

**ITRI624: INDIVIDUAL RESEARCH ASSIGNMENT -**

**SOFTWARE DEVELOPMENT METHODOLOGIES**

**19 October 2022**

**Llewellyn Anthony 32969694**

**Prof. N. Kruger**

**TABLE OF CONTENTS**

[**ABSTRACT** 1](#_Toc116820807)

[**CHAPTER 1: INTRODUCTION** 2](#_Toc116820808)

[**CHAPTER 2: RELATED WORKS** 2](#_Toc116820809)

[**CHAPTER 3: SYSTEM MODELS** 4](#_Toc116820810)

[**CHAPTER 4: PROBLEM STATEMENT** 5](#_Toc116820811)

[**CHAPTER 5: SOLUTION** 5](#_Toc116820812)

[**CHAPTER 6: FRAMEWORKS VS METHODOLOGIES** 6](#_Toc116820813)

[**CHAPTER 7: SOFTWARE MANAGEMENT METHODOLOGIES** 7](#_Toc116820814)

[**7.1.** **AGILE/SCRUM** 7](#_Toc116820815)

[**7.1.1.** **Life Cycle** 7](#_Toc116820816)

[**7.1.2.** **Functionalities** 8](#_Toc116820817)

[**7.1.3.** **Benefits** 9](#_Toc116820818)

[**7.1.4.** **Issues** 9](#_Toc116820819)

[**7.1.5.** **Companies from South Africa employing the methodology** 10](#_Toc116820820)

[**7.2.** **WATERFALL** 11](#_Toc116820821)

[**7.2.1.** **Life Cycle** 11](#_Toc116820822)

[**7.2.2.** **Functionalities** 12](#_Toc116820823)

[**7.2.3.** **Benefits** 13](#_Toc116820824)

[**7.2.4.** **Issues** 14](#_Toc116820825)

[**7.2.5.** **Companies from South Africa employing the methodology** 14](#_Toc116820826)

[**7.3.** **DEVOPS** 15](#_Toc116820827)

[**7.3.1.** **Life Cycle** 15](#_Toc116820828)

[**7.3.2.** **Functionalities** 16](#_Toc116820829)

[**7.3.3.** **Benefits** 17](#_Toc116820830)

[**7.3.4.** **Issues** 18](#_Toc116820831)

[**7.3.5.** **Companies from South Africa employing the methodology** 18](#_Toc116820832)

[**CHAPTER 8: RESULTS AND ANALYSIS** 19](#_Toc116820833)

[**CHAPTER 9: CONCLUSION** 23](#_Toc116820834)

[**BIBLIOGRAPHY** 24](#_Toc116820835)

**LIST OF FIGURES**

[**Figure 1: The System Model Proposal** 4](#_Toc116820715)

[**Figure 2: The Agile Scrum Life Cycle sprint (Despa, 2014:45)** 8](#_Toc116820716)

[**Figure 3: The Waterfall Life Cycle (Despa, 2014:41)** 12](#_Toc116820717)

[**Figure 4: The DevOps Life Cycle (Yarlagadda, 2021:115)** 16](#_Toc116820718)

**LIST OF TABLES**

[**Table 1: Scrum Benefits and corresponding codes** 9](#_Toc116820721)

[**Table 2: Survey Results from South African Companies** 19](#_Toc116820722)

# **ABSTRACT**

In most, if not all, organizations, the desire for financial gains and a competitive edge is driving technological advancements. ICT breakthroughs are becoming ever more prone to shorter life spans, rapid replacements, and obsolescence. Several Chief Executive Officers, Chief Information Officers, and Government Information Technology Officers struggle to effectively manage ICT and to establish a lasting ICT environment that supports human and social demands. This article includes a comparison between the global IT job market and South Africa's, highlighting the most popular software development methodologies utilized by both. In order to substantiate evidence as to what the most popular software development methodologies in the modern era, various small, medium, and large business case studies will be explored. An overview of at least five South African companies and the software development methodologies they have utilized will be discussed in this article. In order to illustrate the differences between these organizations in terms of the projects they completed, the software methodologies they adopted, successes and failures of the companies and implementations, and furthermore, the differences must be visualized via using tables and/or figures. A conclusion must be drawn about the most popular software methodologies utilized in the South African IT industry, based on reputable evidence provided by this article. An evaluation must occur as to why the methodologies are effective/ineffective in each case and suggestions must be offered for the future regarding software development approaches within companies, keeping in mind the effect Covid-19 had on South African industries. The essence of this article is to determine which software development methodologies are the most suitable and efficient for the international IT industry, specifically the South African technology industries.

**Keywords: Software development methodologies, Agile, SCRUM, DevOps, Framework, project management, organization, technology, methodology, Information Technology, South Africa**

# **CHAPTER 1: INTRODUCTION**

This section introduces the content about the article. Chapter 1 introduces the structure of the article. Chapter 2 discusses the related work to the article, highlighting previous research that have been conducted on the topic. Chapter 3 demonstrates the System Model, which lists all the hypotheses and assumptions of the environment, based on the research problem. A system model diagram represents the flow between arguments and sections in the article. Chapter 4 highlights the problem statement, which states the research question/problem. Chapter 5 presents the solution to the research question: A thorough justification is provided of the solution's principles, ideas, and functionalities. One should ensure that the solution does not rely on a theorem or other unproven ideas, they need to be clarified before moving on to the specific descriptions. This section's primary component is a comprehensive overview of the problem and the essence of the solution. The solution is broken down into components and described, respectively. Chapter 6 discusses the difference between a framework and a methodology, regarding the IT industry and project management. Chapter 7 lists the different software development methodologies that is utilized in the IT industry, within each subsection there is a focus on the life cycle, perks and issues regarding a certain methodology. Section 7.1 introduces the Agile SCRUM methodology, the functionalities thereof and a description of the South African companies that utilizes it. Section 7.2 introduces the Waterfall methodology, the functionalities thereof and a description of the South African companies that utilizes it. Section 7.3 introduces the DevOps methodology, the functionalities thereof and a description of the South African companies that utilizes it. Section 7.4 focuses on a hybrid Agile and Waterfall methodology, the functionalities thereof and a description of the South African companies that utilizes it. Chapter 8 visualizes the results and analysis conducted on the previous chapters and the results from a survey on software development methodologies, from five South African IT companies. Finally, a conclusion is drawn based on the research conducted and evaluation of the survey results. This article identifies the most feasible and efficient software development methodologies that happens to be a trend in the current IT industry.

# **CHAPTER 2: RELATED WORKS**

The ongoing failure of both agile and traditional software development projects has prompted discussion, debate, and study of critical success criteria, which are the elements most essential to the success of a software engineering process (Chiyangwa & Mnkandla, 2017:1). Though crucial success elements and approaches are becoming more diverse, there are only a few conceptual frameworks that can explain how they relate causally (Chiyangwa & Mnkandla, 2017:1).

