

Waterfall and Agile information system project success rates—A South African perspective

Lucas Khoza , Carl Marnewick 

Department of Applied Information Systems, College of Business and Economics, University of Johannesburg, South Africa

ABSTRACT

Even though software projects do add value to the organisation, studies reveal that some software projects are still failing at an alarming rate and do not always provide the anticipated value to the organisation. This has been the case for the last couple of decades. Software projects use predominantly Waterfall as a methodology. This raises the question whether new ways of working can be introduced to improve the success rate. One such new way is Agile as an approach to developing software. A survey was done to determine whether Agile projects are more successful than Waterfall projects, thus contrasting the old and the new ways of working. Some 617 software projects were evaluated to determine the success rate based on the methodology used. Success was measured on a continuum of five levels and not just the triple constraint. The results imply that Agile projects are more successful than Waterfall projects to some extent, but that there are still concerns that need to be addressed.

Keywords: Agile, waterfall, software projects, SDLC, success

Categories: • Software and its engineering ~ Agile software development

Email:

Lucas Khoza lucask@uj.ac.za (CORRESPONDING),
Carl Marnewick cmarnewick@uj.ac.za (CORRESPONDING)

Article history:

Received: 16 Feb 2019
Accepted: 4 Dec 2019
Available online: 20 Jul 2020

1 INTRODUCTION

Opposites. Life consists of opposites, such as day-night, north-south, male-female. Another opposite in the software industry is Agile and Waterfall. These two methodologies, which are used to develop software, directly oppose each other. Like all opposites, people tend to take sides but which side is better when it comes to software project success? Anecdotal evidence shifts the scale towards Agile software development, but what is the truth?

IS or IT project success has been debated for years. Reports such as the Chaos Chronicles and the Prosperus reports indicate that software project success is still below acceptable norms (Marnewick, 2013; The Standish Group, 2014). Different sources provide different statistics

Khoza, L. and Marnewick, C. (2020). Waterfall and Agile information system project success rates—A South African perspective. *South African Computer Journal* 32(1), 43–73. <https://doi.org/10.18489/sacj.v32i1.683>
Copyright © the author(s); published under a [Creative Commons NonCommercial 4.0 License \(CC BY-NC 4.0\)](https://creativecommons.org/licenses/by-nc/4.0/).
SACJ is a publication of the South African Institute of Computer Scientists and Information Technologists. ISSN 1015-7999 (print) ISSN 2313-7835 (online).

on software project success but the average success rate of software projects is about 30–40% depending on the research (Marnewick, 2013; Marnewick, Erasmus & Joseph, 2017; The Standish Group, 2013, 2014). Since software projects are used to implement the organisation's IT strategies, the question is how successful the implementation of such strategies can be. The implication is that 60–70% of strategies are not fully implemented and realised, which is a major cause for concern (Marnewick, 2013; Marnewick et al., 2017).

The latest results indicate that software projects incorporating Agile principles tend to be more successful than those following the Waterfall method (The Standish Group, 2014; VersionOne Inc., 2016, 2017). The concern is that project success is still measured based on the triple constraint of time, cost and scope. This way of measurement has been scrutinised and new models or frameworks are used to measure project success. These new models and frameworks emerged as the measuring of project success matured. One such framework measures IT project success on a continuum of five levels (Bannerman, 2008; Bannerman & Thorogood, 2012). Previous studies have highlighted various factors that contribute to software project success. One of these factors is the type of methodology used. Although the methodology is highlighted, little evidence really exists whether Agile as a methodology is more successful than the Waterfall methodology. The few studies that did such an exercise focused on the triple constraint as a measurement of success. For example, Serrador and Pinto (2015) focused on the triple constraint and stakeholder management, the Chaos Chronicles focused on the triple constraint (The Standish Group, 2013, 2014) and Kisielnicki and Misiak (2017) concluded that for business intelligence projects, there is a definite benefit to using Agile as a methodology. The research discussed in this article had two purposes. The first was to determine whether there really is a difference between Agile and Waterfall projects when it comes to software project success. The second was to measure the success of these two types of projects based on a continuum.

Questionnaires were used to determine the success of software projects. Success was measured based on five levels (Bannerman, 2008; Bannerman & Thorogood, 2012). A distinction was made between the type of project (Agile or Waterfall). This was then used to differentiate between the success rates of software projects that incorporate either Agile or Waterfall as a methodology.

This article is structured as follows: section 2 provides in-depth literature review focusing on traditional software development methodologies, Agile methodologies, comparisons of Agile and traditional methodologies, software project success and software project success rates. Section 3 is the research methodology, the detailed presentation and the analysis of the results is presented in section 4. Lastly, section 5 concludes with the detailed comparison of success criteria of Agile and Waterfall projects.

2 LITERATURE REVIEW

Software project success is highly dependent on the software development methodology used (Marnewick & Langerman, 2018; The Standish Group, 2013, 2018). Every software project

has some component of software development. It might be integrations, customisations or even a totally new system. It is therefore imperative that the two sides of the coin (software project success and software methodologies) be discussed and compared to determine what the underlying philosophical differences are between Waterfall and Agile methodologies.

2.1 Traditional software development methodologies

The software development life cycle (SDLC) is a methodology for designing and developing software products (Alshamrani & Bahattab, 2015). SDLC models have limitations, advantages and disadvantages and a series of phases that are followed when designing software (Alshamrani & Bahattab, 2015; Kumar, Zadgaonkar & Shukla, 2013). As a number of methodologies have evolved to meet the new demands of the 21st century, traditional SDLC methodologies no longer meet the requirements of the current market (Mahalakshmi & Sundararajan, 2013). The Waterfall methodology, Rational Unified Process (RUP) and the V-model are classified as the heavyweight methodologies under the traditional software development methodologies. The traditional methodologies follow a series of steps when developing a software, that is, requirements must be defined properly, then the product is built and tested and it then goes through the deployment process. The success of a software developed using traditional methodologies relies heavily on well-defined requirements before the team begins the development. Table 1 highlights some of the drawbacks experienced when using traditional SDLC.

Agile methodologies were introduced to solve some challenges around the traditional development methodologies as highlighted in Table 1.

2.2 Agile methodologies

Agile is an incremental and iterative approach to software development that is effective in delivering high-quality products and also satisfying customers' expectations (Ghani & Bello, 2015). Popular Agile software development methodologies include Scrum, Feature Driven Development, Kanban, Extreme Programming, Adaptive Software Development, Crystal, Pragmatic Programming and Dynamic System Development (Ghani & Bello, 2015; Matharu, Mishra, Singh & Upadhyay, 2015). As organisations invest in software projects and have the zeal to improve the delivery and the success of their projects, it is vital for them to use the correct methodology that can result in the speedy delivery of the project's deliverables (Clara, 2013; Matharu et al., 2015). Traditional methodologies such as Waterfall allow requirements to be defined during the initial stages of the SDLC without giving customers the opportunity to review deliverables until the end of the project and this has led to the failure of many software projects (Farlik, 2016; Matharu et al., 2015). Due to the complexities of software projects and unclear requirements upfront, users get to understand what they actually want only as the project progresses in its SDLC (Clara, 2013; Farlik, 2016; Mahalakshmi & Sundararajan, 2013). Customers fail to visualise the end product at the beginning until they see it and this is why customers' requirements keep on changing during every stage of the SDLC (Farlik, 2016;

Table 1: Drawbacks of traditional SDLC

Drawbacks of traditional SDLC	Author
Unclear understanding of the requirements	Clara (2013), Mahalakshmi and Sundararajan (2013)
Problems of each phase are not solved completely and result in problems in the next phase	Mahalakshmi and Sundararajan (2013)
Scope freezing	Mahalakshmi and Sundararajan (2013)
Delivery schedule of a project is projected/envisioned	Kumar, Zadgaonkar and Shukla (2013), Mahalakshmi and Sundararajan (2013)
More complicated documentation that takes too much time	Bindal and Mehta (2015), Mahalakshmi and Sundararajan (2013)
Difficulty in responding to change	Bindal and Mehta (2015), Kumar, Zadgaonkar and Shukla (2013), Mahalakshmi and Sundararajan (2013)
Project cost determined during planning	Mahalakshmi and Sundararajan (2013)
Team has limitations to being creative and flexible	Bindal and Mehta (2015), Kumar, Zadgaonkar and Shukla (2013), Mahalakshmi and Sundararajan (2013)
Approval sought at the end	Arora and Arora (2016)

Mahalakshmi & Sundararajan, 2013; Matharu et al., 2015). Flexible methodologies that allow for change in scope of work or requirements are necessary in software projects and Agile methodologies are a possible solution (Farlik, 2016; Matharu et al., 2015). Research indicates that Agile methodologies improve the quality of projects, satisfy customers and are more reliable in supporting changes and complexities in software projects (Ghani & Bello, 2015). These benefits of Agile methodologies lead to the success of software projects. Figure 1 summarises the benefits of Agile methodologies.

The results indicate that organisations are harvesting the benefits of introducing Agile methodologies into the organisation. The benefits are spanning the technical side of projects, the team component of the project as well as the organisational level. It is also evident from the results that the initial benefits are much higher but once Agile methodologies are embedded into the organisation, the benefits are not as drastic as with the initial introduction.

Agile methodologies were introduced to improve software project success and increase understanding of system requirements. Therefore, if organisations can find a way to control the knowledge required in order to improve project success, this can have a positive impact on competitive advantage in the economy (Heikkilä, Damian, Lassenius & Paasivaara, 2015).

