Assessing the Association between Agile Maturity Model Levels and Perceived Project Success

Thesis Presented to the Department of Information Systems

University of Cape Town

By

Vaughan Henriques (HNRVAU001)

Supervised by Associate Professor Maureen Tanner

In partial fulfilment of the requirements

for the

Masters of Commerce in Information Systems (INF5005W)

2017

PLAGIARISM DECLARATION

1. I know that plagiarism is a serious form of academic dishonesty.
2. I have read the document about avoiding plagiarism, am familiar with its contents and have avoided all forms of plagiarism mentioned there.
3. Where I have used the words of others, I have indicated this by the use of quotation marks.
4. I have referenced all quotations and properly acknowledged other ideas borrowed from others, using APA version 6 referencing standard.
5. I have not and shall not allow others to plagiarise my work.
6. I declare that this is my own work.

Signature: Signed by candidate Date: 28 February 2018

Full Name of Student: Vaughan Henriques Student Number: HNRVAU001

ABSTRACT

The underlying philosophy of the agile manifesto is embodied in principle one which promotes the continuous delivery of software that is deemed valuable by the customer, while principle twelve encourages continual improvement of the delivery process. This constant improvement, or maturity, is not a concept unique to agile methods and is commonly referred to as a maturity model. The most common of maturity model is the Capability Maturity Model Integrated (CMMI). However, research consensus indicates CMMI is incompatible with agile implementation, specifically at higher levels of maturity without sacrificing agility. Agile maturity models, which are aligned to agile principles encourage continuous improvement while maintaining agility. Given the underlying philosophy of the agile manifesto, this research hypothesises that an increase in agile maturity is associated with improved perceived project success, by using a conceptual model based on an existing agile maturity model and how each of the maturity levels are related to the perceived project success. The research also brings to light the concept of perceived project success, showing success in an agile environment is a subjective concept. Conducted quantitatively, the findings of this research show which specific focus areas within each of the maturity levels is most strongly correlated with perceived project success and concludes an increasing correlation between the maturity levels and perceived project success.

DEDICATION

First and foremost, I would like to dedicate this work to my personal Lord and Saviour, Jesus Christ, without whom none of this would be possible. Then, to my supportive wife, Jenine and daughters, Taryn and Leah who afforded me the opportunity and allowing me time away from our precious family life to commit to this study – thank you for continued love and support in everything I do.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The 2015 version of the chaos report by the Standish Group continues to report high project failure rates for Information Technology (IT) projects, using traditional success measures of scope, resource and schedule. When traditional (waterfall) methods are utilised, an eleven percent success rate is reported (Hastie & Wojewoda, 2015). Waterfall struggles to deliver in situations where IT project requirements are not fully defined (Hastie & Wojewoda, 2015) and continues to struggle to deliver in environments where requirements are constantly changing (Serrador & Pinto, 2015).

As a result of the ever-changing environment and high software project failure rates, the agile manifesto was created in 2001, containing a set of four value statements and twelve principles for agile methodologies known as the agile manifesto (Fowler & Highsmith, 2001). The agile manifesto is a guide for agile methodologies in achieving higher project success rates.

A recent (2016) survey reports the primary drivers of agile adoption in organisations as, improved software delivery, meeting customer requirements, the ability to manage changes in priorities, increased productivity and accelerated delivery (VersionOne, 2016). These drivers align with principle one of the agile manifesto which focuses on the continuous delivery of quality software of value to customer, which can be viewed as the underlying philosophy for agile development (Fowler & Highsmith, 2001).

Compared to their waterfall counterparts, agile projects continue to report a higher success rate of thirty-nine percent (Hastie & Wojewoda, 2015). The research focus on critical success factors for agile projects has also increased and Chow and Cao (2008) conclude that the only truly critical success factors are related to the principles defined in the agile manifesto. The manifesto is not prescriptive of the implementation method employed and as such a number of agile methodologies have come into being such as Scrumban, Kanban, Lean Development, Feature-Driven-Development (FDD), eXtreme Programming (XP) and Scrum, with the latter being the most popular (VersionOne, 2016). Regardless of methodology, adherence to the agile values and principles should improve the success of software implementations.

Principle twelve of the agile manifesto encourages teams to continually strive to improve the process of software delivery. In an effort to attain this constant improvement organisations typically utilise maturity models such as Software Process Improvement (SPI) frameworks. A maturity model is a predefined process improvement model for improving a desired outcome (Fontana, Meyer, Reinehr, & Malucelli, 2015). The most popular SPI currently used is the

Capability Maturity Model Integrated (CMMI) (Leppänen, 2013). However, given its historical background the CMMI is not aligned with agile principles and does not lend itself to agility (Leppänen, 2013). The incompatibility between CMMI and agile methods is further explained in section 2.3.2.1. In an effort to maintain agility, research has explored the concept of an agile principle-based maturity model.

Patel and Ramachandran (2009a) propose an Agile Maturity Model (AMM) which is based on agile principles. The AMM proposes a five-level model of increasing maturity with key agile process focus areas at each level. Each maturity level fulfils at least one of the agile principles and corresponds to a more mature agile implementation. Since the agile manifesto was created to improve project success higher levels of the agile principle-based AMM is expected to improve project success rate.

1.1 Problem Statement and Research Purpose

SPI frameworks such as CMMI and SPI and Capability Determination (SPICE) (Schweigert, Vohwinkel, Korsaa, Nevalainen, & Biro, 2014) are utilised to achieve continual improvement in the software delivery methodology being utilised. CMMI remains the most commonly used in corporate environments (Leppänen, 2013), providing a pre-defined delivery process for ensuring process maturity and has been shown to improve project success (Humble & Russel, 2009) in waterfall environments (Galin & Avrahami, 2006) using tradtional project success measures of scope, resource and schedule (McLeod, Doolin, & MacDonell, 2012). A number of studies highlight the misalignment between agile and CMMI (Fritzsche & Keil, 2007; Łukasiewicz & Miler, 2012; Marçal, et al., 2008; Potter & Sakry, 2009). Organisations wish to leverage the investment in CMMI (Leppänen, 2013) even though higher levels of CMMI maturity have been shown to be incompatible with agile methods (Fritzsche & Keil, 2007).

In general, the intent of a maturity model is the continual improvement of a desired outcome (Fontana, Meyer, Reinehr, & Malucelli, 2015) and since the underlying philosophy of agile is the delivery of quality software it can be logically concluded that an increase in agile maturity should relate to an increase in the perceived success of a project. A review of the current literature highlights a notable lack of research relating improved maturity in an agile maturity model to project success. While studies exist which relate the maturity levels of CMMI to improved project success, Gren, Torkar and Feldt (2015) conclude it would be useful to perform such a study in the context of an agile maturity model. Without an empirically validated agile maturity model (Gren, Torkar, & Feldt, 2015), there is no guide for practitioners to reference which agile processes in the AMM will increase the project success rate. Though

Patel and Ramachandran (2009a) propose an agile principle based maturity model, research has not yet been conducted to investigate whether higher AMM maturity relates to improved perceived project success. Thus at present there is no agile alternative to the CMMI.

Addressing the stated research problem, the purpose of this research is to ascertain whether improving (maturing) in agile discipline is associated with improved perceived project success. The research is intended to provide empirical evidence independent of the agile method, industry or organisation in which it is being applied.

1.2 Research Objective and Questions

The primary objective of this research is to provide empirical research which investigates the association between an agile maturity model and the perceived project success, whilst providing an explanation of these associations. Achieving the objective will address the problem previously mentioned and provide practitioners with a guide of which AMM process areas will sustain an increase in the perceived project success rate as the agile implementation matures.

Literature evidences the underlying intent of a software maturity model is the continual improvement of the software delivery process (Fontana, Meyer, Reinehr, & Malucelli, 2015). Furthermore, since the underlying philosophy of agile methodologies is continued customer satisfaction *"through early and continuous delivery of valuable software"* (Fowler & Highsmith, 2001, p. 34), it is expected higher maturity levels in an agile maturity model should improve perceived project success.

This research aims to answer the primary research question "How are the different maturity levels of the Agile Maturity Model (AMM) as proposed by (Patel & Ramachandran, 2009a) associated with perceived project success?". The related sub question being "How are the specific process areas of the different maturity levels in the AMM associated with perceived project success?".

2 LITERATURE REVIEW

This section focuses on current related literature relating to maturity models (both traditional and agile), project success and agile methods and is structured as follows; it begins with an overview of agile methodologies, highlighting the underlying philosophy of agile and how the term "team" needs to be considered in an agile environment. The concept of perceived project success is then explored and a suitable working definition adopted from literature. Thereafter maturity models are introduced and the most popular maturity model, the CMMI, is explained. The suitability of CMMI in an agile environment is examined as well as the relation between CMMI and project success. Current work undertaken in agile maturity models and work relating maturity models to project success is explored, highlighting the gap in current research. An AMM is then selected for this research and explained in more detail, highlighting its alignment to the agile principles.

2.1 Agile Methodologies and Principles

Early in the new millennium, a group of seventeen software industry practitioners assembled to address the prevalent problems faced by software projects (What is Agile?, n.d.), the outcome of which was the agile manifesto (Fowler & Highsmith, 2001). This is largely considered the birth of the modern agile development movement.

The agile manifesto, is a collection of four value statements "*Individuals and interactions over processes and tools", "Working software over comprehensive documentation", "Customer collaboration over contract negotiation"* and *"Responding to change over following a plan"* (Fowler & Highsmith, 2001, p. 29) which while not discounting the items on the right, preference is given to items on the left (Fowler & Highsmith, 2001).

The agile manifesto value statements are supported by twelve principles which focus on collaboration between stakeholders, continuous feedback cycles and delivery to the customer. Studies focussing on critical success factors in agile development confirm the relation between adherence to the principles and improved project success (Chow & Cao, 2008).

2.1.1 Underlying Agile Philosophy

The agile manifesto is intended as a guide for software development projects to be more responsive to changing business requirements whilst continuously delivering quality software. This is underpinned by principle one, *"Our highest priority is to satisfy the customer through* *early and continuous delivery of valuable software"* (Fowler & Highsmith, 2001, p. 30), which can be viewed as the driving philosophy for agile methodologies.

2.1.2 Agile Methods

Though the conceptualisation of the agile manifesto is considered the birth of modern agile methodologies, it served as a consolidation of the principles and values being applied in major iterative development methodologies. The origin of agile methods stems back to Iterative and Incremental Design and Development (IIDD) dating to as early as the mid nineteen fifties (Glazer, Dalton, Anderson, & Konrad, 2008). Principally based on Deming's Plan-Do-Check-Act (PDCA) cycle (Glazer, Dalton, Anderson, & Konrad, 2008) originally developed in the nineteen thirties (Johnson, 2002), IIDD implemented a continuous process improvement approach in which feedback and collaboration were continuously sought in developing software or a product to ensure customer satisfaction and improvement (Johnson, 2002).

IIDD was adopted by various organisations during the nineteen seventies and as software development became more mainstream and matured in the corporate environment, the more modern-day variants of agile, such as eXtreme Programming (XP), Scrum, Feature Driven Development (FDD), Crystal and the Rational Unified Process (RUP) came into being in the nineteen nineties (Glazer, Dalton, Anderson, & Konrad, 2008). Ultimately, in February 2001, the thought leaders behind these methodologies congregated in Snowbird Utah to compile the agile manifesto (Glazer, Dalton, Anderson, & Konrad, 2008).