Software development professionals have been sluggish to adopt agile approaches despite the potential advantages (Chiyangwa & Mnkandla, 2017:2). In a global study, Chan and Thong found that 60% of the organizations they examined were neither utilizing agile nor any traditional approaches, only 6% were adhering to a methodology by book, and 79% had no intention of adopting one (Chiyangwa & Mnkandla, 2017:2). The fact that early adopters of technology are adamantly opposed to changes to new technology is one of the factors contributing to the lack of acceptance of agile in particular (Chiyangwa & Mnkandla, 2017:2). Additionally, it is believed that agile approaches are adaptable to software development during a project and that they are generically applicable (Chiyangwa & Mnkandla, 2017:2). The adoption of agile software development approaches is still a problem that agile practitioners are focused on (Chiyangwa & Mnkandla, 2017:2).

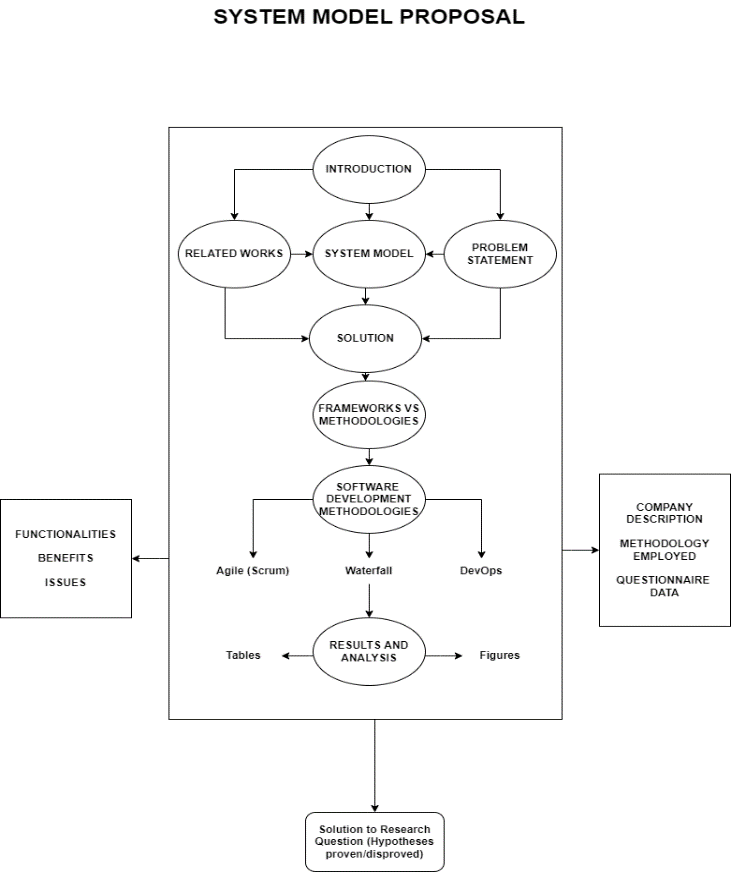
It seems that over 65% of software initiatives in South Africa were seen as failures and experiencing difficulties (Chiyangwa & Mnkandla, 2017:2). This may have occurred as a result of software development initiatives that were not finished on schedule or within the projected budget (Chiyangwa & Mnkandla, 2017:2). Thirty-four percent of the software development initiatives were thought to be successful (Chiyangwa & Mnkandla, 2017:2). In a related study, it was shown that in South Africa, 59% of the software development projects evaluated in 2011 were regarded as successful, while 41% were considered unsuccessful and suffering challenges (Chiyangwa & Mnkandla, 2017:2).

It was discovered that process factors and performance expectancy factors are causally related to success factors (Chiyangwa & Mnkandla, 2017:8). Additionally, organizational, process, people, and project aspects all directly influence how well agile software development projects are executed (Chiyangwa & Mnkandla, 2017:8). Findings from the study conducted by Chiyangwa and Mnkandla (2017:8) demonstrate that organizational variables and performance expectancy factors are the key indicators, which determines whether people become engaged in agile software development projects. The performance expectancy is the single most crucial aspect that needs to be considered while promoting software projects in South Africa, even though organizational characteristics involving agile software development projects were determined to be statistically significant (Chiyangwa & Mnkandla, 2017:8).

There happens to be insufficient proof to prove that organizations only employ one particular SDM (Systems Development Methodology) type (Moyo *et al.*, 2022). According to a study conducted by Moyo *et al.* (2022),  the results indicate that SDMs cannot be divided into the plan-driven SDM and the agile SDM in the context of systems development in South Africa. The dataset for the study revealed the utilization of hybrid SDMs (Moyo *et al.*, 2022). According to the findings, organizations implement basic SDMs and generate hybrid SDMs in response to the contextual stresses unique to each systems development project (Moyo *et al.*, 2022). The base SDM is combined with SDM components from the same class of SDMs or a different class of SDMs to create the hybrid SDMs (Moyo *et al.*, 2022).

# **CHAPTER 3: SYSTEM MODELS**

The system model is derived from hypothesis derived from the introductory sections. A hypothesis is derived from the introductory section: a system development methodology is chosen based on the type of company and projects that is being managed by that company. The system model is structured as follows: an introduction to the article, followed by related works, system model description and the problem statement. The three aforementioned sections are interrelated. The solution to the problem statement is explored afterwards. A discussion is followed on frameworks VS methodologies, which clearly indicates the differences between a framework and a methodology. Three popular software development methodologies are discussed, which consists of Agile (SCRUM), Waterfall and DevOps. Each methodology is provided a description, the life cycle of the methodologies and the functionalities thereof, as well as the known benefits and issues associated with each methodology. A description of five South African companies will be provided that utilizes one of those methodologies to manage their software projects. An analysis occurs after the discussion of the methodologies, tables and figures will be used to substantiate the evaluations. The results from surveys filled in by the five South African companies will be thoroughly analysed and visualized, as well as more generic statistics found on the software methodologies. Finally, a definite conclusion will be derived based on the analysis of the results and research conducted in this article. **Figure 1.1** demonstrates the system model:



**Figure 1: The System Model Proposal**

# **CHAPTER 4: PROBLEM STATEMENT**

The research question can be described as follows: one needs to describe the differences between popular software development methodologies, and based on industry research, determine which proves to be the most popular methodologies employed by IT companies in South Africa. Research must be conducted on at least five South African companies, examining which software development methodologies they utilize via a survey. Given the impact Covid-19 has on South African industry, an assessment of why the techniques is successful or unsuccessful in each situation and recommendations for the future in terms of software development methods within organizations are required. The primary objective of this article is to identify the software development approaches that are most effective and appropriate for the global IT sector, in particular the South African technology sectors.