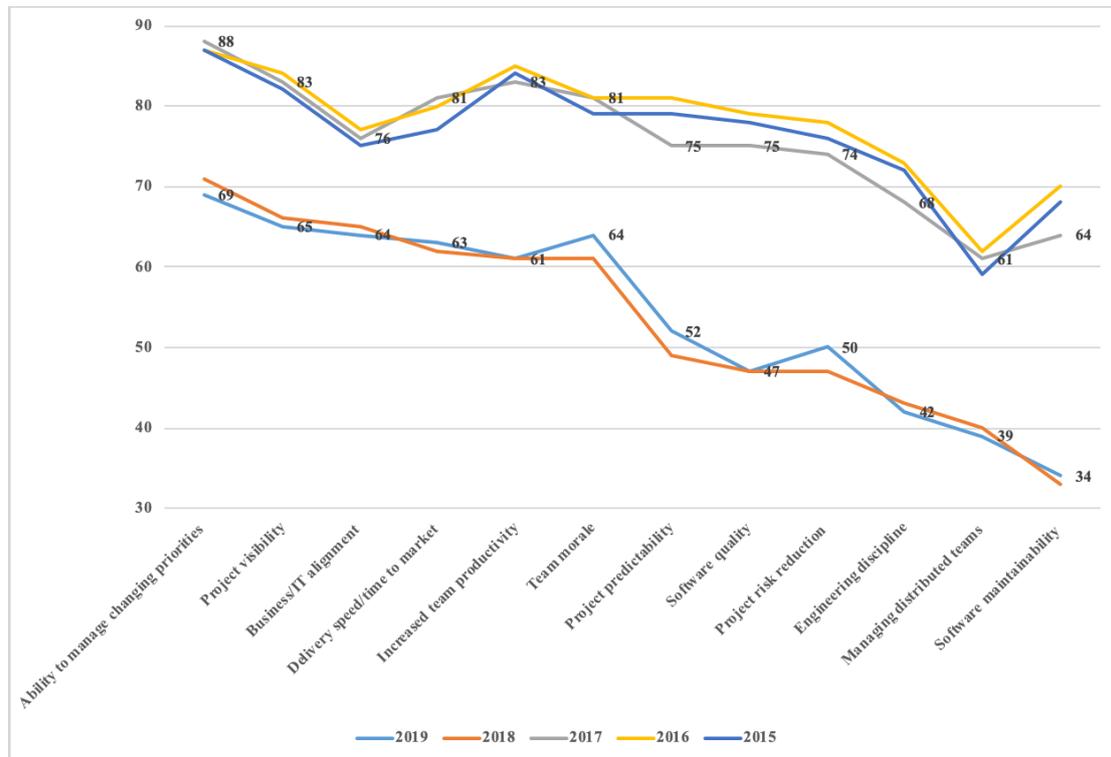


Figure 1: Benefits of Agile methodologies (Matharu, Mishra, Singh & Upadhyay, 2015; VersionOne Inc., 2015, 2016, 2017, 2018, 2019)

2.3 Comparing Agile and traditional methodologies

Traditional and Agile methodologies differ in certain parameters (Matharu et al., 2015). Table 2 compares these methodologies using different parameters.

The next section provides a brief overview of project success and how it is viewed nowadays by scholars and practitioners.

2.4 Project success

Project success is a topic that is still being researched and is still one of the major schools of thought within the project management discipline (Silvius, 2017). This can be attributed to the fact that the measurement of project success has evolved from the original triple constraint, i.e. time, cost and scope, to strategic alignment and benefits realisation. Project success is defined differently by the various project management standards. The Project Management Institute measures success based on quality, timeliness, customer satisfaction and budget compliance (Project Management Institute, 2017). The Association for Project Management only uses customer satisfaction as a success criterion (Association for Project Management, 2006). The

Table 2: Comparison of traditional and Agile methodologies (Ahimbisibwe, Cavana & Daellenbach, 2015; Matharu, Mishra, Singh & Upadhyay, 2015)

Parameter	Traditional methodologies	Agile methodologies
Adaptability to change	Change sustainability	Change adaptability
Development approach	Predictive	Adaptive
Development orientation	Process oriented	People oriented
Project size	Large	Small/medium
Planning scale	Long term	Short term
Management style	Command and control	Leadership and collaboration
Learning	Continuous learning during development	Learning is secondary to development
Documentation	High	Low
Development team	Plan-oriented Adequate skills Access to external knowledge	Agile Knowledgeable Co-located and collaborative
Customers	Pre-structured teams Minimal commitment Not co-located Not empowered	Self-organising teams Dedicated Knowledgeable Co-located Collaborative Representative and empowered
Requirements	Known early Largely stable	Largely emergent Rapid change
Architecture	Designed for current requirements	Designed for current and foreseeable requirements
Fundamental assumption	Systems are fully specifiable, predictable and built through meticulous and extensive planning	High-quality adaptive software developed based on principles of continuous design improvement and testing based on rapid feedback and change

Project Management Association of Japan focuses on novelty, differentiation and innovation as success criteria (Ohara, 2005).

Marnewick (2012) analysed the various project success criteria and concluded that success is measured across various criteria ranging from the iron triangle, realization of strategic objectives, business success, end-user's satisfaction as well as benefits to stakeholders. Bannerman (2008) and Bannerman and Thorogood (2012) incorporated all the various success criteria into one framework focusing specifically on IT project success. For this reason, Bannerman's five levels of measuring project success is used in this study as it covers all levels used to measure project success. Bannerman (2008) and Bannerman and Thorogood (2012) posit that project success should be measured across five levels:

Level 1 (Process): This level's success is measured based on the discipline-specific technical and management processes that are used to achieve the project objectives.

Level 2 (Project management): Success is measured at this level based on the triple constraint.

Level 3 (Product): The success of the product itself is determined at this level. Measurement criteria include whether specifications and requirements have been met or whether the users have accepted and are using the final product.

Level 4 (Business): Success is determined based on the realisation of the promised benefits and whether business objectives have been met.

Level 5 (Strategic): Ultimately, organisations are there to make a profit and be sustainable. The success of a project and its subsequent product should contribute to the strategic success of the organisation.

Project success is unfortunately still measured based on the triple constraint or a variation of it by industry (The Standish Group, 2013, 2014). However, there is a definite shift away from the traditional triple constraint to a more strategic view of project success.

It is clear that researchers focus more on the negative side of project success in order to find a solution on how to increase the success rate of software project success. Sauer, Gemino and Reich (2007) as well as Glass (2006) argue that even though projects are still failing or challenged (32%), there are more projects that are performing well (67%) and deliver value to the organisation. There is an improvement in software project success due to the change on how software projects are measured and therefore, there is value in software projects but not always as expected (Marnewick, 2012).

The next section briefly deals with the success rates of software projects. The results are retrieved from two longitudinal studies, i.e. the Chaos Chronicles (The Standish Group, 2013, 2014) and the Prosperus reports (Marnewick, 2013; Sonnekus & Labuschagne, 2003).

2.5 Software project success rates

Software project success has been studied over the last few decades by academics and practitioners alike. The reason for this almost frenetic research is that there is enough evidence to indicate that software projects are still failing at an alarming rate. Results from international research are shown in Figure 2.

The results clearly indicate that something is drastically wrong as the rate to successfully implement a software project is on average 28%. Of concern is the fact that the success rates have stagnated at about 30% for the last decade, implying three things:

- software departments and professionals actually do not care about these results,
- we do not understand the complexity of software projects, and
- we are measuring the success of software projects incorrectly.

Software project success rates within South Africa do not look much better, as indicated in Figure 3.

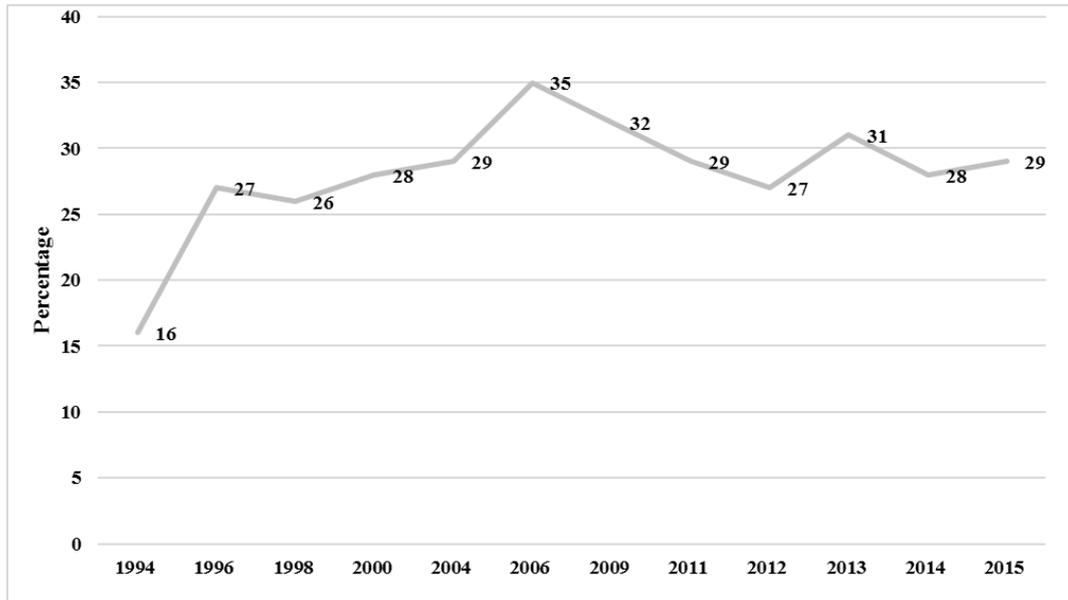


Figure 2: International software project success rates (The Standish Group, 2013, 2014)

Although the South African success rates look better than those internationally, South African companies compare well with international companies where on average 47% of software projects are perceived as successful.