Having its origins in the PDCA cycle, the agile manifesto does not prescribe a specific methodology but instead provides a set of guiding values and principles. Understandably, a recent (2016) worldwide survey of agile methods and practices shows thirteen agile methodologies being used in practice with Scrum being most predominant (VersionOne, 2016). For research which aims to look at agile holistically, the focus cannot be limited to a specific agile methodology.

2.1.3 Agile Teams Including Business Representative

The agile principles strongly support and promote the concept of a cross-functional, self-organised team (Fowler & Highsmith, 2001). Typically, agile teams consist of five to nine people, having all the roles required to deliver the software (Papadopoulos, 2015). This allows the team to be self-sufficient and translates to an improvement in project success (Stettina & Hörz, 2015). This is consistent with the 'whole team' concept from XP in which *"people with* *all the skills and perspectives necessary for the project to succeed"* (Beck & Andres, 2004, p. 38) are included.

Critically, besides the presence of technical competencies such as architecture, design, development, testing, database administration and project management, agile teams include roles for business representation (Beck & Andres, 2004) aligned with agile principle four. Consensus in research has shown that the role of business representation as part of the team is critical to the success of an agile project (Chow & Cao, 2008; Sverrisdottir, Ingason, & Jonasson, 2014; Tanner & von Willingh, 2014) and forms the vital link between the business, customer and end user and the development team (Ambler, 2012b). Sverrisdottir, Ingason and Jonasson (2014) highlight the criticality of the business representative in the agile team setup in obtaining feedback from end users of the software being delivered, providing the development team a sense of the satisfaction of the delivery (Sverrisdottir, Ingason, & Jonasson, 2014). The naming convention for the business representative, typically the product owner, varies across different agile methodologies. Ambler (2012b) highlights that there is a distinction between a job title or position and a role in an agile environment, the latter being the skill or combination of skills required to achieve the desired outcome. When referring to a team or team member in the agile context, job titles and position become irrelevant, providing the individual is fulfilling at least one of the aforementioned roles.

2.1.4 Agile Teams and Business Feedback

The agile manifesto repeatedly reinforces the concept of feedback and collaboration in both the value statements and principles, stressing the primary measure of success being the delivery of software. Principles two *"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage"* and four *"Business people and developers must work together daily throughout the project"* (Fowler & Highsmith, 2001, p. 35) encourages daily feedback from the customer while functionality is being developed, to ensure the delivery of business value. Studies have found a lack of customer feedback can cause a loss to business (Hoda, Noble, & Stuart, 2011) due to the development focus being misaligned with the business priority or the required functionality being misinterpreted (Sverrisdottir, Ingason, & Jonasson, 2014). Besides this continuous customer feedback during the development iteration, principle twelve of the agile manifesto "*At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly*." (Fowler & Highsmith, 2001, p. 35), encourages scheduled feedback sessions with the broader business. This is formalised in agile methods as an opportunity for the development team to demonstrate new functionality developed, allowing the broader business to provide feedback on satisfaction directly to the agile team (Gonçalves & Linders, 2013). Literature thus indicates the agile team (inclusive of the business representative) is attuned with the stakeholders' satisfaction with delivery.

2.2 Perceived Project Success

A number of studies have been conducted (Geoghegan & Dulewicz, 2008; Joosten, Basten, & Mellis, 2011; Jugdev & Müller, 2005; McLeod, Doolin, & MacDonell, 2012; Przemysław, 2013; Raymond & Bergeron, 2008) on the topic of project success. Thomas and Fernández (2008) introduce the topic by equating it to the capturing of Proteus, the mythical sea god of elusive sea change. The popularity of this topic in academic literature is evidence of the protean nature of project success.

Jugdev and Müller (2005) look at the evolution of the concept and the associated changes in beliefs on how this is to be measured over a forty-year period. Customary measures are based on the 'Iron-Triangle' of project management (see Figure 1), the dimensions being scope, resources and schedule. Whilst these dimensions are the fundamentals upon which projects are executed and reported (McLeod, Doolin, & MacDonell, 2012), there is agreement of an important distinction to be made between project management success and project success (McLeod, Doolin, & MacDonell, 2012).

The former refers to proper management of the three dimensions of the iron-triangle; whilst the latter is a more fluid, perceived and subjective concept. An often-quoted example depicting this is the Sydney Opera House, which cost fourteen times the original budget and took fifteen years to complete (Jugdev & Müller, 2005). Assessing only the time and budget aspects, this would be considered failure in the project management sense, yet it is considered a landmark achievement in architectural terms, a project success. This non-interrelation of the two measures is further supported by research work conducted by Joosten, Basten and Mellis (2011) who conclude, though projects continue to be reported on using the iron-triangle measures, project decision makers continue to use context specific subjective measures to determine the success of a project.

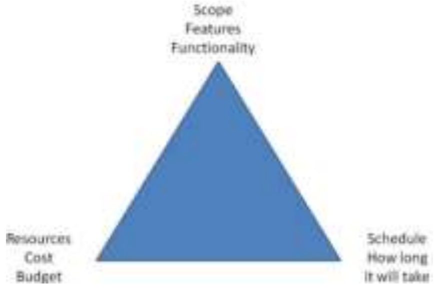

Figure 1: The traditional "Iron-Triangle" of project management (Highsmith, 2004)

A study by Thomas and Fernández (2008) focussing on subjective (perceived) project success was not able to distil a single definition of the concept. This view is further supported by McLeod, Doolin and MacDonell (2012) who further highlight that stakeholders within a project could evaluate the success of the project differently, based on their perspectives, perceptions and context for the evaluation. This is consistent with Jugdev and Müller (2005) who highlight that different line managers involved in the same project, could perceive success to greater or lesser extent based on its contribution towards achieving an overall business goal. Simplistically reproduced here, Figure 2 depicts how stakeholder perceptions influence the evaluation of the success of a project, i.e. the perceived success of a project is dependent on the evaluator and their perception of the value contributed.

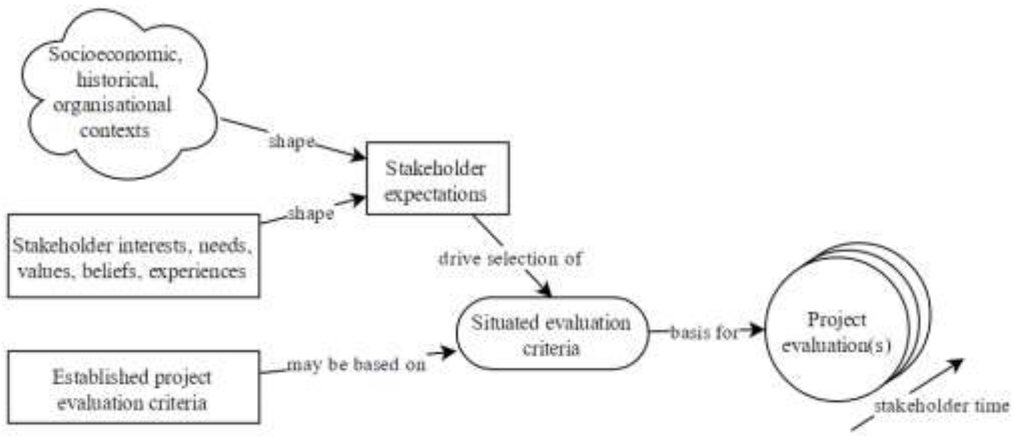

Figure 2: Stakeholder perspective can influence project success evaluation over time (McLeod, Doolin, & MacDonell, 2012)

Highsmith (2004) proposes the iron-triangle in the agile project world, see Figure 3. Whilst the traditional 'iron-triangle' remains, it serves only as a constraint in the agile-iron-triangle, with the latter focussing on value and quality delivery. Both the value and quality form part of the subjective measures, with quality being *"the most subject to variation in perception by multiple project stakeholders"* (Prabhakar, 2009, p. 7). Agile methods have altered the traditional view on project success, with the focus shifting more towards stakeholder satisfaction (Leppänen, 2013).

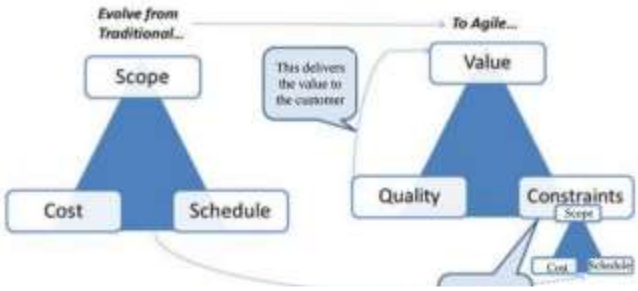

Figure 3: Agile Iron-triangle (Highsmith, 2004)

In a study researching the relation between agile planning efforts and project success, Serrador and Pinto (2015) divide the *"Overall Project Success"* between the traditional iron-triangle *"Project Efficiency"* measures and the perceived *"Stakeholder Success"* (Serrador & Pinto, 2015, p. 1043) aspects. The stakeholder success aspect relates to the value and quality dimensions of the agile iron-triangle, i.e. the subjective perceived project success. Using factor analysis, Serrador and Pinto (2015) conclude the questions relating to stakeholder success are a better indicator of perceived project success. The questions utilised for measuring perceived project success was a combination of three dimensions being perception of the project teams' satisfaction with the deliverables, the teams' perception of the clients' satisfaction with the deliverables and the teams' perceptions of the end users' satisfaction with the deliverables (Serrador & Pinto, 2015).

Given the context and perspective sensitivity of project success, current literature and research in this field shows it is based on perception and thus a subjective measure. In the context of this study the working definition for perceived project success will be as defined by the definition of stakeholder satisfaction (Serrador & Pinto, 2015) as shown in Figure 4.

Perceived Project Success

Teams' Perceived Satisfaction with Deliverables

Teams' Perception of Clients' Satisfaction with Deliverables

Teams' Perception of End Users' Satisfaction with Deliverables

Figure 4: Perceived project success construct – defined as stakeholder satisfaction (Serrador & Pinto, 2015)

2.3 Maturity Models

2.3.1 Maturity Model Characteristics

Maturity is in reference to the software development process. Specifically, *"maturity is the extent to which a specific process is explicitly defined, managed, measured, controlled, and effective"* (Paulk, Curtis, Chrissis, & Weber, 1993, p. 20). A maturity model describes how a process can evolve (mature) over time. Each phase of evolution (maturity level), is a progressive step along an improvement path, improving the desired outcome (Fontana, Meyer, Reinehr, & Malucelli, 2015). With the first phase being the least mature and the last phase equating to optimal maturity, each maturity level defines the focus areas required and success criteria to be assessed to provide evidence of the maturity level being achieved (Leppänen, 2013). When fully matured, the process operates optimally (Leppänen, 2013) and is executed consistently, producing steady predictable outcomes (Paulk, Curtis, Chrissis, & Weber, 1993).

2.3.2 The Capability Maturity Model Integrated (CMMI)

As per Figure 5 (Team, 2010), in the mid-1980s the Department of Defence (DOD) in the USA commissioned an investigation into the recurrent poor performance in life-critical software projects. The projects were fulfilled by third party contractors in an inherently low trust environment (Glazer, Dalton, Anderson, & Konrad, 2008), which is in contradiction to the underlying trust elements in agile methods (McHugh, Conboy, & Lang, 2012). In response, the Software Engineering Institute (SEI) of the Carnegie Mellon University published the first version of the Capability Maturity Model (CMM) in 1991. CMM provided the DOD with a mechanism for assessing the software development process maturity of third party vendors, providing some reassurance for quality software being delivered on the basis that managing the process would improve the outcome (Glazer, Dalton, Anderson, & Konrad, 2008).