# **CHAPTER 5: SOLUTION**

The solution to answer the research question is structured as follow: firstly, a discussion will occur on the differences between a framework and a methodology, to avoid confusion of using terms and clearly highlighting the definition of a framework and a methodology. A discussion follows on three popular software development methodologies utilized in the industry: Agile (SCRUM), Waterfall and DevOps. Each methodology’s discussion is structured as follows: an introduction to the methodology, the life cycle of the methodology, the methodology’s functionalities, as well as the benefits and issues associated with each methodology. After a discussion on the methodologies, a description will be provided of South African tech companies which utilizes those software development methodologies. This serves as an introduction to the companies that will be researched, and their survey results will be used to analyse and determine the most popular methodologies employed in the South African IT industry. After the discussion of the methodologies, an analysis will occur on the software development methodologies employed by the five South African companies. This will be done via analysing the survey results and visually representing it as graphs. More generic statistics on the software development methodologies will be used to substantiate the evidence from the survey, this can be done via providing tables and figures to support the results from the survey. The results received from conducting the research must attempt to solve the research question and indicate which software development methodologies are currently preferred in the industry, especially in South Africa. Finally, a conclusion must be derived as to which is the preferred methodology utilized in the industry, based on the research conducted, the methodology’s functionalities, benefits, issues and adaption to the modern era of project management and other constraints such as COVID-19.

# **CHAPTER 6: FRAMEWORKS VS METHODOLOGIES**

In general, a framework offers a basic abstraction of a solution to a variety of problems that have some similarities (Mnkandla, 2009:2). Without going into specifics about what must be done during each phase of implementation, a framework will often identify the actions or phases that must be followed to solve a problem (Mnkandla, 2009:2).

Where there is ambiguity, a framework and a methodology may be integrated into one entity (Mnkandla, 2009:2). When dealing with complicated challenges, the architectural overview of solution designs suffers in such circumstances (Mnkandla, 2009:2). The practical realities of implementing software development projects call for the adoption of approaches that can be quickly adjusted to the various and changing business environments without impacting the architecture of the system (Mnkandla, 2009:2). Therefore, if the methodology is entwined with the framework, the result is a monolithic structure rather than a more ideal modular or layered one (Mnkandla, 2009:2).

A group of approaches utilized in the creation of applications might be referred to as software engineering methodologies (Mnkandla, 2009:2). The specifics of what should be done during each stage of the software development process are provided by methodologies (Mnkandla, 2009:2). Methodologies do not always provide instructions on how something should be done (Mnkandla, 2009:2). This level of detail typically enables the business to adapt the methodology to its surroundings, for instance by creating templates, and additional documents outlining the proper way to do tasks (Mnkandla, 2009:2). A methodology is just a method of executing tasks; in this case, it refers to the culture that is adhered to when designing software systems (Mnkandla, 2009:2).

A framework should be used at a more abstract level; therefore, one requires a methodology to implement a framework (Mnkandla, 2009:3). However, sometimes people confuse the meaning of a methodology and a framework to the point where one may find that a methodology and a framework is used interchangeably (Mnkandla, 2009:3). A project management framework like PMBOK, which enables users to utilize their own methodology to implement the process in each knowledge area, comes to mind as an illustration (Mnkandla, 2009:3).

At the most basic level, a framework merely states that the task or project should be completed in stages or phases, but it typically does not mention the specific phases (Mnkandla, 2009:5). We can therefore conclude that the framework's main purpose is to provide instructions on how things should be done (Mnkandla, 2009:5).

Methodologies, on the other hand, dives into the stages and provide details of what those phases are and the related tasks in each phase (Mnkandla, 2009:5). Methodologies provide more precise instructions on what should be done (Mnkandla, 2009:5). To individuals who are interested in maximizing simplicity in systems design, implementation, and maintenance, frameworks and methodologies are not the same concept (Mnkandla, 2009:5).

# **CHAPTER 7: SOFTWARE MANAGEMENT METHODOLOGIES**

This section introduces the Agile (Scrum), Waterfall and DevOps software development methodologies. Each methodology’s life cycle will be described, as well as some of their functionalities and known benefits and issues when utilizing this methodology to manage a project. Finally, a description will be provided of a South African company known to employ the specific methodology to manage their projects.

## **AGILE/SCRUM**

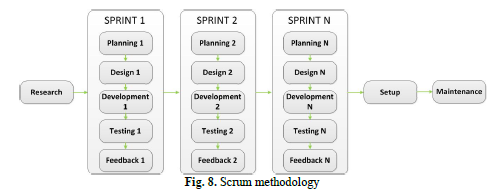
Agile is a hybrid strategy based on the renowned "Agile Manifesto" and an evolutionary methodology focused on iterative development (Jones, 2017:49). Around 17 renowned software specialists gathered in 2001 at the Snowbird resort in Utah to discuss the problems with software development and possible solutions (Jones, 2017:49). Agile is the most widely used software development methodology as of 2016 (Jones, 2017:49). The Agile Manifesto, which was released in February 2001, was the outcome of this conference (Jones, 2017:49). The following are the key tenets of the Agile Manifesto:

* People and relationships are preferable to formally established procedures and instruments (Jones, 2017:49).
* A working program is preferable to extensive documentation (Jones, 2017:49).
* Collaboration with the customer is preferable to extensive contracts (Jones, 2017:49).
* Following a strict plan is preferable to adapting to change (Jones, 2017:49).

The Agile Manifesto and the widely accepted Agile methods are designed for relatively small internal projects where the user community and the developers are housed in the same building (Jones, 2017:50).

## **Life Cycle**

Scrum is an approach for developing software progressively in complicated contexts (Despa, 2014:44). The product owner creates stories, which are a prioritized list of software requirements (Despa, 2014:45). The Product Backlog is made up of all the stories (Despa, 2014:45). The Scrum methodology implements a time-boxed strategy in which development cycles, known as sprints, are limited to 4 weeks and conclude with a functional version of the product (Despa, 2014:45). The Backlog is made up of all the stories for each sprint (Despa, 2014:45). Daily Scrum meetings, which are only allowed to last 15 minutes, are done to evaluate the progress being made (Despa, 2014:45). The project manager and other team members are not in charge of assigning tasks (Despa, 2014:45). Each team member participates in the work assignment process in self-organized Scrum development teams (Despa, 2014:45). A Scrum Master directs the team's work and keeps them on track (Despa, 2014:45). The project owner provides comments at the conclusion of each sprint (Despa, 2014:45).



**Figure 2: The Agile Scrum Life Cycle sprint (Despa, 2014:45)**

## **Functionalities**

The following is how Agile Scrum is typically implemented:

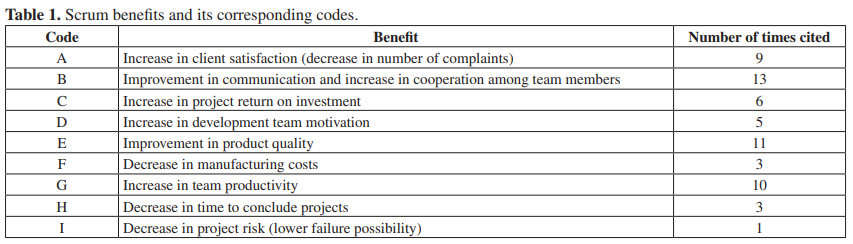
* Users who are embedded and supply requirements, are part of the development process (Jones, 2017:50)
* One makes use of user stories to study requirements (Jones, 2017:50)
* Dividing larger projects into about two-week sprints (Jones, 2017:50)
* Daily status reports in Scrum meetings (Jones, 2017:50)
* Pair programming, or two programmers going back and forth between navigating and coding (Jones, 2017:50)
* Writing test cases before writing code, often known as test-driven development (Jones, 2017:50)
* Agile-specific metrics like burn down, burn up, and velocity, etc. (Jones, 2017:50)
* Introducing agile concepts are usually done via agile “coaches”. (Jones, 2017:50)

An Agile sprint is a compact work package that can be completed in a few weeks and once finished, offers users the features they have requested (Jones, 2017:50). Over a string of numerous sprints, the entire project will be developed (Jones, 2017:50). A daily status meeting presided over by a Scrum master is referred to as a Scrum meeting. Progress, issues, and fascinating technical subjects will be discussed as necessary (Jones, 2017:50).

