunda, D and Brooks, L., 2000. Assessing organisational obstacles to component-based development: a case study approach. *Information and Software Technology*, 42: 715–725
- Mcavoy, J. and T. Butler, 2007. The impact of the Abilene Paradox on Doubleloop learning in an Agile team. *Inform. Software Technol.*, 49: 552–563.
DOI: 10.1016/j.infsof.2007.02.012
- Nerur, S., R. Mahapatra and G. Mangalaraj, 2005. Challenges of migrating to Agile methodologies. *Communications ACM*, 48: 72-78.
DOI: 10.1145/1060710.1060712
- Ohno, T., 1988. Toyota production system: Beyond large-scale production. Productivity Press.
- Pagano, D., & Bruegge, B. 2013. User Involvement in Software Evolution Practice: A Case Study. *Proceedings of the International Conference on Software Engineering*, San Francisco, pp: 953-962.
DOI: 10.1109/ICSE.2013.6606645
- Poppendieck, T. and M. Poppendieck, 2003. Lean software development: An Agile toolkit for software development managers. Boston: Addison-Wesley.
- Ramesh, B., R. Baskerville and L. Cao, 2010. Agile requirements engineering practices and challenges: An empirical study. *Information System J.*
- Runeson, P. and M. Höst, 2008. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Eng.*, 14: 131-164. DOI: 10.1007/s10664-008-9102-8
- Schatz, B. and I. Abdelshafi, 2005. Primavera gets Agile: A successful transition to Agile development. *IEEE Software*, 22: 36-42.
DOI: 10.1109/MS.2005.74
- Shafika, I., 2007. Survey of ICT and education in Africa: Zambia Country Report.
- Stephens, M. and D. Rosenberg, 2003. Extreme programming refactored: The case against XP. Berkeley: Apress.
- Strauss, A. and J. Corbin, 1990. Basics of qualitative research: Grounded theory procedures and techniques. London: Sage Publications.
- Svensson, H. and M. Host, 2005. Views from an organization on how Agile development affects its collaboration with a software development team. *Proceedings of the International Conference on Product Focused Software Process Improvement Product Focused Software Process Improvement*, Berlin: Springer Verlag, pp: 487-501.
- Tessem, B., 2003. Experiences in learning XP practices: A qualitative study. *International Conference on Extreme Programming and Agile Processes in Software Engineering: Extreme Programming and Agile Processes in Software Engineering*, Berlin: Springer Verlag, pp: 131-137.
DOI: 10.1007/3-540-44870-5_17

- Vijayarathy, L.R. and D. Turk, 2008. Agile software development: A survey of early. *J. Inform. Technol. Management*, 19: 1-8.
- Waardenburg, V.G. and V.H. Vliet, 2013. When Agile meets the Enterprise. *Inform. Software Technol.*, 55: 2154-2171. DOI: 10.1016/j.infsof.2013.07.012
- Williams, L. and A. Cockburn, 2003. Agile software development: it's about feedback and change. *IEEE Comput.*, 36: 39-43.
DOI: 10.1109/MC.2003.1204373
- World Bank. 2015. Country data sheets. Zambia.
- Yin, R.K., 1994. *Case study research: Design and Methods*. London: Sage Publications.