The majority of the research into software project success does not distinguish between Waterfall and Agile software projects. However, recent studies distinguish between these types of projects. Results from the 2015 Chaos Chronicles indicate that software projects using Agile as a development methodology are more successful than those that still follow the traditional ways of developing software (The Standish Group, 2018). Agile projects are 28% more successful than traditional software development projects. In a limited study focusing only on Business Intelligence (BI) projects, an improvement was seen when BI projects were implemented using Agile (Kisielnicki & Misiak, 2017).

Table 3 presents contemporary research on the success rates of Agile versus Waterfall projects.

These studies, although limited in their scope, provide some evidence that Agile software projects are and should be more successful than software projects employing Waterfall as a methodology. The shortcoming of these studies is that they are limited in the way that success is measured. Success is measured either in the traditional way of successful, challenged or failed, or in the case of Kisielnicki and Misiak (2017), it is directed more towards the processes and value creation. Another shortcoming is that previous studies were done in developed countries and not in a developing country such as South Africa.

This led to the research question for this study:

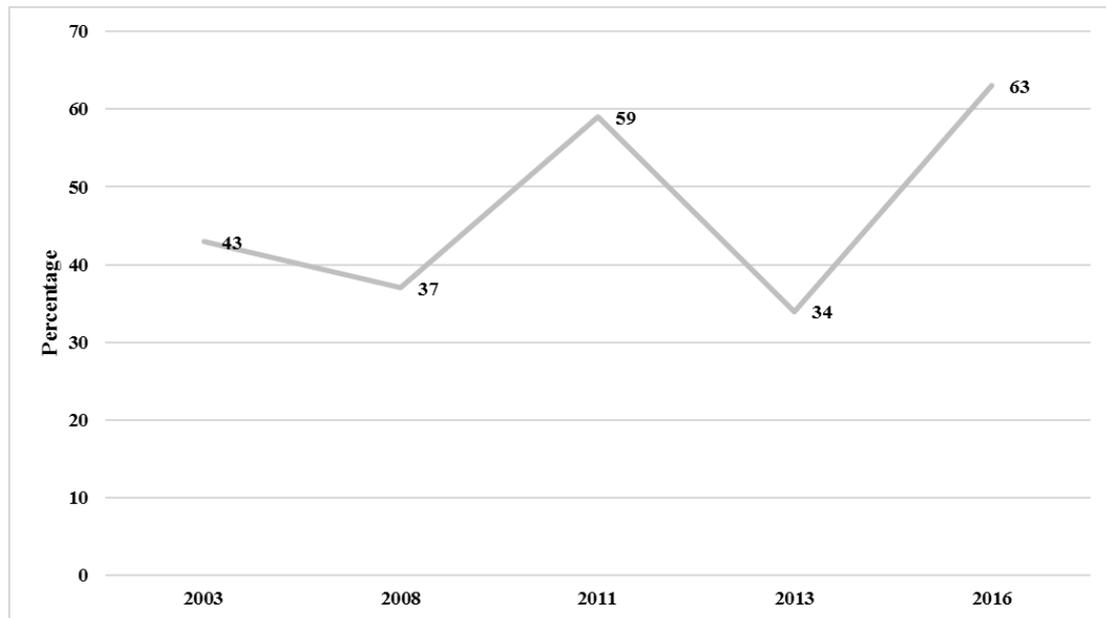


Figure 3: South African software project success rates (Labuschagne & Marnewick, 2009; Marnewick, 2013; Marnewick, Erasmus & Joseph, 2017)

Are software projects using Agile principles more successful than those that follow the traditional method of software development?

This question was answered based on the five levels of project success as per the Bannerman framework (2008, 2012). Measuring success across a continuum provides insight into the effectiveness and efficiency of either Waterfall or Agile methodologies. The success of a project is not just measured at a single point in time, but across the entire life cycle of the project, including the project deliverable, i.e. the product. This alternative way of measuring software project success is the value contribution of this article.

3 RESEARCH METHODOLOGY

The research followed the route of a quantitative approach and questionnaires were used to determine the success rates of software projects (Denscombe, 2007; Thomas, 2013). The unit of analysis was a software project and the respondents were asked to determine the success of the project based on five levels of success. The questionnaire was divided into four sections. The first section focused on the respondents' biographical information. This information can be used to determine whether biographical information have an impact on software project success. Most of the respondents were male (154) and 83 females. A third of the respondents were in their thirties with 23.2% in their twenties. Twenty percent of the respondents were in

Table 3: Research comparing Waterfall and Agile methods

Study	Measurement	Success rate
Kisielnicki and Misiak (2017)	1. ROI	1. Agile: 50%; Waterfall: 20%
	2. Reducing process cycle time	2. Agile: 93%; Waterfall: 20%
	3. Process improvement	3. Agile: 70%; Waterfall: 10%
	4. BI adding value	4. Agile: 53%; Waterfall: 0%
Ambler (2018)	1. Successful	1. Agile: 39%; Waterfall: 11%
	2. Challenged	2. Agile: 52%; Waterfall: 60%
	3. Failed	3. Agile: 9%; Waterfall: 29%
The Standish Group (2014)	1. Successful	1. Agile: 39%; Waterfall: 11%
	2. Challenged	2. Agile: 52%; Waterfall: 60%
	3. Failed	3. Agile: 9%; Waterfall: 29%
Serrador and Pinto (2015)	1. Project efficiency	1. A direct positive correlation between success and Agile
	2. Stakeholder satisfaction	2. A direct positive correlation between stakeholder satisfaction and Agile

their forties. Table 4 indicates the project role and a breakdown on the percentage involved in Agile or Waterfall projects. Section 2 focused on the type of software project that was the unit of analysis. Two important aspects that form part of a software project had to be covered. The first aspect speaks to the type of software project and the second to the type of method used to implement the software project. Section 3 focused on software project success itself. This section was designed around the Bannerman framework (2008, 2012). Each individual component within a specific success level was measured. A Likert scale was used where the response could vary between poor, fair, good, very good and excellent. This was applicable to the four levels of process, deliverable, business and strategic success. Actual figures were provided for the cost and duration of the project, which form part of the project management success level.

Probability sampling was used since this research focused on providing a representative view of the unit of analysis for the purpose of generalisability. Simple random sampling was se-

Table 4: Position and project type involvement

Position	Agile projects (%)	Waterfall projects (%)
Assistant PM	2	1
Project coordinator	1	1
Project manager	6	8
Senior PM	3	6
Project leader	3	2
Programme manager	3	2
Portfolio manager	0	2
Project implementation manager	1	1
IT manager	5	2
Business analyst	6	3
Project management consultant	1	1
Iteration manager	2	0
Other	8	2

lected because it not only provides results which are highly generalisable, but also adequately represents the target population. A total number of 617 valid responses were collected and used for the data analysis.

Reliability is a measure of the extent to which a research instrument consistently reflects the construct it is measuring (Field, 2018; Thomas, 2013). That is to say, if a study is conducted again at another point in time under similar circumstances, then the results of the second study should be comparable to those of the first. This study made use of scales in the assessment of project success and, as such, Cronbach's alpha was used as it measures the level of reliability (Field, 2018). The Cronbach's alpha results are presented in Table 5. A total alpha value of 0.935 resulted from the analysis and indicates that there was reliability.

Table 5: Reliability results

Items	Cronbach's alpha
Process success (4 items)	0.831
Deliverable success (7 items)	0.904
Business success (6 items)	0.792
Strategic success (5 items)	0.851
All (22 items)	0.935

Validity measures what it purports to measure (Field, 2018). If a questionnaire does not measure what it is supposed to measure, then the conclusions and statistical analysis might also be invalid. There are three major types of validity: (i) construct validity, (ii) internal validity and (iii) external validity (Balnaves & Caputi, 2001). Construct validity was used for this study.

4 RESULTS

The results are discussed as per the five levels of project success (Bannerman, 2008; Bannerman & Thorogood, 2012). The five levels of project success are discussed in details as follows: (4.1) process success, (4.2) project management success, (4.3) deliverable success, (4.4) business success and (4.5) strategic success. A final conclusion is made based on the results determining the overall success rate of a software project. The descriptive results indicate any potential differences between the success results of software projects incorporating Agile principles versus those incorporating the more traditional Waterfall approach. Analysis of variance (ANOVA) was used to determine whether there was a statistical difference between the two approaches.

4.1 Process success

Process success is the very first level of success and determines how well project teams select, integrate and implement processes. This level focuses on the various managerial and technical processes critical throughout the project lifecycle (Bannerman, 2008; Bannerman & Thorogood, 2012). Since the process success is the first level, it is important to ensure that the scope to produce the final product aligns with the project’s overall processes as it can affect the entire project life cycle and lead to project failure. The results depicted in Figure 4 indicate the extent to which processes are followed in the Waterfall methodology. Although this methodology has been used for decades, the perception is created that project teams are still not mastering it.



Figure 4: Process success: Waterfall

Teams were good (45%) or very good (32.3%) at choosing the relevant process (Waterfall) applicable for the project. The respondents were confident (77.8%) that the Waterfall methodology was aligned with the overall project management processes. The results also indicate that the Waterfall methodology was fully integrated (77.8%) and properly implemented (77.3%). These results indicate the amount of effort and time spent in choosing, aligning, integrating and implementing the relevant process for a specific project.

Project teams that opted for Agile as a methodology portrayed the same confidence as the teams that selected the Waterfall methodology. The results are depicted in Figure 5.