Agile requirements are frequently obtained as user stories or scenarios that describe how particular features will function when the project is in use (Jones, 2017:50). An epic is a collection of user stories (Jones, 2017:50). Embedded users are one or more application users that collaborates on a daily basis with the development team to help define the requirements for each sprint rather than being kept apart from them (Jones, 2017:50). Typically, firms hire agile coaches when they first decide to embrace the methodology. They often take part in the first sprints and planning meetings while also educating managers and team members about the fundamentals of Agile (Jones, 2017:50).

## **Benefits**

**Table 1: Scrum Benefits and corresponding codes**



**Source: De Carvalho and Mello (2011:44)**

## **Issues**

The Agile methodology loses effectiveness for larger projects with geographically dispersed teams (particularly across numerous time zones) (Jones, 2017:51). Agile can also be challenging when working on projects with numerous subcontractors (Jones, 2017:51). Agile is not ideal for distributed development, particularly if foreign development departments are involved, where there may be numerous time zone inconsistencies that make even Skype or web-based Scrum meetings challenging (Jones, 2017:51). Agile is not the best approach for improving legacy programs or maintaining and fixing bugs because it is largely focused on newer projects (Jones, 2017:52). It is also clear that Agile is a poor fit for projects requiring official government certifications, such as avionics kits and medical devices (Jones, 2017:52). The informal Agile approach to documentation is inappropriate for the legally required paperwork that government regulations impose on projects (Jones, 2017:52). Planning and estimating for Agile projects have proven to be a little more challenging than for other methodologies due to the numerous sprints and Agile's ability to accommodate frequent requirements changes (Jones, 2017:52). Agile also struggles to measure outcomes, which provides little information on quality and productivity (Jones, 2017:52). Comparisons of the productivity of Agile projects to those of other methodologies are challenging due to the peculiar structure of several sprints (Jones, 2017:52).

## **Companies from South Africa employing the methodology**

**RETRO RABBIT**

Retro Rabbit is a UI/UX company that creates outstanding software. They provide specialized solutions that foster brilliance and creativity. They produce innovative products and provide clients with the most comprehensive information technology design solutions by fusing data science, human-centred design, and the most recent software development methodologies.

They are the epitome of agile development, collaborating directly with their clients to build a solution that will meet the needs of the users and backed by their team of skilled individuals. Their specialties include game development, experience design, cloud computing, artificial intelligence, and mobile/web development. From engineering to big data to financial services, they have experience in every industry. Standard Bank is one of their most notable clients. The company’s headquarters is based in Wapadrand, Pretoria.

**ENTELECT**

Entelect is an innovative software engineering firm that combines technology and best practices to provide innovative solutions to a variety of industries. Entelect has an unrivalled level of individuals and depth of knowledge dedicated to answering customers' needs on their terms. Entelect was founded on the uncompromising philosophy of highly educated, best-in-class professionals.

Leading publicly traded firms as well as specialized private businesses in need of specialized software solutions make up the majority of their clients. The best software developers in South Africa work for them, and they have successfully designed and put hundreds of unique systems into operation both in South Africa and abroad.

With numerous internal training programs, from leadership development to specialized technology courses, Entelect provides its employees with unequalled professional growth options. Their 1up rewards program provides a range of exciting advantages for high performers, whether they help out co-workers or the foundation's youth, work together on initiatives, go to social events, or simply do a wonderful job. Their headquarters is based in Melrose, Johannesburg.

**AGILE BRIDGE**

The software consulting firm Agile Bridge is situated in Hazelwood, Pretoria. The Agile Bridge team is still dedicated to studying in order to stay on the forefront of programming and software development techniques. They take pleasure in their capacity to challenge one another to improve, and they are constantly learning from one another, their clients, and the top professionals in their industry, both locally and globally.

The development of integrated and enabling software solutions for medium and big corporate clients in South Africa is driven by their special combination of expertise, certification, and in-depth business understanding.

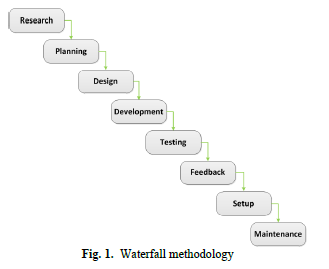
The term Agile Bridge contains the ideology they uphold. The word Agile has two different meanings. On the one hand, it shows how adaptable the business is when trying to meet the needs of client. The meeting location where they co-create business software solutions with clients is indicated by the moniker Bridge. The bridge is another way of saying the software bridge pattern language. The concept of a bridge is extremely common in the language of software and systems. A bridge is employed at an elevated level to connect various domains. A bridge is used to separate abstraction from implementation at a lower level.

## **WATERFALL**

After cowboy, this methodology is the second-oldest approach (Jones, 2017:523). It first appeared in the late 1960s, when software projects expanded to the point that they needed teams and were therefore expensive (Jones, 2017:523). A previous approach with a similar purpose was established in 1956 for the Semi-Automatic Ground Environment (SAGE) defence system (Jones, 2017:523). Between roughly 1970 and 2014, waterfall development was utilized on more than 2,000,000 projects, including some significant applications that are still in use today (Jones, 2017:523). Although several alternative methodologies, like Agile, team software process (TSP), and Rational Unified Process (RUP), have replaced waterfall, it is still frequently used (Jones, 2017:523). The waterfall method's quality and productivity results fall short of those of its superior replacement approaches (Jones, 2017:523). Due to the fact that this technique's visual depiction mimics a stream flowing over a string of waterfalls, it is known as waterfall development (Jones, 2017:523). Despite the name, the waterfall stages don't truly end before the next one starts in reality (Jones, 2017:523). As an illustration, requirements are typically only around 50% complete when design, 60% complete when coding, and only about 35% complete when testing begins (Jones, 2017:523). Due to the phases' overlap, it is difficult to plan and estimate projects because the overall number of phases does not add up to the whole number of project schedule hours (Jones, 2017:523). Despite being used frequently in many countries today, waterfall is seen as something of an antique that is steadily losing favour as of 2016 (Jones, 2017:525).