Once publicly available, practitioners produced a number of variants which were initially consolidated by SEI into version one of the CMM Integrated (CMMI) in 2000. Subsequent updates occurred with version 1.2 and version 1.3 being released in 2006 and 2010 respectively. The 2006 CMMI version 1.2 update saw the introduction of maturity models focussing on three different disciplines, termed constellations in CMMI parlance. The first of these constellations was specifically focussed on software development and officially named CMMI-Dev (Heffner, 2006; Kitson, Vickroy, Walz, & Wynn, 2009). Amongst other changes, the 2010 CMMI version 1.3 release included changes addressing implementation in agile environments (SEI, 2010).

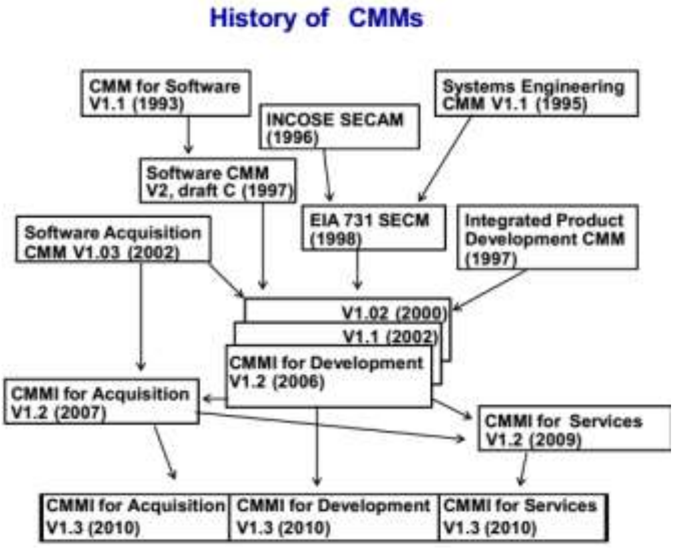

Figure 5: Diagrammatic history of the CMMI - (Team, 2010)

CMMI defines five levels of increasing maturity shown in Table 1 with the focus area showing the key improvement to be achieved at that level. The five maturity levels consist of twenty-two process areas. To attain a given level of maturity all the corresponding process areas need to be addressed, including the process areas of the lower levels (Team, 2010).

Table 1: CMMI maturity levels, focus and process areas (ProcessGroup, 2015)

Level	Focus	Process Area
5 - Optimising	Continuous Process Improvement	Causal Analysis and Resolution Organisational Performance Management
4 - Quantitatively Managed	Quantitative Management	Organisational Process Performance Quantitative Project Management

Level	Focus	Process Area
3 - Defined	Process Standardisation	Decision Analysis and Resolution
		Integrated Project Management
		Organisational Process Definition
		Organisational Process Focus
		Organisational Training
		Product Integration
		Requirements Development
		Risk Management
		Technical Solution
		Validation
		Verification
2 - Managed	Basic Project Management	Configuration Management
		Measurement and Analysis
		Project Monitoring and Control
		Project Planning
		Process and Product Quality Assurance
		Requirements Management
		Supplier Agreement Management
1 - Initial	Competent People and Heroics	

CMMI level one (initial) is solely reliant on competent individuals with no defined process in place (Team, 2010). As such it can be considered as a starting point for the maturity model, indicating a lack of any formal maturity being in place.

CMMI level two (managed) consists of seven process areas and although termed 'managed' the focus is dual purposed, both on establishing the practices and the initial management thereof (Team, 2010). Solely concerned at a project level, activities take guidance from the governing policies and procedures of the organisation and focus on defining and documenting the various activities (Team, 2010).

CMMI level three (defined) contains eleven process areas and is characterised by an effort to standardise the various processes across the organisation. The activities at this level are aimed at ensuring that standardised documentation for all processes has been properly communicated to the organisation to ensure consistency (Team, 2010). Whereas level two takes guidance from the organisation, a project aiming to achieve level three must adhere to the organisational standards and will tailor the process based on such. Documented processes at this level must adhere to an organisational minimum defined process documentation standard of *"the purpose, inputs, entry criteria, activities, roles, measures, verification steps, outputs, and exit criteria"* (Team, 2010, p. 28).

CMMI level four (quantitative management) focuses on project management activities, based solely on the quantitative measures established in lower levels of maturity, e.g. project monitoring and control in level two (Team, 2010). The focus can be at a project, process or organisational level (Team, 2010). If at the project level the quantitative inputs are primarily project efficiency and project management efficacy measures of the iron-triangle.

CMMI level five (optimizing) shifts the focus to continuous improvement at an organisational level. The key process areas being Organisational Process Management (OPM) and Causal Analysis and Resolution (CAR). The OPM is primarily concerned with advancing the organisation, incrementally if necessary and can be interpreted as strategic planning in the organisation. Activities which enable this process area to be achieved can utilise input from various sources, including academia and other external sources (Team, 2010). The CAR activities focus on root cause analysis and removal of impediments detrimental to the organisational outcomes. The key distinction at this level of maturity is the focus being solely on the organisation as a whole (Team, 2010).

It is evident from the focus and process areas that, besides level two activities which focus on the project delivery, the CMMI is aimed at achieving organisational maturity (Fritzsche & Keil, 2007).

2.3.2.1 Compatibility of CMMI and Agile Methods

Significant research effort has been invested in attempting to find a level of compatibility between CMMI and agile methods. Consensus exists in research literature that the co-existence of the higher levels (four and five) of CMMI maturity and agile is difficult to achieve without sacrificing the agility (Fritzsche & Keil, 2007; Łukasiewicz & Miler, 2012; Marçal, et al., 2008; Potter & Sakry, 2009).

Boehm and Turner (2005) highlighted the potential problem that the introduction of agile methods in mature organisations could affect the maturity ratings. Fritzsche and Keil (2007) and Łukasiewicz and Miler (2012) attribute the incompatibility to the nature of the two disciplines with CMMI focusing on the organisational level whereas the agile focus is the successful delivery of a project. This is consistent with research finding a decline in project success rates (Ambler, 2012a; Dingsøyr & Moe, 2014) when organisational elements such as corporate governance are considered (Laanti, 2014).

Nevertheless, research indicates a high degree of compatibility between agile methods and CMMI at maturity levels two and three. Łukasiewicz and Miler (2012) found a seventy percent exact or partial match between agile methods and CMMI levels two and three. In a similar study Fritzsche and Keil (2007) extend the mapping between the agile methods of Scrum and XP and CMMI levels two through five, finding no evidence of support at levels four and five in agile practices, again due to the organisational focus of these maturity levels.

Marçal, *et al*. (2008) focus their research on mapping Scrum to the project management activities of CMMI levels two, three and four finding sixty-five percent, forty-three percent and zero percent compatibility respectively. Mapping Scrum to the requirements management, project planning and process monitoring and control activities of CMMI levels two and three, Potter and Sakry (2009) find a satisfactory compatibility, although acknowledging complete absence in other more organisationally focussed process areas at these levels, such as supplier agreement management at level two. Even at this relatively low level of maturity the deficiency of agile methods in addressing organisation-wide activities is evident.

Research by Sutherland, Jakobsen and Johnson (2008) found the introduction of the Scrum methodology in a CMMI level five compliant organisation had the effect of successfully decreasing the amount of rework required. Whilst at first this might seem contradictory to literature, it is noted that the introduction of agile methods was after the organisation had attained the level five rating (Sutherland, Jakobsen, & Johnson, 2008). Thus, agile was not implemented in isolation to achieve the maturity rating.

Current research indicates using an agile method in isolation to achieve maturity *"levels higher than the third require some far-reaching compromises that significantly affect the benefits of agile methodologies"* (Łukasiewicz & Miler, 2012, p. 417), primarily due to change to an organisational focus at these higher levels. The use of CMMI in an agile environment can be counter-productive to the agility and continuous delivery being sought. When using CMMI to mature an agile method, *"the best improvement approach in an agile environment is to stop at CMMI level 3"* (Fritzsche & Keil, 2007, p. 24).

In summary, though CMMI is the most popular SPI framework (Leppänen, 2013) it has been shown to be inconsistent with agile methods, specifically at the higher levels of maturity (Fritzsche & Keil, 2007). CMMI and agile methods have been shown to be able to successfully co-exist (Sutherland, Jakobsen, & Johnson, 2008) but the majority of research findings indicate the application of CMMI is detrimental to the ultimate goals of agile methods.

2.3.2.2 CMMI and Project Success

Given the original intent of the CMMI, a positive relation between achieving higher levels of software development process maturity (as the independent variable) and software project success (as the dependent variable) is expected (Agrawal & Chari, 2007). Using this conceptual model, Rönkkö, Peltonen and Frühwirth (2011) investigated the relation between CMMI process maturity and the development efficiency, effectiveness and innovativeness in small companies, the latter three in combination being defined as the project success measures for a software product. Applying regression analysis to the model their findings did not support the expected relation between software development process maturity and project success due to the study being applied to small companies (Rönkkö, Peltonen, & Frühwirth, 2011).

Using a similar conceptual model, in organisations of different sizes using a waterfall methodology, Jiang, Klein, Hwang, Huang and Hung (2004) applied regression analysis and concluded that the utilisation of CMMI to improve software development maturity could improve project success rates. Similarly Galin and Avrahami (2006) concluded the investment in CMMI initiatives is beneficial for the productivity gains achieved. Existing research using the conceptual model of the CMMI maturity level as the independent variable and project success as the dependent variable confirms a positive relation between the two constructs in waterfall environments.

2.3.3 Maturity Models for Agile Environments

The application of agile principle twelve *"At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly"* (Fowler & Highsmith, 2001, p. 34) encourages the continuous improvement of the software delivery process which aligns with the intent of a maturity model (see section 2.3.1). Whilst the CMMI focuses on process maturity, given the people-centric and collaborative nature of agile environments, there is agreement that a CMMI equivalent agile maturity model should align to agile principles and practices (Fontana, Reinehr, & Malucelli, 2014; Gren, Torkar, & Feldt, 2015). This subsection reviews research work done in the area of agile maturity models and details the agile maturity model adopted for this study.

2.3.3.1 Current Research on Maturity Models in Agile Environments

A systematic literature review by Henriques and Tanner (2017), focussing on research themes conducted with agile maturity models as the independent variable found two major themes emerging. The two major themes being, "Agile/CMMI" focussing on *"adapting agile practices and principles to fit current software maturity models"* (Fontana, Meyer, Reinehr, & Malucelli, 2015, p. 89) and "Agile Maturity", which focus on maturity models based on agile principles, leading to improved agility (Leppänen, 2013), as shown in Figure 6. Due to the existing investment organisations have in CMMI (Leppänen, 2013) unsurprisingly, as depicted the majority of the articles (59%) focus on the "Agile/CMMI" topic.