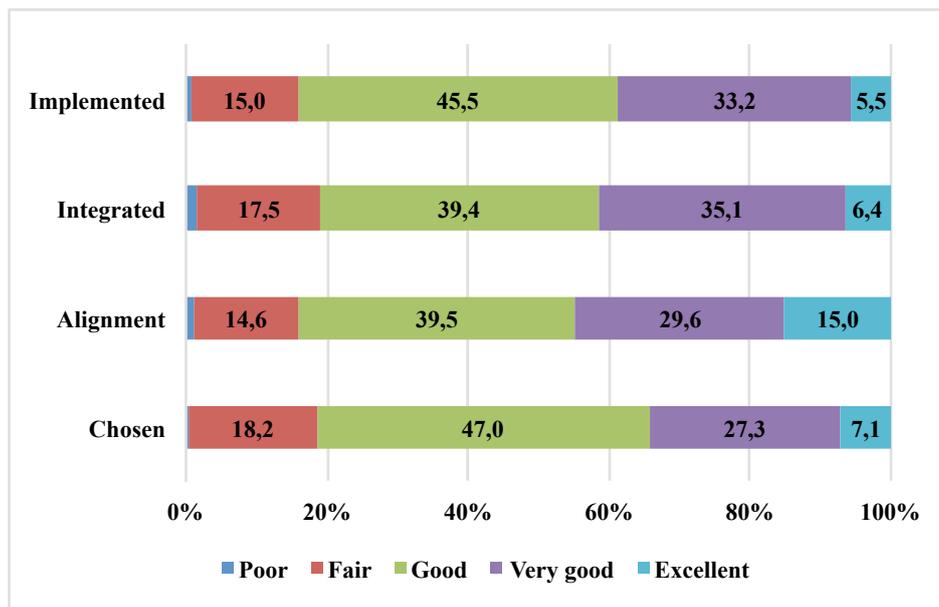


Figure 5: Process success: Agile

The results in Figure 4 and Figure 5 appear at a glance to be the same, but when these two methodologies are compared with each other, then it is clear that Agile project teams were more successful than Waterfall project teams. The results in Figure 6 show that Agile projects were more successful than Waterfall projects in the selection, integration, implementation and alignment of processes.

Agile projects average a success rate of 88.2% across all four criteria, whereas Waterfall projects are on average only 47% successful. The difference in success rates is 41.25%. The implication is that Agile as a methodology is easier to integrate into project management or that it is easier to manage Agile projects as pure development projects without the overhead of project management processes.

The results in Table 6 show that there is a statistically significant difference regarding the alignment and integration of Agile and Waterfall processes. There is a statistically significant difference between groups (alignment) as determined by one-way ANOVA ($F(1, 440) =$

5.105, $p = 0.024$). There is also a statistically significant difference between groups (integrated) as determined by one-way ANOVA ($F(1, 438) = 8.353, p = 0.004$). In all tables showing ANOVA results, “df” stands for degrees of freedom and “Sig.” is the p -value.

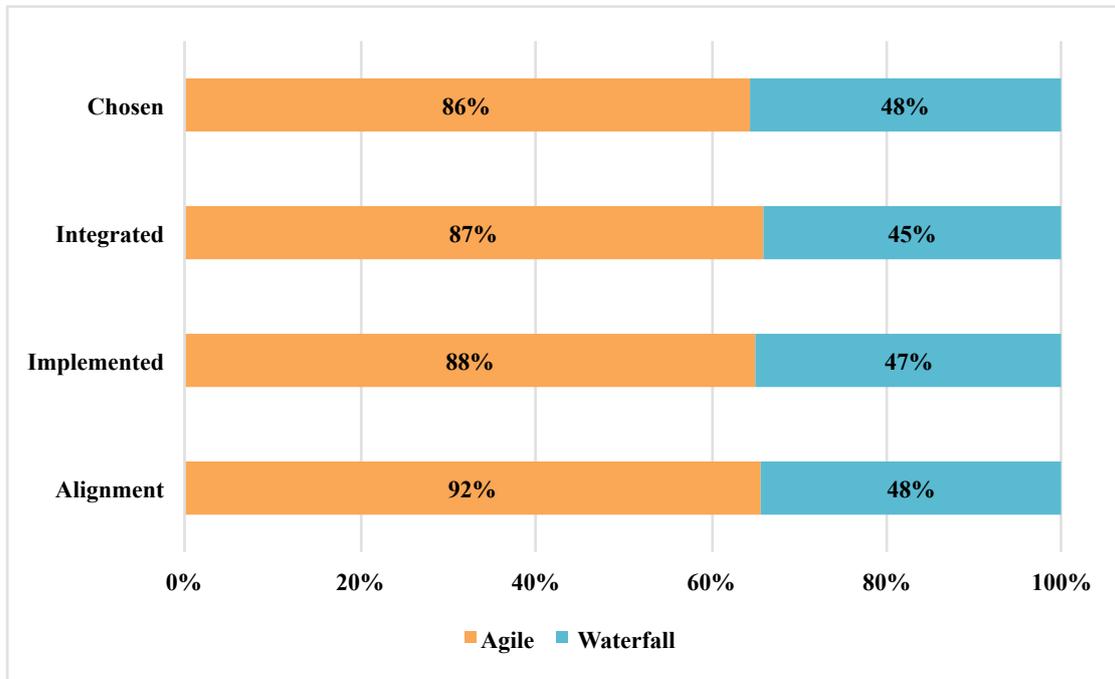


Figure 6: Process success comparison: Agile versus Waterfall

Table 6: ANOVA of process success

		Sum of squares	df	Mean square	<i>F</i>	Sig.
Chosen	between groups	0.001	1	0.001	0.001	0.978
	within groups	317.375	440	0.721		
	total	317.376	441			
Alignment	between groups	4.531	1	4.531	5.105	0.024
	within groups	390.564	440	0.888		
	total	395.095	441			
Integrated	between groups	6.167	1	6.167	8.353	0.004
	within groups	323.387	438	0.738		
	total	329.555	439			
Implemented	between groups	2.094	1	2.094	2.834	0.093
	within groups	325.056	440	0.739		
	total	327.149	441			

These two differences make logical sense as the Agile principles were developed to align and integrate with an organisation's current processes. Although the descriptive results in Figure 6 indicate that Agile methodologies are better at incorporating the processes, the ANOVA results indicate that Agile is only better in two processes.

Although the results in Figure 6 indicate a difference between the way processes are chosen and implemented, the statistical analysis indicates no significant difference. It can be deduced that irrespective of the process chosen, it is well implemented.

4.2 Project management success

Table 7 presents the comparison between Agile and Waterfall methodologies with regard to the original triple constraint. It is interesting to note that Agile software development projects are 42.62% more expensive than the original budget. Waterfall projects are, by contrast, 10.16% cheaper than originally budgeted. When Agile and Waterfall projects are compared with each other regarding time, then Waterfall projects take 13.65% longer than estimated and Agile software development projects take 22.4% longer than estimated.

Table 7: Reliability results

	Agile	Waterfall
Original budget (average)	R5 526 876.09	R39 398 302.94
Actual budget (average)	R7 882 681.12	R35 396 477.77
Original time (average)	9.60	15.22
Actual time (average)	10.91	18.63

The results presented in Table 7 create a mixed message. According to literature, Agile projects should be delivered quicker and cheaper than Waterfall projects, but this was not the case in this instance. The study was of a quantitative nature and no evidence was uncovered for these discrepancies. It might be useful to conduct interviews with the various respondents to uncover the truth behind these discrepancies.

The third constraint is the scope of the project. In the context of a project, scope is defined as the features and functions that characterise the project's deliverable (Project Management Institute, 2017). The majority of software projects delivered on the scope of the project, with 87.5% of software projects delivering between 60% and 100% of the scope. Figure 7 shows the comparison between Agile and Waterfall projects. There is no difference between these two methodologies when it comes to the delivery of the scope. Agile projects delivered between 61% and 100% of the scope in 87.5% of the cases in comparison with Waterfall that delivered between 61% and 100% of the scope in 89.5% of the cases.

When Agile and Waterfall methodologies are compared with regard to the triple constraint, then there is not much difference between the two. Project management success was therefore not dependent on the methodology that was chosen.

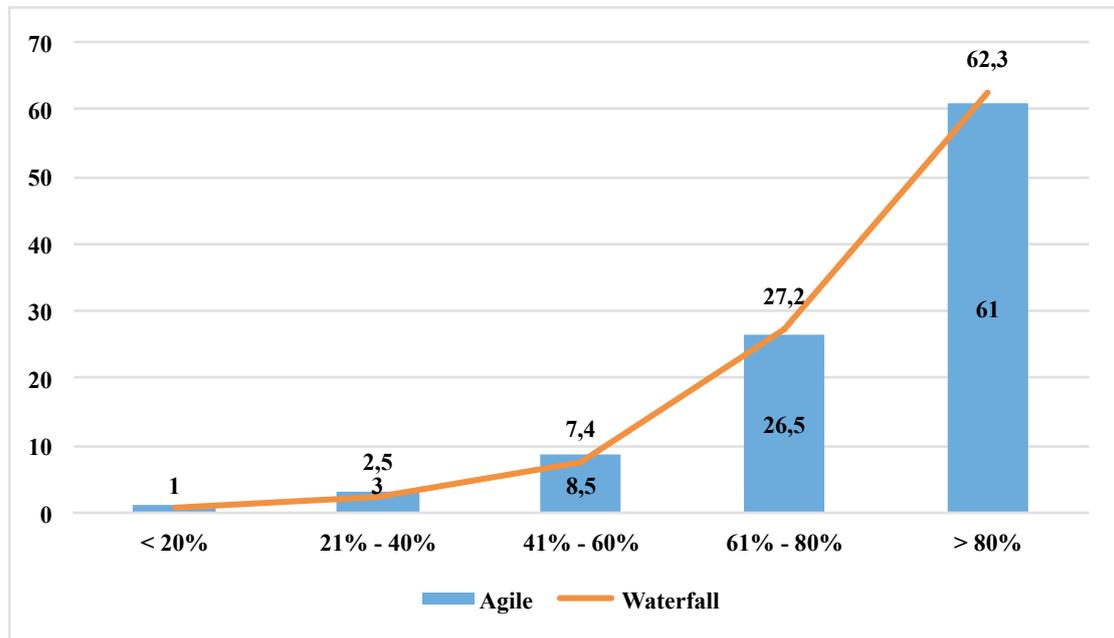


Figure 7: Scope comparison (Agile vs Waterfall)

4.3 Deliverable success

This level of project success is measured based on the final deliverable or service of the project. Seven criteria form part of this level. Figure 8 indicates the level of success for each of the criteria using the Waterfall methodology.