## **Life Cycle**

Waterfall places a focus on rigorous preparation and produces thorough documentation (Despa, 2014:41). The Waterfall methodology is a sequential, linear process in which each step begins only after the one before it is complete (Despa, 2014:41). Deliverables are specific to each level (Despa, 2014:41). The Waterfall methodology emphasizes meticulous software planning and architecture and is predictable (Despa, 2014:41). After the software application has been fully developed and tested, the project owner is consulted (Despa, 2014:41). The Waterfall methodology is appropriate for small-scale software development projects where requirements are clear and thorough project planning is simple to create (Despa, 2014:41).



**Figure 3: The Waterfall Life Cycle (Despa, 2014:41)**

## **Functionalities**

**Analysis Phase - software requirements specification (SRS):**  thorough and detailed explanation of how the software's behaviour will be described. System and business analysts are involved in defining both needs for both functional and non-functional requirements (Bassil, 2012). Usually, use cases are used to specify functional requirements (Bassil, 2012). This describes how users interact with the program (Bassil, 2012). They include specifications like purpose, scope, and perspective, capabilities, software characteristics, and user features, functions, requirements, and any other requirements (Bassil, 2012). Non-functional requirements are any criteria, constraints, limitations, or requirements placed on the software's operation and design as opposed to specific actions (Bassil, 2012). Reliability, scalability, testability, availability, maintainability, performance, and quality standards are some of its characteristics (Bassil, 2012).

The planning and problem-solving process for a software solution is called the **design phase** (Bassil, 2012)**.** To describe the strategy for a solution, which comprises algorithm design, software architecture design, database conceptual schema and logical diagram design, concept design, graphical user interface design, and data structure definition, it involves software developers and designers (Bassil, 2012).

**Phase of implementation**: Through programming and deployment, a concrete executable program, database, website, or software component is created from the realisation of business needs and design standards (Bassil, 2012). The database and text files are developed during this stage, and the actual code is written and assembled into a usable program (Bassil, 2012). In other words, it is the process of transferring all of the specifications and plans into a real-world setting (Bassil, 2012).

**The testing phase**, sometimes referred to as verification and validation, is the procedure used to ensure that a software solution complies with the original requirements and specifications and serves the intended purpose (Bassil, 2012). In reality, validation is the process of evaluating software during or after the development process to determine whether it satisfies specific requirements, whereas verification is the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (Bassil, 2012). Additionally, the testing stage is where debugging is carried out, in which problems and system malfunctions are identified, fixed, and improved as necessary (Bassil, 2012).

**The maintenance phase** involves making changes to a software solution to improve performance and quality, refine output, and fix problems after it has been delivered and deployed (Bassil, 2012). This stage also allows for the performance of additional maintenance tasks such software environment adaptation, user requirement accommodation, and software reliability improvement (Bassil, 2012).

Companies and projects that still utilize waterfall can supplement it with more contemporary techniques like automated testing and static analysis (Jones, 2017:525). In a waterfall setting, formal inspections of important deliverable subjects including requirements, design, and code inspections are particularly helpful (Jones, 2017:525).

## **Benefits**

Waterfall is supported by all parametric estimating tools and has more valid benchmark data than any other methodology (Jones, 2017:525). This means that estimating and measuring waterfall projects is fairly easy to do (Jones, 2017:525). The benefits of using the Waterfall Method are undeniable because it delivered the discipline and organization that were most required at the time the method was developed (Simão, 2011:43). It introduced organization, preparation, and discipline to a world where coding and fixing was the primary tactic (Simão, 2011:43). The cost of a project increases significantly when unanticipated requirements are found (Simão, 2011:43).

## **Issues**

The primary flaw of waterfall, which appears to have some merit, is that it strives to do too much too soon, such as developing complete requirements before beginning design (Jones, 2017:525). When design work begins, requirements are rarely more than 50% firm (Jones, 2017:525). The fundamental tenet of the waterfall is "measure twice, cut once," which is both its most cherished and despised concept (Simão, 2011:43). The Waterfall approach is referred to as Big Design Up Front (BDUF), where first it is carefully determined what/how and whether it is worthwhile to be done (measure"), and then start coding the product ("cut") (Simão, 2011:43). An ideal model that can only be roughly approximated is the Waterfall approach (Simão, 2011:44). The methodology's worst naysayer is the discipline and perfectionism it promotes (Simão, 2011:44). A project shouldn't waste a lot of time in the pre-coding phases, as most clients and users will not become aware of certain needs until later in the project (Simão, 2011:44). On the other hand, moving up the Waterfall processes is far more expensive than moving through the phases (Simão, 2011:44). A feedback loop between phases is also recommended by some authors in order for the Waterfall technique to be effective (Simão, 2011:44). This will prevent errors made early on from affecting later phases, and the linearity of the process is the key to success when employing the model (Simão, 2011:44).

Common criticisms of the model include the idea that it is difficult to strike a balance between resources and requirements and that documents or specifications should not be changeable because software development uses a static document that can be changed without having to return to a previous phase (Simão, 2011:44). Additionally, the needs for real-world projects change more frequently than expected, and the method's monolithic architecture makes it difficult for it to adapt to these changes (Simão, 2011:44). Despite the fact that it has drawbacks, Waterfall is still regarded as one of the most significant techniques that has significantly impacted the software industry (Simão, 2011:44).

## **Companies from South Africa employing the methodology**

**DIMENSION DATA**

With a portfolio of services that includes systems integration, managed services infrastructure, cloud solutions, business applications, customer experience, and intelligent security solutions, Dimension Data is a top African-born technology firm with operations in the Middle East and Africa. Dimension Data offers innovative solutions that maximize the rapidly changing technological landscapes of today and empower customers to utilize data in the digital age.

Dimension Data, a proud member of the NTT Group, one of the top information and communication technology (ICT) firms in the world, was founded in 1983 and is headquartered in Johannesburg. The NTT Group is made up of a number of different international technology enterprises.

In 2020, Dimension Data finished integrating all of its companies under a single Dimension Data brand, employing approximately 10,000 people in fifteen different countries. With the best technology from across the world, Dimension Data continues to aggressively invest in innovation, offering everything from consultancy, technical, and support services to a completely managed solution.

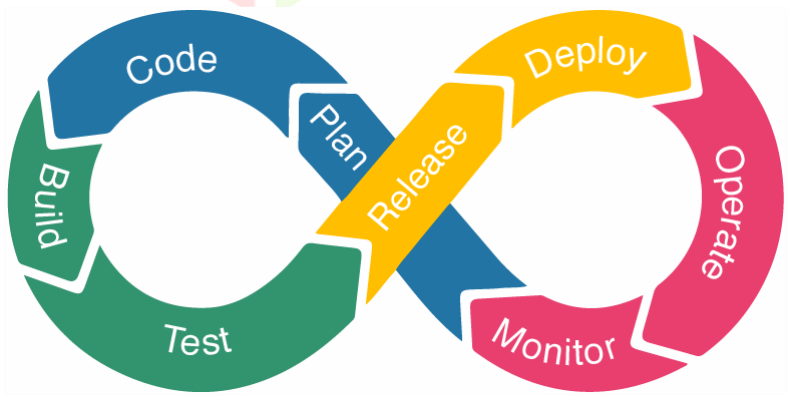
After successfully completing a big BBBEE transaction in 2019, Dimension Data is now a level 2 BBBEE contributor. Dimension Data will keep putting initiatives into action to make sure it makes a meaningful contribution to the shift toward an inclusive society.