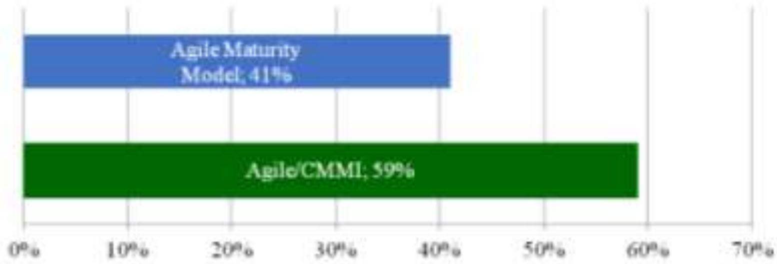

Figure 6: Primary research theme distribution involving agile maturity models (Henriques & Tanner, 2017)

The "Agile/CMMI" theme is further divided in two subthemes, being "What is the mapping between a given CMMI level and/or process area and agile practices and "Can agile methods and CMMI be used simultaneously?" (Henriques & Tanner, 2017). Examples of the coexistence of agile and CMMI are found in Fritzsche and Keil (2007), Glazer, Dalton, Anderson and Konrad (2008) and Łukasiewicz and Miler (2012). The latter subtheme is addressed in research focussed on mapping specific process areas of CMMI to agile practices. For example Marçal, *et al*. (2008) focus on mapping the project management process areas, Potter and Sakry (2009) map a number of process areas across maturity levels and Sutherland, Jakobsen and Johnson (2008) provide a case study of introducing Scrum into a CMMI level five organisation. The remaining 41% of the articles represent the major theme of Agile Maturity, with subthemes addressing agile process improvement, agile mature model assessment, proposing an agile maturity model, comparison between agile maturity models and agile adoption frameworks. The majority of these articles focus on proposing an agile maturity model (Henriques & Tanner, 2017).

Agile maturity models are proposed by a number of researchers, focussing either on a specific methodology (Scrum or XP) such as Nawrocki, Walter and Wojciechowski (2001) and Yin, da Silva and Figueiredo (2011) or on general agile practices by Ambler (2010) and Patel and Ramachandran (2009a). Agile adoption strategies, which are similar to maturity models but do not provide maturity levels, focus areas or process areas are provided by Ambler (2011), Lui and Chan (2005), Packlick (2007) and Qumer and Henderson-Sellers (2008) and Sidky, Arthur and Bohner (2007). Agility assessment studies are provided by Benefield (2010), Fontana, Reinehr and Malucelli (2014) and Gren, Torkar and Feldt (2015), providing guidelines for an organisation to assess alignment of the current agile implementation to agile principles and practices.

Fontana, Meyer, Reinehr and Malucelli (2015) conclude that because of the self-organisation of agile teams, maturity models can tend to become specific to a team and their unique experiences. Interestingly, the only research work found with agile maturity as the independent variable relating to project success as the dependent variable is by Rönkkö, Peltonen and Frühwirth (2011) as discussed in section 2.3.2.2 (CMMI and Project Success). The review of the literature highlights the lack of research relating improved maturity in an agile maturity model to project success and Gren, Torkar and Feldt (2015) conclude it would be useful to perform such a study.

2.3.3.2 Analysis of Proposed Agile Maturity Models

Fontana, Meyer, Reinehr and Malucelli (2015) attribute the first agile maturity model to Nawrocki, Walter and Wojciechowski (2001). A number of agile maturity models currently exist either explicitly by name, i.e. "Maturity Model" or provide a maturity assessment and adoption framework. Examples of explicit maturity models are proposed by Ambler (2010), Nawrocki, Walter and Wojciechowski (2001), Patel and Ramachandran (2009a) and Yin, da Silva and Figueiredo (2011). Benefield (2010), Fontana, Reinehr and Malucelli (2014), Lui and Chan (2005), Packlick (2007) and Qumer and Henderson-Sellers (2008) and Sidky, Arthur and Bohner (2007) (Fontana, Meyer, Reinehr, & Malucelli, 2015; Leppänen, 2013) provide maturity assessment and adoption frameworks.

Though each of the models provide between three and six maturity levels, there are shortcomings (Fontana, Meyer, Reinehr, & Malucelli, 2015; Leppänen, 2013) when viewed relative to the characteristcs mentioned in section 2.3 or in the scope to which they can be applied, summarised in Table 2.

The maturity models proposed by Benefield (2010), Lui and Chan (2005), Nawrocki, Walter and Wojciechowski (2001) and Yin, da Silva and Figueiredo (2011) are limited to either a Scrum or XP agile methodology. As mentioned in section 2.1.2 (Agile Methods), agile methods are broader than only these methodologies and an agile maturity model would need to cater for all agile methods by being based on agile principles and not solely on the practices of a specific methodology. Similarly the maturity model proposed by Packlick (2007) is too narrowly focussed with its development being limited for a specific company (Fontana, Meyer, Reinehr, & Malucelli, 2015).

The models proposed by Ambler (2010), Fontana, Reinehr and Malucelli (2014) and Qumer and Henderson-Sellers (2008) provide no focus areas or associated success criteria (Fontana, Meyer, Reinehr, & Malucelli, 2015; Leppänen, 2013) which are intrinsic characteristics of a maturity model (Leppänen, 2013). The model provided by Sidky, Arthur and Bohner (2007) provides a framework for the assessment of agility and how to progress to the next level (Leppänen, 2013) and provides a *"four-stage process for agile adoption"* (Fontana, Meyer, Reinehr, & Malucelli, 2015, p. 91) and thus not a maturity model.

Table 2: Summary of agile maturity models relative to maturity model characteristics and scope (Fontana, Meyer, Reinehr, & Malucelli, 2015; Leppänen, 2013)

Maturity Model	Criticism
Ambler (2010)	Agile adoption framework No assessment criteria No success criteria
Benefield (2010)	Limited to XP Limited to British Telecom
Fontana, Reinehr and Malucelli (2014)	No success criteria defined No ability to assess
Lui and Chan (2005)	Limited to XP No success criteria defined No ability to assess
Nawrocki, Walter and Wojciechowski (2001)	Limited to XP

Maturity Model	**Criticism**
	No success criteria
Packlick (2007)	Limited to Sabre Airline Solutions
Qumer and Henderson-Sellers (2008)	No success criteria defined No ability to assess
Sidky, Arthur and Bohner (2007)	Agile adoption framework
Yin, da Silva and Figueiredo (2011)	Limited to Scrum Limited ability to assess

In the analysis of the agile maturity model by Patel and Ramachandran (2009a) the conclusion is it *"exemplifies how to perform the assessessment"* (Fontana, Meyer, Reinehr, & Malucelli, 2015, p. 91). Thus applying the characteristics of maturity models mentioned in section 2.3.1, the requirement to be applicable across any agile method and the ability to be assessed, the working definition of an Agile Maturity Model is as proposed by Patel and Ramachandran (2009a) is adopted for this research.

2.3.3.3 Agile Maturity Model (AMM)

Patel and Ramachandran (2009a) propose a five-level maturity model for agile process improvement, called the Agile Maturity Model (AMM), as shown in Figure 7, based on agile practices and principles. As mentioned in section 2.3.3.2, the AMM displays all the characteristics of a maturity model, is applicable across agile implementations and provides assessment criteria. Similarly, to CMMI each of the maturity levels show the predefined focus areas for the level. This section explores the AMM in detail, highlighting the alignment of the model to agile principles as well as detailing each of the primary focus areas at each level of maturity.

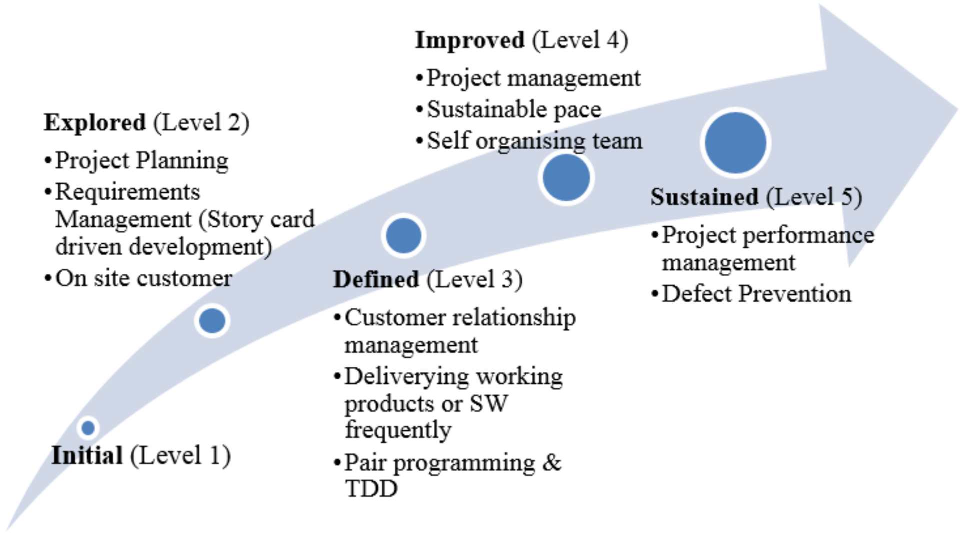

Figure 7: 5 Level AMM for Agile SPI (Patel & Ramachandran, 2009a)

2.3.3.3.1 AMM Levels and Focus Areas

Level One - Initial

The initial level of the AMM is characterised by being dependent on heroic efforts with no specifically defined process in place (Tarnowski, 2014). Outcomes are not repeatable and there is no alignment to agile principles. This level is equivalent to level one of the CMMI.

Level Two - Explored

Level two of the AMM activities covers the initial set of focus areas which organisations implement to establish agile practices (Patel & Ramachandran, 2009a). The focus is on planning based on developer estimates and requirements management in the form of story cards for the current iteration. The customer is present onsite but not necessarily always available for the project team (Patel & Ramachandran, 2009a). This level shows strong alignment with agile principles two, four, six and ten.

The AMM requires the presence of the customer to be available daily to the agile team in a decision-making capacity to direct development efforts (Patel & Ramachandran, 2009a), which aligns with principle four of the agile manifesto (Fowler & Highsmith, 2001). Tanner and von Willingh (2014) confirm previous research on knowledgeable and empowered customer presence being a success factor for agile implementations which should include the tracking of the development progress (Patel & Ramachandran, 2009a). Boehm and Turner (2003) refer to these business stakeholders as being “CRACK” representatives (Collaborative, Representative, Authorised, Committed, and Knowledgeable), who are able to channel the proper business knowledge, provide appropriate feedback and make decisions. Lack of proper customer presence and collaboration has been shown to affect prioritisation, clarity of requirements, loss of productivity and *"in extreme cases, Business Loss"* (Hoda, Noble, & Stuart, 2011, p. 527). The presence of the customer is vital in the prioritization of requirements (Sverrisdottir, Ingason, & Jonasson, 2014) allowing the development to focus only on functionality important to business, aligned with agile principle ten (Fowler & Highsmith, 2001).

The requirement is explained to the development team by the customer at the onset of an iteration (Patel & Ramachandran, 2009a), aligning to agile principle six (Fowler & Highsmith, 2001) and captured by the customer, in a properly structured and standardised story card representation (Patel & Ramachandran, 2009b). At this level of maturity, the story card needs to be explanatory enough to derive the acceptance criteria for the desired functionality. The story card is used only as a guideline and is allowed to change within the iteration (Sverrisdottir, Ingason, & Jonasson, 2014), aligned with agile principle two (Fowler & Highsmith, 2001). Sufficiently detailed story cards allow the proper detail breakdown of the tasks required to implement the requirement (Vlaanderen, Jansen, Brinkkemper, & Jaspers, 2011), which allows for more accurate developer-based estimates and thus improved project or iteration planning. This form of agile requirement management has been shown to be a critical success factor for agile implementations (Chow & Cao, 2008).

Estimation in an agile environment takes the form of planning poker, which is a consensus-based technique similar to using wide-band Delphi estimation (Gandomani, Wei, & Binhamid, 2014). The technique relies on the collective knowledge of the development team based on an estimate-discuss-estimate cycle, which has been shown to produce more accurate and reliable estimates (Surowiecki, 2004) under the proper conditions. Though relying on a mix of expertise, the tendency is for less optimistic, i.e. more realistic outcomes (Mahnič & Hovelja, 2012) which is vital for establishing a proper implementation plan and tracking the development velocity of the team.