The criterion of meeting the specifications was perceived as successful, with 88.4% of the respondents believing that the project deliverable met the specifications. The same cannot be said for the requirements, as 22.7% believed that the requirements were either poor or fair. This confirms the literature stating that one of the drawbacks of the Waterfall methodology is that requirements cannot be determined upfront. Given the fact that not all the requirements are incorporated into the final deliverable, it comes as no surprise that the users’ expectations were not met (26.4%) and that the final product was not fully accepted (19.1%). This correlates with 18% of the users that did not necessarily use the product. This also correlates with the 17% of the users that were not satisfied with the end deliverable. This is in line with benefits realisation where 83% (45.5%—good; 25.9%—very good; 11.6%—excellent) of the respondents felt that the benefits were realised.

Figure 9 unpacks the success rates for the seven criteria using Agile as a methodology. All the respondents felt that the specifications were met in some way. This was also the case with the requirements of the deliverable. The majority of the respondents (57.3%) believed that the requirements were either very good or excellent.

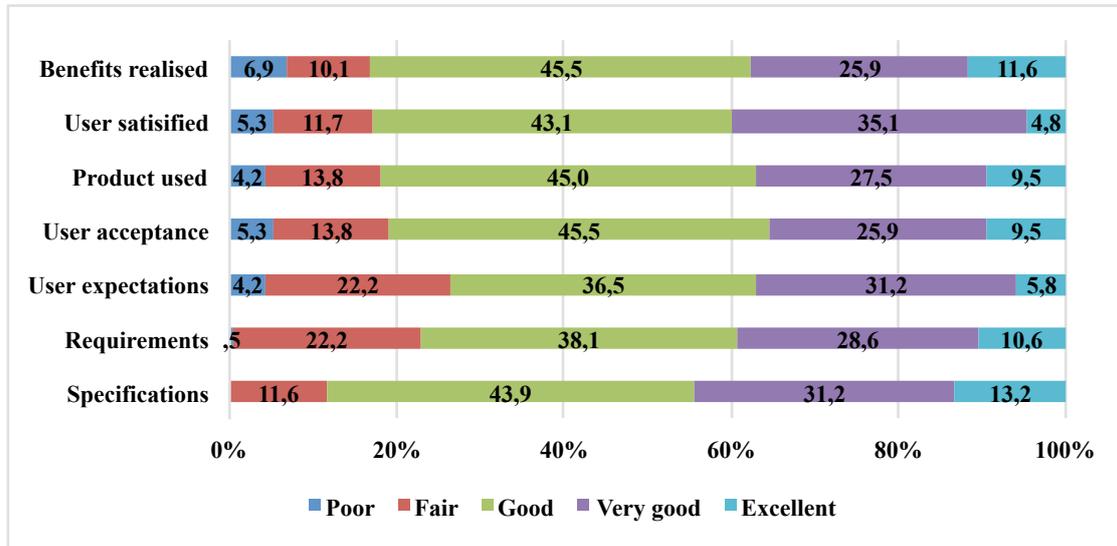


Figure 8: Deliverable success: Waterfall

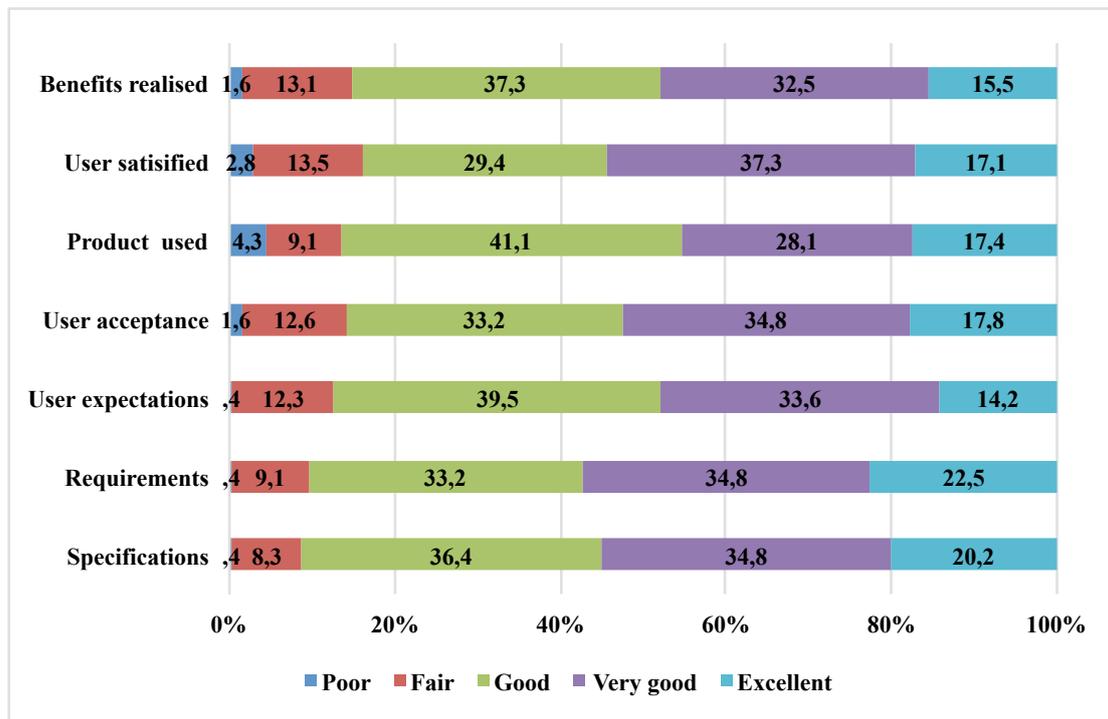


Figure 9: Deliverable success: Agile

This is in contrast to the Waterfall methodology where 39.2% of the respondents felt that the requirements were very good or excellent. Most of the respondents felt that they had met the users' expectations, with 73.1% feeling that they met these expectations either in a good or very good manner. The results also indicate that the users accepted the project deliverable. A small percentage (13.4%) of the users were not comfortable with using the deliverable. As with Waterfall projects, the respondents were confident that the deliverables realised the intended benefits.

Figure 10 illustrates the comparison between Waterfall and Agile deliverable success. The results do not reveal much apart from the fact that project deliverables produced through Agile were slightly more successful than those produced through the Waterfall methodology.

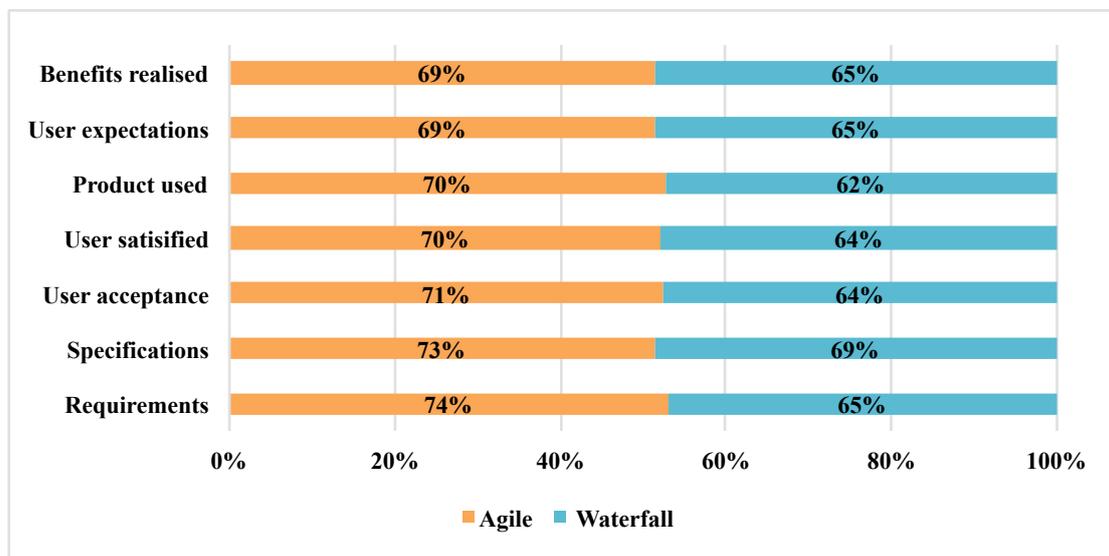


Figure 10: Deliverable success comparison: Agile versus Waterfall

The two biggest differentiators are the requirement (9%) and product used (8%) criteria. This difference can be attributed to Agile's inherent iterative nature of soliciting requirements and this spills over to the usage of the actual product or deliverable. Users will use a product if it satisfies their requirements.

The results in Table 8 clearly indicate that there is a statistically significant difference between the two approaches.