## **DEVOPS**

The words "development" and "operations" are combined to form the term "DevOps" (Jones, 2017:149). One of the more recent approaches is called DevOps, and it is built on cooperation between the development team and the operations and maintenance teams (Jones, 2017:149). DevOps is a rather modern approach, which started in Belgium in 2009 (Jones, 2017:149). Modern techniques like continuous integration and continuous delivery are used in DevOps (Jones, 2017:149). DevOps aims to produce operations that are more stable and reliable while also requiring minimal maintenance (Jones, 2017:149). This is in addition to faster development and higher quality deliverables than Waterfall (Jones, 2017:149). DevOps is obviously targeted for software projects where operations people are located in the same building as development personnel (Jones, 2017:149). DevOps is not intended to be applied to commercial software that will be installed at countless remote sites in hundreds or possibly thousands of businesses (Jones, 2017:149). Many of the largest software developers in the world, including IBM, Computer Associates, Netflix, Facebook, Amazon, Google, and Twitter, as well as thousands of medium-sized and small businesses, have quickly embraced DevOps concepts (Jones, 2017:149). Two of the modern software development approaches that are expanding the quickest are DevOps and container development (Jones, 2017:150). They are not appropriate for all sizes and types of software applications because both are slightly specialized (Jones, 2017:150).

## **Life Cycle**

The DevOps cycles provide cyclic, natural cycles that the practitioners use to mark their lifecycles (Yarlagadda, 2021:115). The loop emphasizes the need for regular communication and repetition for phase improvement despite the DevOps life cycle's sequential flow (Yarlagadda, 2021:115). A DevOps lifecycle consists of six iterative stages that represent the skills, procedures, and methods essential for development (Yarlagadda, 2021:115). The left and right sides of the loop represent various operating talents, procedures, and tools. To ensure that the process's speed, alignment, and quality are maintained, each facet faces independent but ongoing team collaboration and communication (Yarlagadda, 2021:115). Planning, construction, integration and deployment, monitoring, operations, and responding to feedback are part of the life cycle (Yarlagadda, 2021:115). Each team should have access to open tools and guidelines at every step of development before beginning any work (Yarlagadda, 2021:115). The devices can be modified to meet the demands and objectives of the developer (Yarlagadda, 2021:115). The technique leads to the rapid development of trustworthy, high-quality software (Yarlagadda, 2021:115). Because the stages are iterative, identifying bugs becomes simpler and faster (Yarlagadda, 2021:115).



**Figure 4: The DevOps Life Cycle (Yarlagadda, 2021:115)**

## **Functionalities**

DevOps has the sense of a start-up business where developers go beyond just developing and act as quality assurance (QA) specialists, testers, and even have job supporting operations (Jones, 2017:149). These numerous roles are now known collectively under the new title full stack developer (Jones, 2017:149). DevOps also has a strong Agile character and frequently applies several Agile principles, such as test-driven development, embedded users for requirements, and occasionally Scrum sessions (Jones, 2017:149). Some of the functionalities of DevOps include:

* + - * Regular communication between development and operations staff (Jones, 2017:150)
      * Ongoing communication between developers and stakeholders (Jones, 2017:150)
      * Constant improvement of deliverables (Jones, 2017:150)
      * Every day, continuous deliveries (Jones, 2017:150)
      * Initial QA (Quality Assurance) (Jones, 2017:150)
      * A bigger focus on developers' quality roles (Jones, 2017:150)
      * Early development testing (Jones, 2017:150)
      * The absence of commonly used productivity measures like function points (Jones, 2017:150)
      * The absence of uniform quality metrics (Jones, 2017:150)
      * Limited information on defect elimination effectiveness and defect removal efficiencies (DRE) (Jones, 2017:150)

## **Benefits**

DevOps has demonstrated decent performance through 2016 in its primary niche of IT applications with high transaction rates and frequent or continuous runs (Jones, 2017:150). DevOps is becoming increasingly popular, which has sparked the development of dozens of specialized new tools and even specialized new businesses that provide training and support for emerging DevOps organizations (Jones, 2017:150). Agile experienced slower growth in 2014 than DevOps (Jones, 2017:150). Business software applications with large transaction rates and frequent runs, possibly continuous 24/7 runs, are the major focus of DevOps (Jones, 2017:150). Massive solutions have been introduced to the industrial platform via DevOps (Yarlagadda, 2021:116). Through the rapid adoption of methodology and technologies in their development processes, the majority of industries and organizations, particularly those in the United States, have reaped the benefits of DevOps (Yarlagadda, 2021:116). For instance, the American company Enhanced Automation has eliminated the usual aches associated with the cumbersome IT processes (Yarlagadda, 2021:116). Through the advantages of agile, the business has placed its full attention on automation, cooperation, and adaptability (Yarlagadda, 2021:116). The following advantages have been garnered by these organizations: firstly, they spend less time marketing their goods or services since they are current and address current problems (Yarlagadda, 2021:116). As a result, they put their deliverables much quicker into production (Yarlagadda, 2021:116). Secondly, throughout time, the quality of return on investment has greatly improved (Yarlagadda, 2021:117). Thirdly, real-time purchases of services and goods with quicker delivery have resulted in great client satisfaction (Yarlagadda, 2021:117). Fourthly, there has been excellent operational efficiency with less cost and effort due to shorter operational times (Yarlagadda, 2021:117). Automation is used to make that happen. Fifth, the increased collaboration has given developers and IT managers a platform to use straightforward codes to fix current problems (Yarlagadda, 2021:117). Finally, these codes aid in the quick identification and correction of faults and problems (Yarlagadda, 2021:117). The methods and culture of these development teams limit the risks of misunderstandings and process misalignments, enabling them to anticipate changes with ease (Yarlagadda, 2021:117). Efficiency and product quality both improve with clear and regular communication. Regular testing, deployment, and integration also encourage quick process development and error discoveries (Yarlagadda, 2021:117).

## **Issues**

Similar to Agile, DevOps is plagued by subpar measurement techniques and a lack of trustworthy quantitative data on output, timelines, quality, and customer satisfaction (Jones, 2017:150). The DevOps methodology is controversial, and some claim that it lessens the impact of competent programmers by putting them in test and QA jobs for which they may be unprepared, unqualified, or both (Jones, 2017:150). Since DevOps concepts are still evolving and maturing, the technique is still considered to be "new" and not "stable" (Jones, 2017:150). DevOps, as the name suggests, necessitates access to the software's operations team and data centres (Jones, 2017:150). This indicates that DevOps targets internal IT or online initiatives specifically (Jones, 2017:150). It is obviously inappropriate for embedded applications and systems that run unattended after deployment (Jones, 2017:150). DevOps is also not suited for modern computer-controlled medical equipment like pacemakers, cochlear implants, MRI machines, computed tomography scanners, and other types of implants (Jones, 2017:150). The same is true for many types of military software utilized in combat situations as well as weapon systems (Jones, 2017:150). It is advisable for anyone interested in DevOps concepts to perform frequent Google searches to get the most recent information because DevOps concepts are in flux, which seems to fluctuate on a weekly basis (Jones, 2017:150).