Level Three - Defined

Having established the agile practices in level two, level three shifts the focus to better defining the specific agile implementation (Patel & Ramachandran, 2009a) focussing on the use of technical and technological aspects of the implementation. This level is characterised by increased customer relationship management through increased customer presence and customer satisfaction, through constant feedback aligning with principles four and six. Using more collaborative development practices such as pair programming and test-driven development ensures more frequent and regular delivery of working software (Patel & Ramachandran, 2009a), aligning with principles one, three through seven and nine.

As highlighted previously the underlying philosophy of an agile methodology is the continuous, regular delivery of working software to the customer (Fowler & Highsmith, 2001), which is confirmed by critical success factor studies (Chow & Cao, 2008). The AMM at this level of maturity requires *"frequent releases which will create a feedback loop"* (Patel & Ramachandran, 2009a, p. 10), which generally takes the form of a demonstration to the customer of the functionality developed during an iteration. These demonstrations are vital for the customer to provide necessary feedback to the development team and allow surfacing of any incorrect assumptions made during development which assist in improving future iterations (Hoda, Noble, & Stuart, 2011) and relies heavily on the collaborative relation built in the previous maturity level.

This collaboration extends to the development practices employed, with the AMM expecting pair programming, code peer reviews and collective code ownership (Patel & Ramachandran, 2009a). Interestingly these aspects are not explicitly listed in critical success factor research by Chow and Cao (2008) nor explicitly mentioned in the agile manifesto (Chow & Cao, 2008). Unsurprisingly then it remains a contentious issue both in practice and academic research, with findings ranging from showing improvement code quality and increased business knowledge to it having limited success, working only for new and complex problems when the proper mix of skills, personality and expertise are involved (Bipp, Lepper, & Schmedding, 2008; Hannay, Dybå, Arisholm, & Sjøberg, 2009; Lui & Chan, 2006). Lui and Chan (2006), specifically highlight the limitation of these techniques when either experienced developers are paired or the problem domain is well understood.

A further practice assisting the quality of software being delivered is the use of proper agile practices and techniques (Chow & Cao, 2008). Practices such as Test-Driven Development (TDD) in which unit tests are coded before any functionality is developed have been shown to improve the software quality (Crispin, 2006; Sanchez, Williams, & Maximilien, 2007). Building on the previous maturity level, the unit tests are derived from the user story.

Level Four - Improved

The foundation being established in prior maturity levels, level four of the AMM focuses on non-technical aspects such as project, team and people management. It is characterised by a shift toward project management and tracking based on successful delivery (principle seven). Teams are allowed to organise their own development efforts (principle five and eleven), working hours are limited to ensure a sustainable pace (principle eight) and opportunities for improvement are constantly identified (principle twelve) (Patel & Ramachandran, 2009a).

Listed as a critical success factor, proper agile project management techniques (Chow & Cao, 2008) is related to the proper planning, work allocation and progress tracking. Properly prioritised work lists, known as product backlogs, is the responsibility of customer to ensure of the team works on relatively important work items and comprises a list of user stories for future development. A properly prioritised backlog relates to the success of a project (Stettina & Hörz, 2015). In conjunction with the developer estimates for each of the user stories in the previous maturity level, a project plan is compiled.

Having a complete prioritised list of the functionality required allows the selection (pull) of the next piece of functionality to be built by the development team. This pull mechanism of work allocation, based on discussion and collaboration with the customer (Stettina & Hörz, 2015) has been shown to a success factor for agile projects ensuring the most important business functionality is delivered first (Chow & Cao, 2008). This form of work allocation is characteristic of a self-organising team leading to an environment where the development team is trusted to get the work done (Stettina & Hörz, 2015) since the team, inclusive of the customer, have sufficient knowledge and skills to make the correct decisions (Mandarino, 2012). This form of work allocation is a *"classic craftsman environment"* (Boehm & Turner, 2003, p. 7).

Progress tracking for the AMM is performed relative to stories completed (Patel & Ramachandran, 2009a). Ambler (2011) cautions against the use of traditional (iron-triangle) metrics to assess an agile progress team, in favour of the agile iron-triangle (Highsmith, 2004). The AMM suggests the use of agile measures such as burn-down charts to measure overall progress (Papadopoulos, 2015) with story completion rates being used to measure continuous value delivery and risk reduction (Verheyen, 2014).

Principle eight of the agile manifesto introduces the concept of being able to sustain a *"constant pace indefinitely"* (Fowler & Highsmith, 2001, p. 31) which is confirmed by critical success factor research showing adhering to a regular work schedule to be a vital component for success. To achieve a sustainable pace, one of the practices of extreme programming calls for a limit of forty hours (Sauter, 2006).

Level Five - Sustained

The previous maturity levels create an environment in which the agile practice and implementation is properly established. Level five of the AMM switches the focus to project performance management with the team focussing on eliminating the root cause of defects and ensuring quality delivery which meets customer satisfaction (principle one) (Patel & Ramachandran, 2009a), thus maintaining an increased delivery velocity.

In the context of the AMM, project performance management relates to similar focus areas introduced at the lower maturity levels but the expectation at this level is increased. An example of this is, whilst at the lower levels customer presence is required for planning and daily feedback level five requires the customer to be present with and available to the team daily for a minimum of two hours (Patel & Ramachandran, 2009a). Research has shown in mature agile environments the availability and commitment of customers can be more influential for success than satisfaction and collaboration (Misra, Kumar, & Kumar, 2009). Furthermore, story cards produced at this maturity level explicitly insists on the inclusion of acceptance criteria provided by the customer to determine if functionality has been successfully delivered (Patel & Ramachandran, 2009b). Amongst other disciplines the AMM requires, all code to have unit tests, coded prior to the development of the functional code and the release to production only taking place once all unit tests have passed (Patel & Ramachandran, 2009a).

The discipline of testing and quality assurance extends into defect prevention where bugs found in released code are addressed in favour of future functionality (by agreement with customer) (Patel & Ramachandran, 2009b). Root cause analysis is applied to the defect, the cause is addressed, tests are developed to identify the defect and ensure non-recurrence, an approach which has shown to increase both quality and the overall sustainable velocity of delivery (Nagappan, Maximilien, Bhat, & Williams, 2008).

In conclusion, evident from the levels of the AMM is the increased focus on value delivery to the customer. Also apparent are the focus on people through the elimination of overtime and the focus on the trust aspects, key to agile project success (McHugh, Conboy, & Lang, 2012) by allowing the team to take responsibility and rely on self-organisation. With the alignment of the focus areas in the AMM with the agile principles it is expected, achieving higher levels of maturity would contribute to improved perceived project success.

2.4 Summary of Literature Review

Examining the agile manifesto, the underlying philosophy and intent of agile methods is the consistent delivery of value-adding software to a customer, encouraging requirement changes during the project (Fowler & Highsmith, 2001). Though fundamental to agile methods, the agile manifesto does not prescribe a specific methodology which has led to a number of agile methods being used in practice (VersionOne, 2016). Therefore, research intended to study agile methods holistically cannot be limited to a single methodology.

The concept of a team in an agile environment is different to a traditional development team, typically consisting of all the roles required to deliver the software project (Papadopoulos, 2015). Ambler (2012b) highlights the difference between roles and job titles in agile teams which aligns to the concept of the whole team in which *"people with all the skills and perspectives necessary for the project to succeed"* (Beck & Andres, 2004, p. 38) are included.

Given the permissible change in requirements, traditional project success metrics based on scope, cost and schedule are not applicable and literature shows these iron-triangle elements being used only as a constraint measure in agile projects (Leppänen, 2013). Project success is based on the satisfaction perceptions of the stakeholder and thus the construct for this research is perceived project success, consisting of the project teams' perceived satisfaction with deliverables, perception of clients' satisfaction with deliverables and perception of end users' satisfaction with deliverables (Serrador & Pinto, 2015).

In an effort to implement constant improvement, the most commonly adopted model is CMMI which has been shown to be incompatible with proper agility, specifically at higher levels of maturity (Fritzsche & Keil, 2007). The higher levels of CMMI maturity focus on organisational perspectives instead of the project focus of agile methodologies (Laanti, 2014).

A systematic review of current research involving agile maturity shows a distinct focus on merging CMMI and agile methods. Though no research is found relating agile maturity to project success, similar studies use a conceptual model relating CMMI maturity as the independent variable to project success as the dependent variable.

A further focus area for agile maturity is the development of an agile principle-based maturity model. A review of existing proposed maturity models for agile environment, coupled with the intent of this research shows the appropriate model for this research is the AMM as proposed by Patel and Ramachandran (2009a). The AMM proposes a five-level agile principle-based maturity model, with specific agile process areas at each level of maturity. With each level and process area implementing at least one of the agile principles the intent of the AMM is to provide an agile maturity path which should coincide with increased perceived project success.

In general, the intent of a maturity model is the continual improvement of a desired outcome and the underlying philosophy of agile and thus any agile methodology is the delivery of quality software of value to the customer. The concepts of value and quality are aligned to the perceived success of the project as described in section 2.2. It can thus be logically concluded that an increase in agile maturity should relate to an increase in the perceived success of a project.

Though a number of agile maturity models have been proposed, a review of the current literature highlights the lack of research relating improved maturity in an agile maturity model to project success. While studies exist which relate the maturity levels of CMMI to improved project success, Gren, Torkar and Feldt (2015) conclude it would be useful to perform such a study in the context of an agile maturity model. This research will endeavour to address this gap by assessing the relation between the maturity levels in the AMM by Patel and Ramachandran (2009a) and the perceived project success.

3 CONCEPTUAL MODEL, CONSTRUCTS AND HYPOTHESES

The following section introduces the conceptual model selected to address the research gap identified in the literature. Each of the independent constructs are introduced, the associated hypothesis and associated questions, relative to existing literature. The research methodology is also explained in detail, relative to the objective and intent of this research.

3.1 Conceptual Model

The research questions presented in this research combine concepts from agile principles and practices, maturity models and perceived projects with the objective of evaluating the relation between these concepts. Hall (2010) refers to a conceptual model as the combining different concepts with Miles and Huberman (1994) making specific reference to the presumed relations and interactions between the concepts being researched.

Jiang, Klein, Hwang, Huang and Hung (2004) used the conceptual shown in Figure 8 to study the relation between the maturity levels of CMMI as the independent variable and project success as the dependent variable, in a waterfall environment. Using the same conceptual model Rönkkö, Peltonen and Frühwirth (2011) studied the effect of software process maturity and agile methods, as the independent variable on the success of software development efforts, as the dependent variable. As shown in Figure 8, these studies investigate questions similar to those posed in this research using similar constructs in the conceptual model, being the maturity model as the independent variable and project success as the dependent variable. Applying the conceptual model in Figure 8 to the research question results in the high level conceptual model shown in Figure 9. The conceptual model for this research relates the Agile Maturity Model (AMM) as the independent variable to the dependent variable of perceived project success. The following section expounds on the detail of the constructs of the model and introduces the hypotheses for this research.