Users and customers are more involved in the design and development of the deliverable when project teams apply Agile principles. Users are involved from the beginning in determining the specifications and requirements and throughout the project, they ensure that the specifications and requirements are adhered to through the daily stand-up meetings. This implies that the users' expectations are met and they will accept the final product. The end result is that the product is used by a satisfied user, who in turns realises the benefits of the intended product or service. This is approach is different from a Waterfall approach where

Table 8: ANOVA of deliverable success

		Sum of squares	df	Mean square	<i>F</i>	Sig.
Specifications met	between groups	4.317	1	4.317	5.463	0.020
	within groups	347.719	440	0.790		
	total	352.36	441			
Requirements met	between groups	20.476	1	20.476	23.344	0.000
	within groups	385.942	440	0.877		
	total	406.419	441			
User expectations	between groups	14.685	1	14.685	17.119	0.000
	within groups	377.426	440	0.858		
	total	392.111	441			
User acceptance	between groups	12.440	1	12.440	13.043	0.000
	within groups	377.426	440	0.858		
	total	392.111	441			
Product used	between groups	12.440	1	12.440	13.043	0.000
	within groups	419.680	440	0.954		
	total	432.120	441			
User satisfied	between groups	4.645	1	4.645	4.715	0.030
	within groups	413.474	438	0.944		
	total	423.191	439			
Benefits realised	between groups	5.145	1	5.145	5.294	0.022
	within groups	426.615	439	0.972		
	total	431.760	440			

the requirements are gathered at the beginning of the project and the users are only involved again, once the product is delivered.

The fourth level of project success measures the contribution that the product or deliverable makes to business success.

4.4 Business success

The results in Figure 11 highlight that the majority of Waterfall projects were either good or very good at delivering business success. The six criteria that form part of this level indicate that the majority of the respondents were not too positive about the positive impact on business success.

The first criterion determines whether the goal of the project has been achieved. Close to 90% of the respondents believed that the goal was achieved. The same feeling was expressed regarding the success of the business plan where three quarters (74.6%) of the respondents believed that the final deliverable achieved the business plan. The third criterion focuses on governance and the extent to which governance was adhered to during the project. Three quarters of the respondents (75.2%) felt that using the Waterfall methodology contributed to the adherence to governance principles. The fourth criterion addresses the notion of benefits realisation from the perspective of the organisation. Here too, 73.4% of the respondents confirmed that benefits realisation was achieved when the Waterfall methodology was used. The

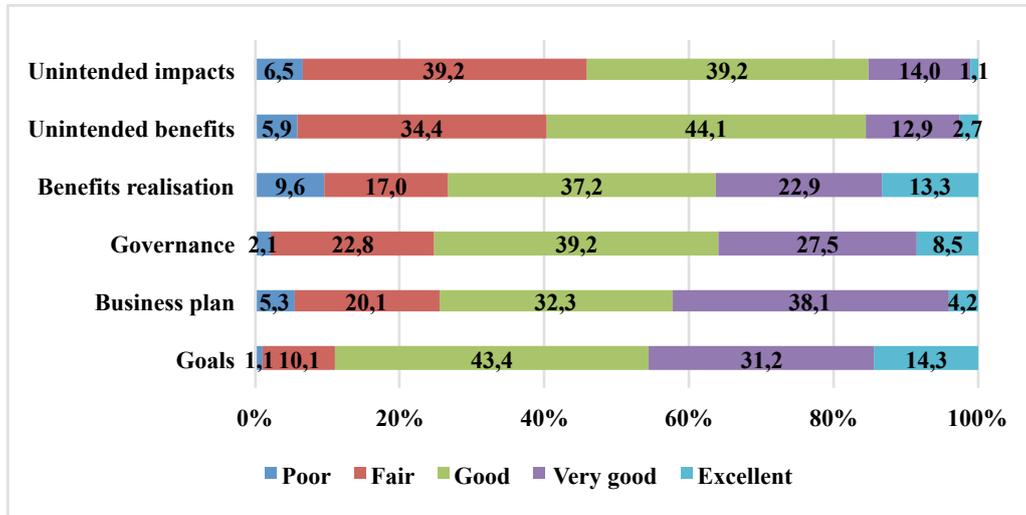


Figure 11: Business success: Waterfall

last two criteria focus on unintended benefits and impacts. Although the respondents were of the opinion that these two criteria were achieved to a certain degree, the achievement was less positive (unintended benefits 59.7% and unintended impacts 54.3%).

The results for Agile projects (Figure 12) look slightly different from those of Waterfall projects but there is no major difference between the two methodologies. The biggest difference is in the criteria of unintended benefits and impact. The results hint that a larger percentage of respondents felt that these criteria were poorly achieved.

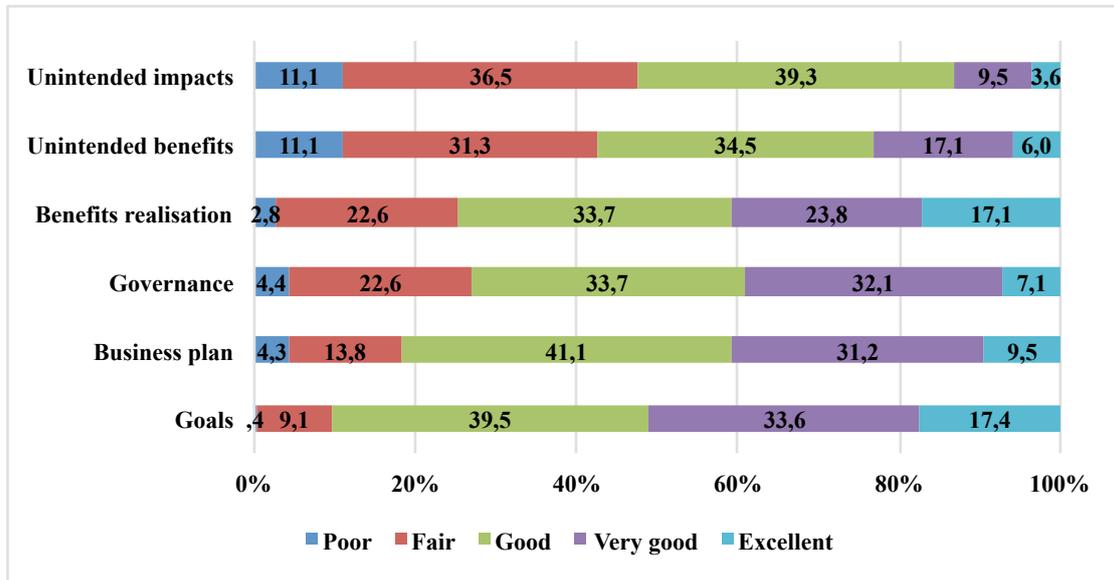


Figure 12: Business success: Agile

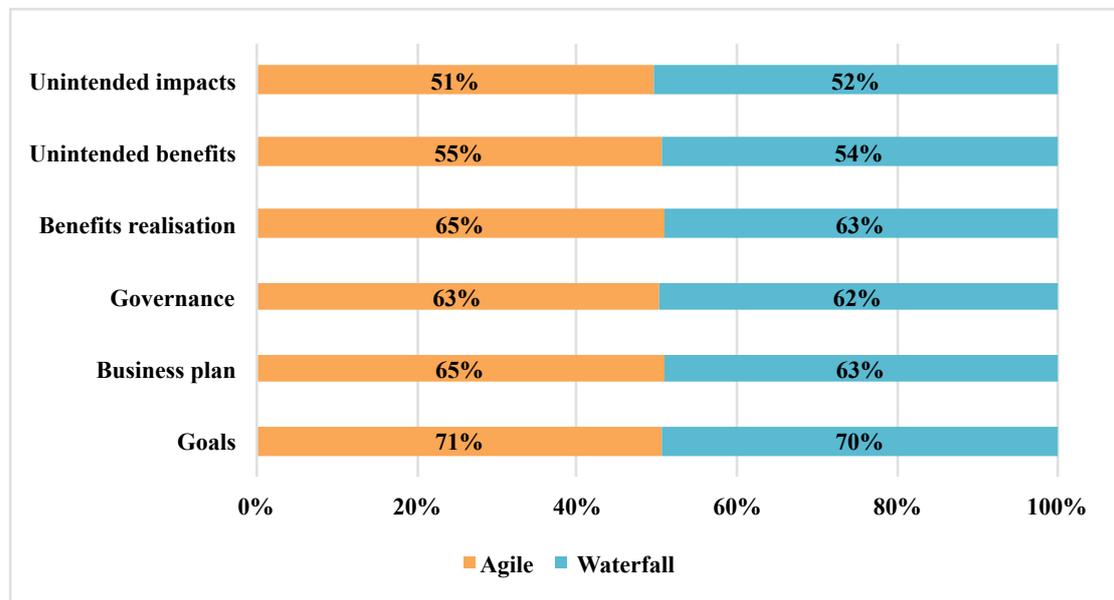


Figure 13: Business success comparison: Agile versus Waterfall

The comparison of Waterfall and Agile projects clearly reveals that there is no difference between these two methodologies when it comes to business success. The results in Figure 13 indicate that these two methodologies were equally good or bad at achieving business success.

The results in Table 9 are also interesting. There is no statistically significant difference between Agile and Waterfall projects with regard to business success. This supports the data as presented in Figure 13 that business success is achieved irrespective of the methodology used. The question is how well business success is achieved and based on the results in Figure 13, the success rates are the same between Agile (61.67%) and Waterfall (60.67%).