## **Companies from South Africa employing the methodology**

**BBD**

With 38 years of technical and developer experience, BBD is a provider of software development and application design solutions for the education, financial services, insurance, gaming, public, and telecommunications industries. BBD is the industry leader in independent software development, employing over one thousand highly qualified, driven, and experienced IT experts.

With operations in Amsterdam, Burgas, Cape Town, Johannesburg, London, Porto, Pretoria, and Pune, they are able to address near- and co-shoring requirements by bringing in the top analysts and software engineers from across the world. Because they have access to in-demand skills, they ensure that they deliver on their commitment to each customer. Their headquarters is based in Amsterdam, the Netherlands.

# **CHAPTER 8: RESULTS AND ANALYSIS**

**Table 2: Survey Results from South African Companies**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Please enter the company name.** | Retro rabbit | BBD | Entelect | Dimension Data | Agile Bridge |
| **Choose the most appropriate Company size.** | Medium (101-500 Employees) | Large (501-1000 Employees) | Enterprise (More than 1000 Employees) | Enterprise  (More than 1000 Employees) | Small (1-100 Employees) |
| **Roles of interviewee.** | Junior Software Developer | Development lead | Software Engineer |  | Software Developer; |
| **What Predominant SDM is your company using?** | Agile (Scrum); | Agile (Scrum)  Waterfall  Rapid (RAD)  Prototyping  Dev Ops | Agile (Scrum); | Waterfall  Agile (Scrum)  DevOps; | Agile (Scrum); |
| **Is there a separation of function and duty between IT PM and the SDM?** | No; | No; | Yes; | Yes; | Yes; |
| **What Project Management Methodology, Body of knowledge or framework, is your company using?** | Agile; | Agile  Waterfall | Agile; | Waterfall  Agile  SAFE Agile; | Agile; |
| **Before COVID-19, most work was done on the premise.** | Strongly agree | Strongly agree | Strongly agree | Strongly agree | Agree |
| **During COVID-19, most work was done on the premise.** | Strongly Disagree | Strongly Disagree | Strongly Disagree | Strongly Disagree | Strongly Disagree |
| **Post-COVID-19, most work was done on the premise.** | Strongly Disagree | Agree | Disagree | Disagree | Strongly Disagree |
| **COVID-19 had an impact on my organisation.** | Agree | Strongly agree | Strongly agree | Strongly agree | Agree |
| **COVID-19 had an impact on the way we conducted Project Management.** | Agree | Strongly agree | Strongly agree | Disagree | Agree |
| **COVID-19 had a positive impact on the way we conducted Software Development.** | Agree | Disagree | Strongly agree | Agree | Agree |
| **COVID-19 has had a positive impact on Organisational Productivity.** | Agree | Agree | Disagree | Agree | Agree |
| **COVID-19 has had a positive impact on Project Management Productivity.** | Disagree | Agree | Disagree | Agree | Agree |
| **COVID-19 has had a positive impact on Software Development Success.** | Agree | Agree | Strongly agree | Agree | Agree |
| **COVID-19 has had a positive impact on my Personal Productivity.** | Strongly Disagree | Strongly agree | Disagree | Agree | Agree |
| **Working in an online environment has been a positive experience.** | Agree | Strongly agree | Agree | Strongly agree | Agree |
| **COVID-19 had the most significant impact on the people within software development and project management.** | Agree | Agree | Agree | Strongly agree | Agree |
| **COVID-19 had the most significant impact on the Processes we follow within the project development and management team.** | Agree | Strongly agree | Strongly agree | Disagree | Disagree |
| **COVID-19 had the most significant impact on the Technology we use within the project development and management team.** | Strongly agree | Agree | Agree | Agree | Agree |
| **Covid-19 had an Impact on Governance and Strategy within the organisation.** | Strongly agree | Agree | Agree | Agree | Disagree |
| **Covid-19 had a Positive Impact on Culture and Maturity within the organisation.** | Agree | Agree | Strongly disagree | Agree | Disagree |
| **Covid-19 had a Positive Impact on Operations and Services within the organisation** | Disagree | Agree | Agree | Agree | Agree |
| **Covid-19 had an Impact on Technology Infrastructure within the organisation** | Agree | Agree | Strongly agree | Agree | Agree |
| **Covid-19 had an Impact on Technology Architecture within the organisation** | Agree | Agree | Strongly agree | Strongly agree | Agree |
| **We collaborate effectively with other teams online.** | Disagree | Agree | Strongly agree | Strongly agree | Strongly agree |
| **Effective communication is easy to handle in the online environment.** | Disagree | Agree | Disagree | Agree | Agree |
| **Building relationships has become easier whilst we are online.** | Strongly disagree | Agree | Strongly disagree | Strongly disagree | Disagree |
| **Work-life balance was easy to handle.** | Disagree | Agree | Strongly disagree | Strongly disagree | Agree |
| **Managing distractions whilst in an online work environment has not been a problem.** | Agree | Strongly agree | Agree | Strongly agree | Agree |
| **Security was one of the sectors that were severely impacted by COVID-19.** | Agree | Agree | Disagree | Strongly agree | Agree |
| **My company was well suited to deal with security threats that presented themselves due to working from home.** | Agree | Strongly agree | Strongly agree | Strongly agree | Agree |
| **I was skilled enough to deal with the cyber threats during COVID-19** | Agree | Strongly agree | Strongly agree | Strongly agree | Agree |
| **Security protocol changed a substantial amount during COVID-19.** | Agree | Agree | Agree | Strongly agree | Agree |
| **The organisation experienced an increase in cyber-attacks During COVID-19.** | Disagree | Strongly Disagree | Strongly Disagree | Disagree | Disagree |
| **There are more security threats while working within an online environment.** | Agree | Disagree | Strongly agree | Agree | Agree |
| **Employees had the training to identify and prevent cyber security attacks and risks.** | Agree | Agree | Strongly agree | Agree | Disagree |
| **Employees used their own devices (WIFI, computers) while working from home** | Agree | Strongly Disagree | Strongly Disagree | Strongly agree | Agree |
| **Various cyber security tools (anti-virus, firewall) were used in the employee's home and work environment** | Agree | Strongly agree | Strongly agree | Agree | Agree |
| **The software used is regularly updated to prevent cyber attacks** | Agree | Strongly agree | Strongly agree | Agree | Agree |
| **The employees have anti-spyware software installed on their devices.** | Agree | Strongly agree | Strongly agree | Agree | Agree |
| **What additional security was necessary to be implemented to secure working from home?** | VPNs | VPN | Better VPN platforms and upgraded firewalls | N/A | VPN, 2FA |

From the survey results from the five South African companies, the following have been determined:

Agile (Scrum) is the dominant software development methodology, although others followed additional software methodologies such as Waterfall and DevOps, or hybrid methodologies. More than half of the companies indicated that there is a clear distinction between their project management and software management methodologies. Agile appears to be the dominant project management methodology according to the survey results.