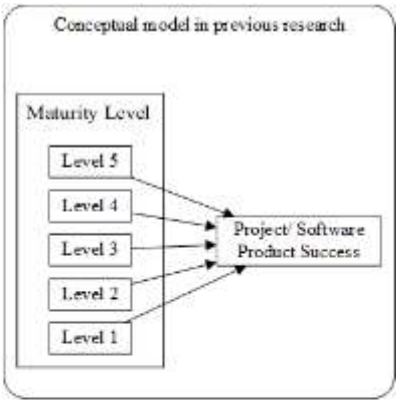

Figure 8: Conceptual model used in previous in research with similar research questions and objectives, (Jiang, Klein, Hwang, Huang, & Hung, 2004; Rönkkö, Peltonen, & Frühwirth, 2011)

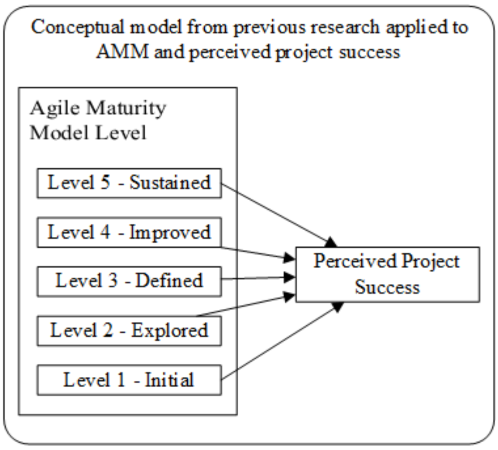

Figure 9: Applying AMM and perceived project success to conceptual models used in previous research

3.2 Constructs and Hypotheses

The high level conceptual model depicted in Figure 9 shows the levels of the agile maturity model as the independent construct and the perceived project success as the dependent construct. This section further explains these constructs in relation to the literature and introduces the hypotheses for this research. The detailed conceptual model is depicted in Figure 10, showing the hypothesised relation between the different focus areas of the AMM and perceived project success.

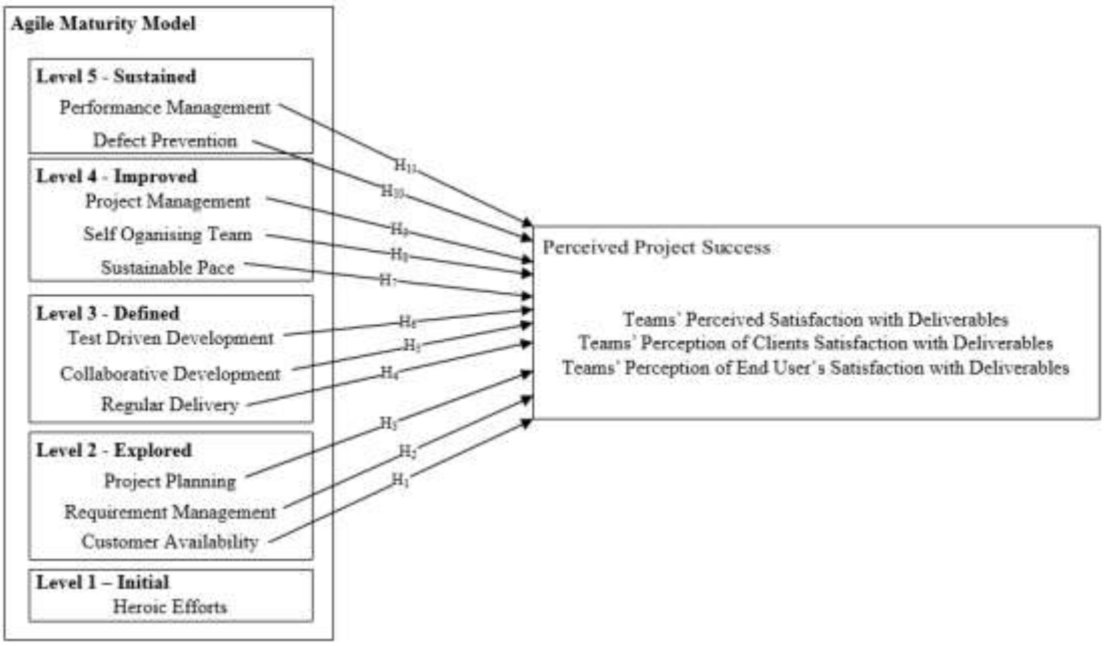

Figure 10: Conceptual model for evaluating the association of focus areas of the AMM (Patel & Ramachandran, 2009a) and perceived project success (Serrador & Pinto, 2015)

3.2.1 Perceived Project Success

Though projects continue to be reported relative to the iron-triangle of scope, cost and schedule, studies have shown project success to be a subjective and perspective-based measure, depending on the stakeholder (McLeod, Doolin, & MacDonell, 2012). Specifically, within an agile environment, Highsmith (2004) proposes an agile iron-triangle which includes value and quality as two of the vertices. Both of these have been shown to be subjective measures (Jugdev & Müller, 2005). In studying overall project success, Serrador and Pinto (2015) conclude that the perceived *"Stakeholder Success"* (Serrador & Pinto, 2015, p. 1043) more accurately predicts project success, which is consistent with findings by Leppänen (2013) who concludes projects using agile methods should use stakeholder satisfaction as the success measure. The dependent variable for this research will thus adopt the definition of perceived project success as being stakeholder satisfaction as defined by Serrador and Pinto (2015). Given that agile teams (inclusive of business representation) are more aware of the customers satisfaction, as substantiated in sections 2.1.3 (Agile Teams Including Business Representative) and 2.1.4 (Agile Teams and Business Feedback) the perceived project success will measured from the perspective of the agile team.

3.2.2 Agile Maturity Model (AMM)

The AMM as proposed by Patel and Ramachandran (2009a) is the independent variable in the high level conceptual model presented in Figure 9. More specifically, to answer the sub question posed in this research the conceptual model needs to be decomposed to the focus areas at each maturity level. The following subsection details these different focus areas and introduces the related hypotheses for the detailed conceptual model for this research, shown in Figure 10.

3.2.2.1 AMM Levels, Focus Areas and Hypotheses

3.2.2.1.1 Level One - Initial

The initial level of the AMM is characterised by being dependent on heroic efforts with no specifically defined process in place (Tarnowski, 2014). Outcomes are not repeatable and there is no alignment to agile principles and is thus excluded from this research.

3.2.2.1.2 Level Two - Explored

Level two of the AMM activities covers the initial set of focus areas which organisations implement to establish agile practices (Patel & Ramachandran, 2009a). This level of maturity focuses on three process areas, being customer availability, requirements management in the form of story cards for the current iteration and project (iteration) planning based on developer estimates. This level shows strong alignment with agile principles two, four, six and ten.

A number of studies have confirmed the presence and availability of an onsite and knowledgeable customer is both a critical success factor in agile environments (Chow & Cao, 2008; Tanner & von Willingh, 2014), is related to improved agile project success (Hoda, Noble, & Stuart, 2011) and aligned to principle four of the agile manifesto. Maturity level two of the AMM requires a knowledgeable customer to be available to the development team on a daily basis and critically be present at the start of an iteration to explain requirements and provide any clarification required (Patel & Ramachandran, 2009a). With research indicating the relation between customer availability and project success as well the alignment with agile principles, the associated hypothesis is:

H_1: Customer availability in an agile team environment is positively associated with the teams' perceived project success.

The corresponding null hypotheses being:

$H_{0\text{-}1}$: Customer availability in an agile team environment is not positively associated with the teams' perceived project success.

The requirements are presented and managed in the form of a story card which provides sufficient detail to derive the acceptance criteria for the functionality and to decompose into the detailed tasks required to deliver the requirement (Patel & Ramachandran, 2009a). Aligning with principle two of the agile manifesto, the story card serves as a guideline to the implementation effort and is allowed to change during the iteration (Sverrisdottir, Ingason, & Jonasson, 2014). The resulting hypothesis is thus:

H_2: Requirement management implemented through the use of story cards which are allowed to change is positively associated with the teams' perceived project success.

The corresponding null hypotheses being:

$H_{0\text{-}2}$: Requirement management implemented through the use of story cards which are allowed to change is not positively associated with the teams' perceived project success.

This granular decomposition of the work required allows developers to provide estimates on the effort required for completion, using methods such as planning poker, similar to wide-band Delphi estimation (Gandomani, Wei, & Binhamid, 2014), relying on the collective knowledge of the team in an estimate-discuss-estimate cycle. The collective approach to estimation has been shown to be more accurate and reliable (Surowiecki, 2004) and tends to be less optimistic estimates (Mahnič & Hovelja, 2012). The combination of the granular decomposition of tasks and related estimates are used to compile the development plan for the iteration (Patel & Ramachandran, 2009a).

H_3: Project planning activities based on estimates by the implementation team is positively associated with the teams' perceived project success.

With the following null hypothesis:

H_{0-3}: Project planning activities based on estimates by the implementation team is not positively associated with the teams' perceived project success.

3.2.2.1.3 Level Three – Defined

Having established the agile practices in level two, level three shifts the focus to better defining the specific agile implementation (Patel & Ramachandran, 2009a) focussing on the use of technical and technological aspects of the implementation. This level is characterised by increased customer relationship management through increased customer presence and customer satisfaction, through constant feedback aligning with principles four and six. Using more collaborative development practices such pair programming and test-driven development ensures more frequent and regular delivery of working software (Patel & Ramachandran, 2009a), aligning with principles one, three through seven and nine.

Principle one and three of the agile manifesto stresses the focus on continuous and regular delivery of working software to the customer (Fowler & Highsmith, 2001), which is confirmed by critical success factor studies (Chow & Cao, 2008). *"Frequent releases which will create a feedback loop"* (Patel & Ramachandran, 2009a, p. 10) are required at this level of maturity which are vital opportunities for the customer to provide necessary feedback to the development team, assisting in improving future iterations (Hoda, Noble, & Stuart, 2011). Being fundamental to the agile manifesto, regular and frequent delivery is expected to relate to an increase in the perceived project success and the associated hypothesis is:

H_4: Regular delivery of software to the customer is positively associated with the teams' perceived project success

With the associated null hypothesis:

$H_{0\text{-}4}$: Regular delivery of software to the customer is not positively associated with the teams' perceived project success

Patel and Ramachandran (2009a) specifically stipulate collaborative development practices such as pair programming, code peer reviews and collective code ownership being characteristic of this maturity level. The related hypothesis for this practice in the AMM is:

H_5: Using collaborative development techniques such as pair programming, peer reviews and collective code ownership is positively associated with the teams' perceived project success.

The associated null hypothesis is:

$H_{0\text{-}5}$: Using collaborative development techniques such as pair programming, peer reviews and collective code ownership is not positively associated with the teams' perceived project success.

Development practices such as TDD in which unit tests are coded before any functionality is developed have been shown to improve the software quality (Crispin, 2006; Sanchez, Williams, & Maximilien, 2007). The tests derive the specifications from the user story cards written by the customer explaining the success criteria. These unit tests enable the regular and frequent delivery being sought at this maturity level (Patel & Ramachandran, 2009a). The corresponding hypothesis is:

H_6: Using test-driven development practices is positively associated with the teams' perceived project success.

With the following null hypothesis:

$H_{0\text{-}6}$: Using test-driven development practices is not positively associated with the teams' perceived project success.

3.2.2.1.4 Level Four - Improved

Having established the foundation for agile methods in the prior maturity levels, the focus of level four shifts to non-technical aspects such as project, team and people management. It is characterised by a shift toward project management and tracking based on successful delivery (principle seven). Teams are allowed to organise their own development efforts (principle five and eleven), working hours are limited to ensure a sustainable pace (principle eight) and opportunities for improvement are constantly identified (principle twelve) (Patel & Ramachandran, 2009a).

Principle eight of the agile manifesto introduces the concept of being able to sustain a *"constant pace indefinitely"* (Fowler & Highsmith, 2001, p. 31). To achieve a sustainable pace, one of the practices of extreme programming calls for a limit of forty hours (Sauter, 2006) and has been found to be an agile success factor (Tanner & von Willingh, 2014) and is specifically required in the AMM (Patel & Ramachandran, 2009a). The corresponding hypothesis is:

H_7: Implementing sustainable pace practices by limiting working hours to forty hours a week is positively associated with the teams' perceived project success.