Table 9: ANOVA of business success

		Sum of squares	df	Mean square	F	Sig.
Goals	between groups	1.280	1	1.280	1.598	0.207
	within groups	352.566	440	0.801		
	total	353.846	441			
Business plan	between groups	1.505	1	1.505	1.608	0.205
	within groups	411.871	440	0.906		
	total	413.376	441			
Governance	between groups	0.061	1	0.061	0.064	0.800
	within groups	417.508	439	0.951		
	total	417.569	440			
Benefits realisation	between groups	2.919	1	2.919	2.375	0.124
	within groups	538.354	438	1.229		
	total	541.273	439			
Unintended benefits	between groups	0.120	1	0.120	0.126	0.723
	within groups	416.208	436	0.955		
	total	416.329	437			
Unintended impacts	between groups	0.391	1	0.391	0.486	0.486
	within groups	350.278	436	0.803		
	total	350.669	437			

The last level deals with strategic success where the focus is on the strategic advantage that the organisation gained from the investment in the software project. Six criteria address business success and the focus is the extent to which the project deliverable contributed to the impact of each criterion.

4.5 Strategic success

The results in Figure 14 highlight that Waterfall projects did not have a major impact on the criteria. Most of the criteria had a poor or fairly low impact on strategic success. Agile projects, on the other hand, had a better impact on strategic success. The results indicate that investor impact had the lowest impact and market impact the highest impact on strategic success.

The impact of these two methodologies on strategic success is illustrated in Figure 14. It is evident that Agile projects had a greater impact on strategic success. On average, Agile projects had 41% more impact on strategic success than Waterfall projects.

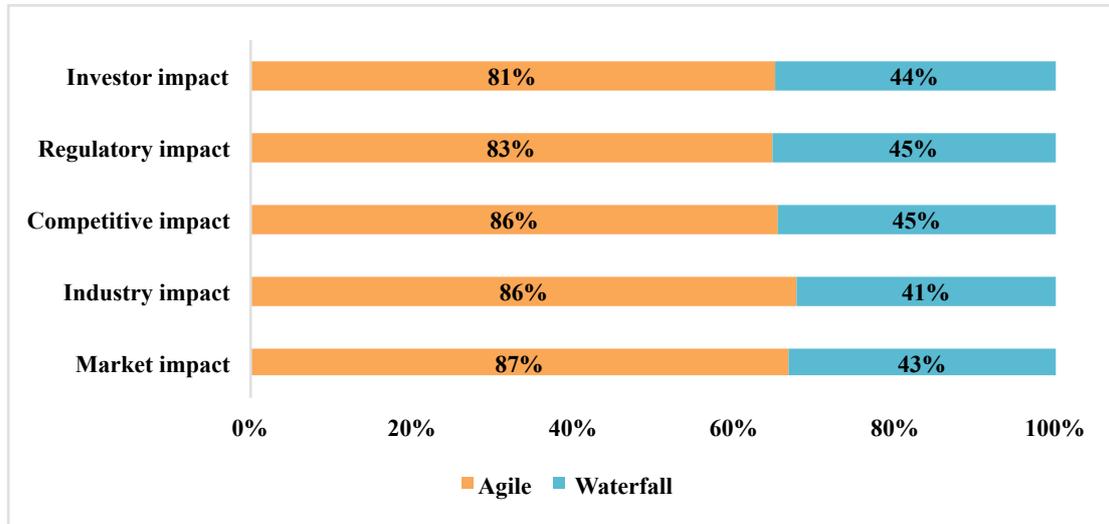


Figure 14: Strategic success comparison: Agile versus Waterfall

The results in Table 10 highlight that Agile projects are more successful with regards to market and industry impact. Market impact ($F(1, 437) = 10.594, p = 0.001$) and industry impact ($F(1, 435) = 21.821, p = 0.000$) are the only two elements indicating a statistically significant difference between Agile and Waterfall projects. This can possibly be attributed to the fact the Agile project deliverables are released quicker to market and the impacts are observed sooner than in the case of Waterfall project deliverables.

Table 10: ANOVA of strategic success

		Sum of squares	df	Mean square	F	Sig.
Market impact	between groups	9.919	1	9.919	10.594	0.001
	within groups	409.156	437	0.936		
	total	419.075	438			
Industry impact	between groups	23.204	1	23.204	21.821	0.000
	within groups	462.585	435	1.063		
	total	485.789	436			
Competitive impact	between groups	2.728	1	2.728	2.607	0.107
	within groups	457.349	437	1.047		
	total	460.077	438			
Investor impact	between groups	0.730	1	0.730	0.668	0.414
	within groups	474.682	434	1.094		
	total	475.413	435			
Regulatory impact	between groups	1.554	1	1.554	1.773	0.184
	within groups	379.581	433	0.877		
	total	381.136	434			

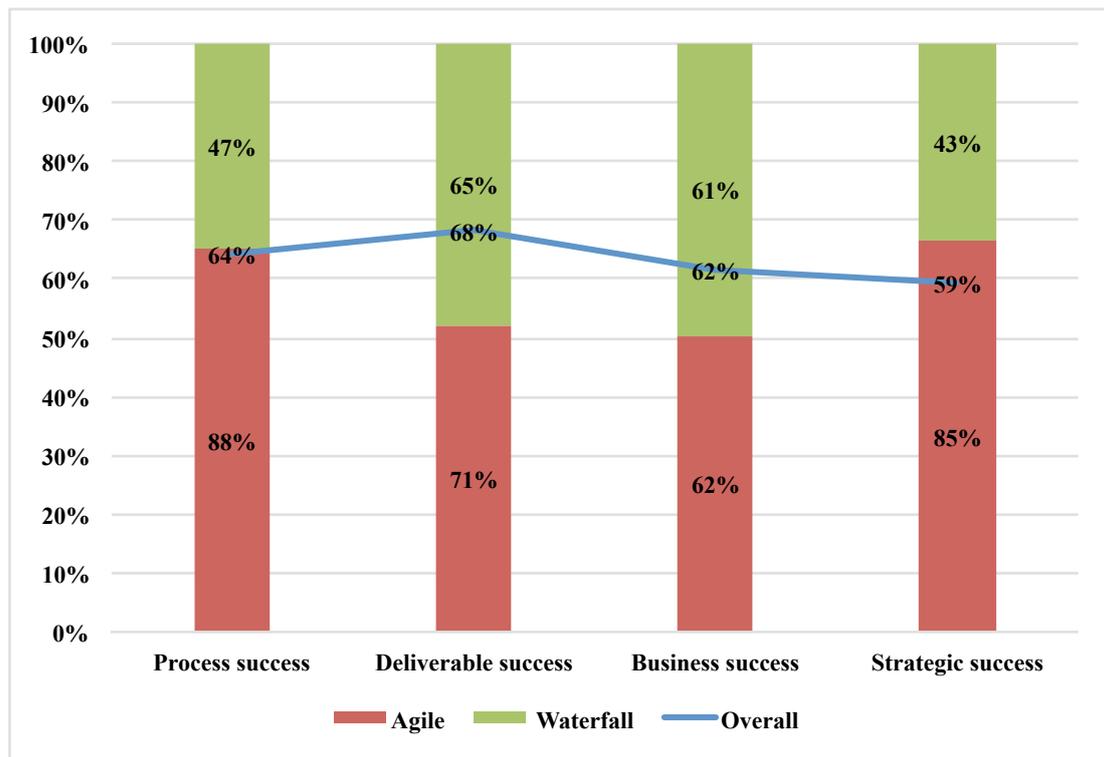


Figure 15: Overall software project success¹

There is no difference in the results given the other four factors between Agile projects and Waterfall projects. These results are in stark contrast to what is depicted in Figure 14. According to the results in Figure 14, Agile projects are, as a whole, strategically more successful than Waterfall projects.

The results paint an interesting picture. A side-by-side comparison of the two methodologies, illustrated in Figure 15, indicates that Agile projects do have an advantage over Waterfall projects. The two major distinctions between Agile and Waterfall are in the measurement of process and strategic success. Software projects using Agile principles were 88% successful when process success is measured and 85% successful when strategic success is measured. This is in stark contrast to the results of software projects that used the Waterfall methodology.

This difference can be attributed to the following factors: One of the Agile principles advocates individuals and interactions over processes and tools. It is evident from the results that project teams using Agile as a methodology adopted and adapted processes and tools as they were needed and were not prescribed as with the Waterfall methodology. Agile projects were delivered within 11 months, whereas Waterfall projects were delivered within 19 months. This implies that Agile project deliverables had an 8-month advantage over the deliverables of Waterfall projects to have a positive impact on the strategic success of the organisation.

The other two success levels are almost on par with each other when the two methodologies are compared. The overall software project success rate also improves when the five levels are used to measure project success (Table 11).

Table 11: Average software project success rates

Average success rate (%age)	
Agile	70
Waterfall	58
Overall	64

Measuring software project success (Waterfall and Agile) based on the five levels showed a significant improvement as the success rate improved from 34% in 2013 to 64%. That is a 30% increase in success. The success rate was even higher (70%) when software projects incorporated Agile principles. Table 12 compares the success rates of Agile and Waterfall projects with other similar studies.

Table 12: Average software project success rates (%age)

	This study	Ambler (2018)	The Standish Group (2014)
Agile	70	55	39
Waterfall	58	29	11

The results from this study are much more positive. This can be contributed to the fact that success is measured across five levels and not just based on the triple constraint.

¹Project success is not included in this comparison as it was not measured on a Likert scale like the other four levels.

5 CONCLUSIONS

Software project success keeps eluding organisations and forces organisational leaders to re-consider the way that success is measured. Secondly, organisational leaders must determine if there are new methodologies that can be used to address the issue of software project failures. The literature has introduced two concepts, i.e. the five levels of measuring project success and Agile methodologies as opposed to the more traditional software development methodologies.

At a glance, the impression is that projects using Agile are more successful than those using a more traditional approach such as Waterfall. A comparison of each individual success criterion provides more insight, as per Table 13. The first grouping, process success, indicates that adopting Agile methodologies contributes to two success criteria based on the ANOVA results. This correlates with the findings of Kisielnicki and Misiak (2017), who state that Agile methodologies are 60% more successful in process improvement than Waterfall as a methodology. The second grouping, deliverable success, indicates that there is a significant difference between projects adopting Agile methodologies versus traditional methodologies. There is not a significant difference between the actual success rates but the ANOVA clearly indicates that there is a significant difference.