COVID-19 also had an enormous impact on these companies. They had to start working on their projects remotely. It also appears that post-COVID-19, companies preferred to switch to a hybrid workplace model between working remotely and being on site. Most of the companies agreed that COVID-19 impacted the way companies manage projects and develop software. COVID-19 also impacted the organisational productivity of these companies. Most of the companies agreed that COVID-19 had a positive impact on project management productivity. All the companies agreed that the pandemic had a positive effect on software development processes. More than half of the companies agreed that COVID-19 had a positive impact on employees' productivity. Employees seem to prefer remote working environments. All companies agreed that COVID-19 had an enormous impact on project management and the development of software products. Due to this fact, it affected the types of technologies employed by companies to develop software and enabling communication between development teams. COVID-19 impacted the way these companies govern projects and strategize. More than half of the companies agreed that COVID-19 had a positive effect on company culture and maturity. COVID-19 had a positive impact on company infrastructure and service deliveries within the companies.

COVID-19 enabled teams to collaborate more efficiently in a remote work environment. Even though communication has seemed to improve, it appears that most employees indicated that they found it more challenging to build personal work relationships given the remote work circumstances. The remote work environment also brought some other issues: most employees found it challenging to balance their work and personal life. Security also seems to be an issue that become more prevalent during COVID times. Even though security became a bigger threat, it appears that all of the companies and employees were well suited to combat security threats. A reason companies might have been able to combat security threats more efficiently was due to security protocol updates to adapt to the COVID-19 pandemic. It also appears that the companies experienced few to no cyber-attacks during the pandemic. Security threats were a bigger issue during the COVID-19 pandemic, however. More than half of the companies indicated that employees used their own resources to finish and collaborate on projects remotely. More than half of the companies indicated that employees utilized cyber security tools in their home and work environment. Software appears to be maintained regularly to avoid vulnerabilities against cyber-attacks. Other tools such as VPNs and 2FA were utilized to ensure that employees work in a safe cyber environment.

# **CHAPTER 9: CONCLUSION**

The following deductions were made based on the research from the article and survey results:

Although other software development approaches, including Waterfall and DevOps, as well as hybrid methodologies, were also used, Agile (Scrum) is currently the most popular. There is a notable difference between their project management and software management techniques, according to more than half of the firms. The study results suggest that agile project management is the most common methodology.

Although other software development approaches, including Waterfall and DevOps, as well as hybrid methodologies, were also used, Agile (Scrum) is currently the most popular. There is a notable difference between their project management and software management techniques, according to more than half of the firms. The study results suggest that agile project management is the most common methodology.

COVID-19 also had a significant effect on these businesses. They had to begin conducting their project work remotely. Additionally, it seems that businesses choose to adopt a hybrid workplace model that combines working remotely and on-site labour after COVID-19. The majority of businesses concurred that COVID-19 had an impact on how they manage projects and create software. The organizational productivity of these businesses was similarly impacted by COVID-19. Employees appear to prefer remote working settings. All businesses concurred that COVID-19 had a significant impact on software product development and project management. As a result, it had an impact on the kinds of technologies used by businesses to create software and facilitate communication between development teams. COVID-19 has an impact on how these businesses plan and manage their projects.

Teams were able to operate more productively together in a remote setting thanks to COVID-19. Even while it appears that communication has improved, the majority of employees seem to have stated that the remote work environment makes it harder to forge individualized working relationships. The majority of employees found it difficult to manage their job and personal lives, which was another challenge brought on by the remote work environment. During the COVID era, security also appears to be a problem that becomes more widespread. Even if security concerns increased, it seems that all of the businesses and personnel were well-equipped to manage them. Due to security protocol modifications to address the COVID-19 pandemic, businesses as well as employees may have been better prepared to tackle security risks.

It would appear that during the modern era, the Agile (Scrum) methodology is preferred to manage software projects, due to being adaptable and iterative, as well as taking changing project requirements into account. Traditional software development methodologies like Waterfall are bust to fade away, due to its rigid structure and substantial number of formal procedures. It would appear, however, that most large companies prefer to employ a traditional software development methodology or a hybrid selection of methodologies to manage projects.

One needs to keep in mind that the survey results are based on only five companies, and therefore does not provide an accurate overview of software management methodologies, since it is an exceedingly small sample of companies within the industry. It is also important to keep in mind, even though Agile (Scrum) is the most preferred software development methodology, at the end of the day it is not the standard and depends on the size of the company and the projects that is being worked on by those companies.

# **BIBLIOGRAPHY**

Bassil, Y. 2012. A simulation model for the waterfall software development life cycle. *International Journal of Engineering & Technology (iJET)*, 2(5), <https://arxiv.org/ftp/arxiv/papers/1205/1205.6904.pdf> Date of access: 15 Oct 2022.

Chiyangwa, T.B. & Mnkandla, E. 2017. Modelling the critical success factors of agile software development projects in south africa. *South African Journal of Information Management*, 19(1):1-8. <https://journals.co.za/doi/pdf/10.4102/sajim.v19i1.838> Date of access: 14 Oct 2022.

De Carvalho, B.V. & Mello, C.H.P. 2011. Scrum agile product development method-literature review, analysis and classification. *Product: Management and Development*, 9(1):39-49. <http://app.periodikos.com.br/article/10.4322/pmd.2011.005/pdf/pmd-9-1-39.pdf> Date of access: 15 Oct 2022. 10.4322/pmd.2011.005

Despa, M.L. 2014. Comparative study on software development methodologies *Database Systems Journal*, 5(3):37-56. <http://www.dbjournal.ro/archive/17/17_4.pdf> Date of access: 10 May 2022.

Jones, C. 2017. *Software methodologies: A quantitative guide*. 6000 Broken Sound Parkway NW, Suite 300

Boca Raton, FL 33487-2742: CRC Press - Taylor and Francis Group.

Mnkandla, E. 2009. About software engineering frameworks and methodologies*.* In. AFRICON 2009. IEEE. pp. 1-5.

Moyo, B., Huisman, M. & Drevin, L. 2022. The state of systems development methodologies use within the systems development organisations in south africa. *South African Computer Journal*, 34(1):103-123. <http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2313-78352022000100006> Date of access: 15 Oct 2022.

Simão, E.M. 2011. *Comparison of software development methodologies based on the swebok.* University of Minho (Portugal) School of Engineering. <https://core.ac.uk/download/pdf/55628173.pdf> Date of access: 15 Oct 2022.

Yarlagadda, R.T. 2021. Devops and its practices. *International Journal of Creative Research Thoughts (IJCRT)*, 9(3):2320-2882. <https://www.researchgate.net/profile/Ravi-Teja-Yarlagadda/publication/350006289_DevOps_and_Its_Practices/links/604b037145851543166f3b6e/DevOps-and-Its-Practices.pdf> Date of access: 16 Oct 2022.