The corresponding null hypothesis being:

$H_{0\text{-}7}$: Implementing sustainable pace practices by limiting working hours to forty hours a week is not positively associated with the teams' perceived project success.

Proper agile project management techniques have been found to be critical to the success of agile implementations (Chow & Cao, 2008). Similarly, to traditional project management, agile project management is concerned with proper prioritisation, planning and progress tracking. The prioritisation of the functionality comprising a list of user stories known as the product backlog, is the responsibility of the customer and has been found to relate to the success of a project (Stettina & Hörz, 2015). The prioritised backlog along with the more accurate estimates are used as input into the project plan. Ideally in agile environments, work is selected (pulled) by the development team in collaboration with the customer, from the prioritised backlog (Stettina & Hörz, 2015). This form of work allocation is characteristic of a self-organising team leading to an environment where the development team is trusted to get the work done (Stettina & Hörz, 2015) since the team, inclusive of the customer, have sufficient knowledge and skills to make the correct decisions (Mandarino, 2012). This form of work allocation is a *"classic craftsman environment"* (Boehm & Turner, 2003, p. 7) and has been found to be a critical success factor for agile projects ensuring the most important business functionality is delivered first (Chow & Cao, 2008). Furthermore, the team accepts full responsibility for the delivery of the work selected (Patel & Ramachandran, 2009a).

Aligning to principle seven of the agile manifesto, progress tracking in the AMM is measured relative to stories successfully completed (Patel & Ramachandran, 2009a). The AMM encourages the use of metrics which relate to the continuous (business) value delivery and (business or technical) risk reduction (Verheyen, 2014). With the alignment of the project management activities of the AMM with agile principles, critical success factors, focus on the self-organising team and on value delivery, which is a key component of perceived project success, an associated increase in perceived project success would be expected. The related hypotheses are:

H_8: Self-organising teams which are allowed to select the work items and organise themselves to deliver the functionality is positively associated with the teams' perceived project success.

H_9: Agile project management activities using customer prioritised backlogs and tracking mechanisms based on value delivery is positively associated with the teams' perceived project success.

With the following null hypotheses:

$H_{0\text{-}8}$: Self-organising teams which are allowed to select the work items and organise themselves to deliver the functionality is not positively associated with the teams' perceived project success.

$H_{0\text{-}9}$: Agile project management activities using customer prioritised backlogs and tracking mechanisms based on value delivery is not positively associated with the teams' perceived project success.

3.2.2.1.5 Level Five - Sustained

At level five of AMM maturity the team focuses on performance management and the elimination of the root cause of any defects found, ensuring quality delivery which meets customer satisfaction (principle one) (Patel & Ramachandran, 2009a), thus maintaining an increased delivery velocity.

Defect prevention extends the testing and quality assurance discipline. Root cause analysis is applied to the defect, rectified and tests developed to ensure non-recurrence of the defect. Leveraging on the customer availability and involvement, collaboratively the team agrees to focus on the defect elimination in favour of future functionality (Patel & Ramachandran, 2009b). The focus on root cause analysis and defect elimination has been shown to increase quality and overall delivery velocity (Nagappan, Maximilien, Bhat, & Williams, 2008) and is thus expected to increase perceived project success. The hypothesis related to defect prevention is:

H_{10}: Implementing defect prevention and root cause analysis in favour of future functionality is positively associated with the teams' perceived project success.

With the following null hypothesis:

H_{0-10}: Implementing defect prevention and root cause analysis in favour of future functionality is not positively associated with the teams' perceived project success.

In the context of the AMM performance management relates to aspects of customer involvement and satisfaction. In environments which have been practicing agile methods for a number of years, the presence, availability, daily involvement and commitment of customers has been shown to be influential in project success (Misra, Kumar, & Kumar, 2009). Customer involvement extends to the detail of story card to include explicit acceptance criteria which is used for the development of the test cases to indicate completion of the requirements (Patel & Ramachandran, 2009b). As in the previous maturity level the successful completion of requirements becomes one of the measures of success, but at this level augmented with quality measures such as defect tracking in production code. At this level customer satisfaction is compulsory and the AMM equates this with meeting the specified acceptance criteria (Patel & Ramachandran, 2009a). The persistent focus on customer satisfaction as a measure of performance management is consistent with principle seven of the agile manifesto which equates progress with software delivery (Fowler & Highsmith, 2001) and would thus lead to improved perceived project success. The corresponding hypothesis for performance management is:

H_{11}: Project performance management activities focussing on customer involvement and satisfaction is positively associated with the teams' perceived project success.

With the following null hypothesis:

H_{0-11}: Project performance management activities focussing on customer involvement and satisfaction is not positively associated with the teams' perceived project success.

3.3 Research Methodology

3.3.1 Ontology and Philosophy

The question of this research is to determine whether an association exists between the independent variable of the focus areas of the AMM and the dependent variable of perceived project success. The independent variables in the question are concerned with whether the agile practice is being performed regardless of any factors which could influence the practice thereof.

The objective of this research is to determine if and how the presence of these agile practices affect the perceived project success, independent of the context in which it is being practiced, i.e. situational factors are not taken into account.

Saunders, Lewis and Thornhill (2009) refer to objectivism as adopting an approach in which the constructs being studied are viewed in isolation of any external influencing factors. This view is consistent with the objective and question of this research and as such an objectivist ontological stance is appropriate and will be adopted.

Creswell (2009) suggest a positivist approach for research aimed at determining a cause and effect type of relation. Consistent with this view Saunders, Lewis and Thornhill (2009) a positivist view can be adopted when the constructs being observed exist independently of the researcher. These definitions are consistent with the questions posed and objective of this research and thus a positivist epistemology is adopted for this research, which aligns with objectivist ontological stance (Creswell, 2009).

As presented in the section 3.1 a conceptual model, hypothesising relations between the constructs is being evaluated in this research. The development of hypotheses and subsequent testing thereof is consistent with a deductive approach which will be used to prove or disprove the previously stated hypotheses (Saunders, Lewis, & Thornhill, 2009).

3.3.2 Approach to Theory

Gregor (2006) provides a taxonomy of theory types found in information systems research as shown in Table 3. The taxonomy provided relates to the goals of the theory and provides distinguishing attributes of each theory classification. Furthermore, in presenting the different theory types, Gregor (2006) stresses a theory type is not constrained by the ontological and philosophical approaches being adopted in the research.

Table 3: A taxonomy of theory types (Gregor, 2006)

Theory Type	Goal and Distinguishing Attributes
Analysis	Says what is. The theory does not extend beyond analysis and description. No causal relationships among phenomena are specified and no predictions are made.
Explanation	Says what is, how, why, when and where. The theory provides explanations but does not aim to predict with precision. There are no testable propositions.

Theory Type	Goal and Distinguishing Attributes
Prediction	Says what is and what will be. The theory provides predictions and has testable propositions but does not have well-developed justifactory causal explanations.
Explanation and Prediction (EP)	Says what is, how, why, when, where and will be. Provides predictions and has both testable propositions and causal explanations.
Design and Action	Say how to do something. The theory gives explicit predictions (e.g. methods, techniques, principles of form and function) for constructing an artefact.

The stated objective of this research is the analysis and explanation of the association between constructs by using the conceptual model and testing the hypotheses detailed in section 3.2 (Constructs and Hypotheses). Thus, the goal of this research relates to theory type IV of the taxonomy and will be both explanatory and predictive. A further point highlighted by Gregor (2006) as a potential limitation of this theory type, is although the terms 'prediction' and 'causal' are used in the taxonomy and attributes, these can be interpreted as associations and correlations and *"does not necessarily imply a causal relationship"* (Gregor, 2006, p. 626). This is consistent with the conceptual model and hypotheses posited in this research.

3.3.3 Research Method

Creswell (2009) lists attributes of a positivist epistemology as being the intent to verify a concept or theory by means of *"empirical observation and measurement"* (Creswell, 2009, p. 6). Likened to scientific research, a positivist approach first develops the underlying theory or conceptual model for the research problem, generates related hypotheses, stating the relations and associations, and then tests these using appropriate measurement and observation (Saunders, Lewis, & Thornhill, 2009). This process equates to a deductive approach to research and is appropriate when the objective of the research is the assessment of a proposed (hypothesised) relationship between constructs (Saunders, Lewis, & Thornhill, 2009). Consistent with the objective of this research, a deductive approach will be employed.

A key constraint of a deductive approach to ensure rigour is *"that the researcher should be independent of what is being observed."* (Saunders, Lewis, & Thornhill, 2009, p. 125), consistent with objectivist ontology being adopted for this research. To maintain rigour and remove the researcher from the agile environments being researched a quantitative approach will be used. Furthermore, Creswell (2009) suggests a quantitative approach is best suited when the objective of the research is the assessment of the association between constructs.

3.3.4 Inquiry Strategy

The two types of inquiry strategies associated with a positivist quantitative approach are experimental and survey strategies, with the former suited for pure scientific research (Creswell, 2009). Survey strategies, which are used to unearth relations between constructs, can take the form of either a questionnaire or structured observation, in which standardised questions are posed to each participant (Saunders, Lewis, & Thornhill, 2009). For this research however, to achieve the objective and maintain independence between the researcher and the participants, a questionnaire was employed as the inquiry strategy (see section 3.3.6 for details).

3.3.5 Distribution

Consistent with maintaining the independence between the researcher and participants, the survey was distributed using the online survey tool Qualtrics. Online surveys have been shown to be more cost effective, providing both ease of distribution and the ability to reach a large number of potential respondents (van Selm & Jankowski, 2006). However, Evans and Mathur (2005) list respondent bias and ethical concerns such invasion of privacy, associated with online surveys, which are addressed in sections 3.3.9 and 3.3.11 respectively.

3.3.6 Research Instrument

This section details the research instrument to be utilised for this research. The detailed layout of the questionnaire, including the questions related to each construct and the associated scale is shown in section 8.

3.3.6.1 Structure

The research instrument contained three sub-sections namely information and consent, demographic information and survey information. The information and consent provided some background to the research being conducted and include an opportunity for the participant to indicate consent to participate in the research. The demographic information prompted for information to categorise the respondent without identifying the individual or the organisation, consistent with the objectivist ontology.

3.3.6.2 Survey Section

The survey section focused on the conceptual model being researched. The questions used in the research instrument was a combination of the AMM questions used by Patel and Ramachandran (2009a) for assessing the presence of the specific focus area characteristics for a given level of agile maturity and by Serrador and Pinto (2015) in determining stakeholder success factors for the perceived project success and will take the form of a five point Likert scale. A Likert scale has been shown to be useful when measuring opinions and attitudes in social sciences (Croasmun & Ostrom, 2011). The five-point scale is used specifically since Cummins and Gullone (2000) advise that a broader selection of options allow respondents the opportunity to provide a useful response yet a balance needs to be attained in scale length to reduce to the survey response time (Cummins & Gullone, 2000). Furthermore, a scale with an odd number of options is particularly used to allow a neutral response option. The absence of a neutral response option could influence the results obtained since respondents will be forced to adopt a stance (Croasmun & Ostrom, 2011). The forced response will tend to more socially acceptable answers thus skewing the results (Garland, 1991).

The wording of the survey questions is in the form of a statement to which the respondent can indicate agreement on a Likert scale selecting from strongly disagree, disagree, neutral, agree or strongly agree. Podsakoff, MacKenzie, Lee and Podsakoff (2003) highlight the importance of phrasing statements negatively to act as *"cognitive 'speed bumps'"* to avoid *"automatic, cognitive processing"* (Podsakoff, MacKenzie, Lee, & Podsakoff, 2003, p. 884). Thus, where possible each construct will include a negatively worded statement.