Table 13: Detailed comparison

	Success criteria	Waterfall (%)	Agile (%)	ANOVA
Process	Implemented	48	86	No difference
	Integrated	45	87	Significant difference
	Alignment	47	88	Significant difference
	Chosen	48	92	No difference
Deliverable	Benefits realised	65	69	Significant difference
	User expectations	65	69	Significant difference
	Product used	62	70	Significant difference
	User satisfied	64	70	Significant difference
	User acceptance	64	71	Significant difference
	Specifications	69	73	Significant difference
	Requirements	65	74	Significant difference
	Business	Unintended impacts	52	51
Unintended benefits		54	55	No difference
Benefits realisation		63	65	No difference
Governance		62	63	No difference
Business plan		63	65	No difference
Goals		70	71	No difference
Strategic	Investor impact	44	81	No difference
	Regulatory impact	45	83	No difference
	Competitive impact	45	86	No difference
	Industry impact	41	86	Significant difference
	Market impact	43	87	Significant difference

There is no difference between projects adopting Agile or traditional methodologies when it comes to business success. Regarding strategic success, only two success measurements, in-

vestor impact and regulatory impact, indicate a significant difference. What is quite interesting is that although the descriptive analysis indicates a significant difference, the ANOVA does not indicate a significant difference. This needs further investigation and analysis.

The results indicate that projects adopting Waterfall are more successful when it comes to project management success. With regard to process success, Agile projects are only successful in two of the four measurements. There is a definite difference when it comes to the success of the deliverable. Projects adopting Agile methodologies are more successful in all seven measurement criteria. There is no difference in business success and Agile projects are more successful in only two measurement criteria when it comes to strategic success. The results of this study are more or less in line with other international studies. The contribution of this research is that it measured project success in more detail (22 measurement criteria) whereas other studies used a limited view on determining project success.

The results provide two insights. The first is that the success rate of software projects improves when it is measured across the five levels. This implies then that project managers should determine the success criteria based on these five levels. The second insight is that Agile projects are perceived to be more successful than Waterfall projects. This is not applicable for all the levels, only for some. There are small or no differences on the other levels.

The article contributes to the current body of knowledge at two levels. In the first instance, this is the first study of its kind that measures software project success to this extent. Software project success was measured across five levels and software projects were compared based on the methodology adopted, i.e. Agile or Waterfall. The results paint a mixed picture with regard to success rates. More in-depth analysis is required in this regard. The second contribution is that the South African results compare with those of other international studies. Although South Africa is a developing country, the results are comparable with those of developed countries. The results also provide further insight into this phenomenon and present additional information that can be used by other researchers in similar studies.

IS project success is complex and is influenced by various aspects. Future research will incorporate qualitative analysis to gain a deeper understanding of this phenomenon.

Which methodology or approach is best? The jury is still out on this.

References

- Ahimbisibwe, A., Cavana, R. Y. & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, 28(1), 7–33. <https://doi.org/10.1108/JEIM-08-2013-0060>
- Alshamrani, A. & Bahattab, A. (2015). A comparison between three SDLC models Waterfall model, Spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues*, 12(1), 106–111.
- Ambler, S. W. (2018). 2018 IT project success rates survey results. Web Page. Last accessed 10 Jul 2020. Retrieved from <http://www.ambysoft.com/surveys/success2018.html>

- Arora, R. & Arora, N. (2016). Analysis of SDLC Models. *International Journal of Current Engineering and Technology*, 6(1), 268–272.
- Association for Project Management. (2006). *APM Body of Knowledge* (5th ed.). Buckinghamshire: Association for Project Management.
- Balnaves, M. & Caputi, P. (2001). *Introduction to Quantitative Research Methods - An Investigative Approach*. London: Sage Publishers. Retrieved from <http://www.uk.sagepub.com/books/Book210544?>
- Bannerman, P. (2008). Defining project success: A multilevel framework. In *PMI Research Conference: Defining the future of project management, Warsaw, Poland, 13-16 July 2008*.
- Bannerman, P. & Thorogood, A. (2012). Celebrating IT Projects Success: A Multi-domain Analysis. In *2012 45th Hawaii International Conference on System Sciences* (pp. 4874–4883). <https://doi.org/10.1109/HICSS.2012.147>
- Bindal, N. & Mehta, A. (2015). Survey on software development processing models. *International Journal of Exploring Emerging Trends in Engineering*, 2(3), 100–109.
- Clara, V. (2013). SDLC and model selection: A case study. *International Journal of Application or Innovation in Engineering & Management*, 2(1), 273–277.
- Denscombe, M. (2007). *The Good Research Guide for Small-scale Social Research Projects* (3rd ed.). New York: Open University Press.
- Farlik, J. (2016). *Project success in agile development software projects* (Thesis).
- Field, A. (2018). *Discovering Statistics using IBM SPSS Statistics* (5th ed.). London: Sage Publications.
- Ghani, I. & Bello, M. (2015). Agile adoption in IT organizations. *KSII Transactions on Internet and Information Systems*, 9(8), 3231–3248. <https://doi.org/10.3837/tiis.2015.08.029>
- Glass, R. L. (2006). The Standish report: Does it really describe a software crisis? *Communications of the ACM*, 49(8), 15–16. <https://doi.org/10.1145/1145287.1145301>
- Heikkilä, V. T., Damian, D., Lassenius, C. & Paasivaara, M. (2015). A mapping study on requirements engineering in Agile software development. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications* (pp. 199–207). <https://doi.org/10.1109/SEAA.2015.70>
- Kisielnicki, J. & Misiak, A. M. (2017). Effectiveness of Agile compared to waterfall implementation methods in IT projects: Analysis based on business intelligence projects. *Foundations of Management*, 9(1), 273. <https://doi.org/10.1515/fman-2017-0021>
- Kumar, N., Zadgaonkar, A. S. & Shukla, A. (2013). Evolving a new software development life cycle model SDLC-2013 with client satisfaction. *International Journal of Soft Computing and Engineering*, 3(1), 216–221.
- Labuschagne, L. & Marnewick, C. (2009). *The Prosperus Report 2008: IT Project Management Maturity vs. Project Success in South Africa*. Johannesburg: Project Management South Africa.
- Mahalakshmi, M. & Sundararajan, M. (2013). Traditional SDLC vs Scrum methodology – A comparative study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192–196.

- Marnewick, C. (2012). A longitudinal analysis of ICT project success. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (pp. 326–334). <https://doi.org/10.1145/2389836.2389875>
- Marnewick, C. (2013). *Prosperus Report - The African Edition*. Johannesburg: Project Management South Africa.
- Marnewick, C., Erasmus, W. & Joseph, N. (2017). *The symbiosis between information system project complexity and information system project success*. <https://doi.org/10.4102/aosis.2017.itpsc45>
- Marnewick, C. & Langerman, J. (2018). Agile maturity: The first step to information technology project success. In G. Silvius & G. Karayaz (Eds.), *Developing Organizational Maturity for Effective Project Management* (pp. 233–252). Advances in Logistics, Operations, and Management Science (ALOMS). <https://doi.org/10.4018/978-1-5225-3197-5.ch012>
- Matharu, G. S., Mishra, A., Singh, H. & Upadhyay, P. (2015). Empirical study of Agile software development methodologies: A comparative analysis. *SIGSOFT Software Engineering Notes*, 40(1), 1–6. <https://doi.org/10.1145/2693208.2693233>
- Ohara, S. (2005). *P2M: A Guidebook of Project & Program Management for Enterprise Innovation* (3rd ed.). Project Management Association of Japan.
- Project Management Institute. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (6th ed.). Newtown Square, Pennsylvania: Project Management Institute.
- Sauer, C., Gemino, A. & Reich, B. (2007). The impact of size and volatility on IT project performance. *Communications of the ACM*, 50(11), 79–84. <https://doi.org/10.1145/1297797.1297801>
- Serrador, P. & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of Agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Silvius, G. (2017). Sustainability as a new school of thought in project management. *Journal of Cleaner Production*, 166, 1479–1493. <https://doi.org/10.1016/j.jclepro.2017.08.121>
- Sonnekus, R. & Labuschagne, L. (2003). *The Prosperus Report 2003*. Johannesburg: RAU Standard Bank Academy for Information Technology.
- The Standish Group. (2013). Chaos Manifesto 2013. Retrieved from <http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/S130301C.pdf>
- The Standish Group. (2014). Chaos Manifesto 2014. Last accessed 18 Jul 2020. Retrieved from <http://www.standishgroup.com/news/5>
- The Standish Group. (2018). The CHAOS Report: Decision latency theory. Retrieved from <https://www.standishgroup.com/store/services/10-chaos-report-decision-latency-theory-2018-package.html>
- Thomas, G. (2013). *How to Do Your Research Project - A Guide for Students in Education and Applied Social Sciences* (2nd ed.). London: Sage Publications.
- VersionOne Inc. (2015). 9th Annual State of Agile™ Survey. Retrieved from <https://explore.versionone.com/state-of-agile/9th-annual-state-of-agile-report-2>

VersionOne Inc. (2016). The 10th Annual State of Agile Report. Retrieved from <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf>

VersionOne Inc. (2017). The 11th Annual State of Agile Report. Retrieved from <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>

VersionOne Inc. (2018). The 12th Annual State of Agile Report. Retrieved from <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>

VersionOne Inc. (2019). The 13th Annual State of Agile Report. Retrieved from <https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>