3.3.6.3 Question Development

The following section particularises the constructs of the conceptual model and formulates the questions. As previously presented, the constructs representing agile maturity levels are adopted from the AMM based on the model by Patel and Ramachandran (2009a), which includes qualitative assessment criteria for each process area. Since this research is quantitative, questions developed will adapt the assessment criteria from the AMM to be more suitable to a quantitative study and a Likert scale.

3.3.6.3.1 Customer Availability

The AMM requires the customer to be present at the onset of an iteration to explain the requirements to the development team (Patel & Ramachandran, 2009a). Thus, to measure this variable the questions posed were; "The customer is present in at the beginning of a development cycle to explain the business requirements", "The customer is available daily to answer questions" and "The customer is NOT knowledgeable in the business domain of the requirements being developed" (Patel & Ramachandran, 2009a).

3.3.6.3.2 Requirement Management

The requirement management construct in the AMM focuses on requirements being presented using a story card containing, the acceptance criteria for the functionality being developed and sufficient detail to decompose into the detailed tasks required to deliver the functionality. The questions which were used are; "Requirements are presented using stories (story cards)", "The user requirements contain sufficient detail to know what to deliver to satisfy the customer", "The requirements are presented so as to allow detailed tasks to be created" and "Changes are NOT allowed to the user requirements" (Patel & Ramachandran, 2009a).

3.3.6.3.3 Project Planning

The project planning construct in the AMM relies on the use of estimates done by the implementation team, using techniques such as planning poker based on the information presented by the customer at the onset of the iteration. These estimates are used as input into the project plan for the iteration and work undertaken takes into account the available resources during the iteration (Patel & Ramachandran, 2009a). To measure this construct the questions; "The implementation team estimates the work required for the functionality to be developed in the iteration", "Estimation techniques such as planning poker are used", "Estimation is done at the start of the iteration cycle", "The customer is present during estimation", "The estimates provided are used as input in planning the work for the iteration" and "Availability of team members for the iteration are NOT taken into account when doing the iteration planning" (Patel & Ramachandran, 2009a).

3.3.6.3.4 Regular Delivery

The AMM requires not only regular frequent delivery but also insists on it being used to "*create a feedback loop"* (Patel & Ramachandran, 2009a, p. 10) with the customer. This construct was measured using the statements "Functionality developed is demonstrated to the customer at regularly intervals", "When functionality is demonstrated the customer provides feedback" and "Feedback on functionality previously demonstrated is used in future iterations" (Patel & Ramachandran, 2009a).

3.3.6.3.5 Collaborative Development

The AMM stipulates that *"All code is pair programmed"*, the development team *"Perform Peer-reviews"* and *"Use collective code ownership"* (Patel & Ramachandran, 2009a, pp. 10, 11). AgileAlliance (2015) defines collective code ownership as *"the explicit convention that 'every' team member is not only allowed, but in fact has a positive duty, to make changes to 'any' code file as necessary: either to complete a development task, to repair a defect, or even to improve the code's overall structure"*. The construct was evaluated using the statements "Implementation is done using pair programming", "Coding peer reviews are conducted" (Patel & Ramachandran, 2009a) and "Developers are allowed to alter any part of the source code or system required to complete a development task" (AgileAlliance, 2015).

3.3.6.3.6 Test-Driven Development

The AMM requires test cases, derived from the user story cards, to be developed prior to the functionality being developed, implying all production code has corresponding, successfully executed tests (Patel & Ramachandran, 2009a). This construct was measured using the "Test cases are created before the corresponding code is developed", "Test cases are derived from user requirements", "All tests cases must pass before promoting code to production" and "All newly developed code must have accompanying test code" (Patel & Ramachandran, 2009a).

3.3.6.3.7 Sustainable Pace

The AMM states the sustainable pace requirement simply as *"No overtime (40 hours a week)"* and *"Management team offers sustainable pace"* (Patel & Ramachandran, 2009a, p. 13) which is consistent with the practices of extreme programming (Kumar, Singh, & Dwivedi, 2015) as well as principle eight of the agile manifesto (Fowler & Highsmith, 2001). Thus to measure this variable the questions posed were; "Management limit the number of hours worked weekly to a maximum of 40 hours" (Patel & Ramachandran, 2009a), "On average I do not work more than 40 hours a week" (Patel & Ramachandran, 2009a) and "I never work more than 40 hours a week for two consecutive weeks" (Kumar, Singh, & Dwivedi, 2015).

3.3.6.3.8 Self-Organising Team (H_8)

The self-organising team component of the AMM focuses on activities where the team, in conjunction with the customer is allowed to select the work being undertaken in the iteration, challenge the current development methods being employed and regularly look for potential improvements (Patel & Ramachandran, 2009a). The questions used for this variable were; "The team look at areas for improvement affecting the successful delivery of functionality", "The development team, including the customer, select the work being undertaken for an iteration" and "The development team are allowed to implement the simplest solution to meet the requirement".

3.3.6.3.9 Agile Project Management (H_9)

The AMM states project progress is measured based on value delivery, the related question being "Progress within an iteration is tracked using measures such as burn-down charts or stories/features completed or similar measure" (Patel & Ramachandran, 2009a). Furthermore, the work being undertaken is based on a backlog prioritised by the customer and the team fully commits to all agreed work for the iteration. The corresponding questions being; "The team only undertake work which can be completed in an iteration" and "Only work which is of high priority for the customer is undertaken".

3.3.6.3.10 Defect Prevention (H_{10})

The AMM stresses the importance of eliminating the cause of problems by applying root cause analyse and collaboratively sacrificing future functionality in favour of such fixes and automated test cases are developed to ensure the non-reoccurrence of the problem (Patel & Ramachandran, 2009a). The related questions from the AMM are "If/when bugs are found in production;" "the team (including the customer) allocates time to diagnose and fix the root cause of the problem", "test cases are implemented to avoid the future reoccurrence of the bug", "the scope of the current iteration is NOT sacrificed in favour of resolving the problem" (Patel & Ramachandran, 2009a).

3.3.6.3.11 Performance Management (H_{11})

Performance management in the AMM is aimed at meeting the explicitly stated acceptance criteria the customer has stipulated in the initial story card. Only once these criteria have been fulfilled is the functionality considered to have been delivered. Furthermore, production defects being reported are used a quality metric. The questions related to this variable are "Functionality is only accepted for development if the acceptance criteria are explicitly stated", "Functionality is only considered to be completed once all acceptance criteria have been met" and "We DO NOT keep track of the number of production bugs being reported within a development iteration" (Patel & Ramachandran, 2009a).

3.3.6.3.12 Perceived Project Success

As discussed in section 3.2.1 (Perceived Project Success) the working definition adopted for this construct is as used by Serrador and Pinto (2015). The questions for this variable were; "How do you rate the project team's satisfaction with the quality of the project deliverables?", "How do you rate the customer's satisfaction with the quality of the project deliverables?", "How do you rate the end users' satisfaction with the quality of the project deliverables?", "How do you rate the project team's satisfaction with the value delivered in the project?", "How do you rate the customer’s satisfaction with the value delivered in the project?", "How do you rate the end users' satisfaction with the value delivered in the project?".

3.3.7 Research Time Horizon

Saunders, Lewis and Thornhill (2009) suggest a longitudinal time horizon is to be used when the objective is related to understanding a change in a phenomenon over a time period. Since the objective of this research is to understand the association between the constructs represented in the conceptual model at a point in time, a cross-sectional time horizon was adopted for this research. The questionnaire was available online for a month.

3.3.8 Target Population & Sample

The objective and questions posed in this research is independent of any specific agile methodology. The AMM utilised as the independent construct in the conceptual model is consistent with the non-dependence of a specific agile methodology. Literature indicates agile team members are not bound by job titles, with Ambler (2012b) highlighting the difference between roles and job titles in agile teams. Given the conceptual model for this research the population for the study includes respondents practising any agile methodology and cannot be limited to any specific job title.

With the focus on maturity levels, organisations representing the five levels of maturity need to be included in the survey. With the novelty of the agile maturity research area there is no empirical data indicating the agile maturity level of an organisation. Based on CMMI, Shrum and Phillips (2004) provide guidance suggesting organisations can take up to seven years to obtain the higher level of maturity, which was used as guidance in terms of the population being approached.

3.3.9 Sampling Method

Given no comprehensive database exists which defines the sample frame of agile practitioners a combination non-probabilistic sampling approaches will be used (Saunders, Lewis, & Thornhill, 2009). Purposive sampling was initially utilised, including contacts within the industry, agile communities and practitioners. Purposive sampling is a technique used to select respondents who will best answer the research question being posed (Saunders, Lewis, & Thornhill, 2009) based on their knowledge and expertise in the field being researched (Tongco, 2007) and best match the population defined for the study. Though a non-random technique, it has been shown to *"be just as effective as, and even more efficient than, random sampling"* (Tongco, 2007, p. 155). Thereafter a snowball sampling technique was employed, which is characteristic of research involved in a non-deterministic sample frame (Saunders, Lewis, & Thornhill, 2009).

3.3.9.1 Potential Limitations of Sampling Method

Non-probabilistic sampling suffers from common limitations, most notably that of bias and self-selection, potentially leading to a homogenous sample (Saunders, Lewis, & Thornhill, 2009). However, given the objective is to obtain responses from agile practitioners, the initial use of purposive sampling should mitigate this limitation.

3.3.10 Data Analysis

Once the survey period was concluded the data responses were downloaded from the online survey tool. The initial data preparation and cleansing was conducted using Excel due to familiarity with the application. The data was then imported into a statistical analysis tool for further analysis.

Descriptive statistics will initially be utilised to analyse the data pertaining to the range of respondents relative to the demographic information. This will allow analysis showing the distribution of the respondents and highlight whether the results obtained is in any way biased (Saunders, Lewis, & Thornhill, 2009). Normality distribution tests will be conducted using Shapiro-Wilk and Chi-Squared tests for the nominal and ordinal data respectively.

Item reliability and consistency tests will be conducted to ascertain consistency of responses per variable, obtaining a Cronbach alpha measure. A Cronbach alpha of >= 0.7 shows a high degree of consistency (Mitchell & Jolley, 2012). A high internal consistency measure allows the use of an average of the responses per variable to be used in further analysis. Validity tests, specifically Kaiser-Meyer-Olkin (KMO), will be excluded from this analysis since the existing AMM factors are being adopted for the conceptual model and the aim of the research is not to discover new factors via exploratory factor analysis (Williams, 2010).

Correlation analysis will be used because the objective of the research is the investigation of the association between the two constructs and the data being collected is discrete numeric data (Saunders, Lewis, & Thornhill, 2009). Since the data collected will be ordinal, Spearman correlation (rho) will be used to determine the strength and direction of the correlation. Significance testing will be conducted for each correlation hypothesised, to test the probability of the correlation occurring by chance. For resulting p-value < 0.05 the correlation will be deemed to be statistically significant, allowing the relevant null hypotheses to be rejected (Saunders, Lewis, & Thornhill, 2009).

3.3.11 Ethics

The questions posed in this research do not seek any information of an ethical nature. The most common ethical concerns with online surveys include potential invasion of privacy due to broadcast approaches used, such as widespread email, lack of confidentiality and lack of consent being obtained (Buchanan & Hvizdak, 2009). The ethical nature of the questions will be verified by the appropriate ethics approval committee.

3.3.11.1 Privacy and Consent

The initial invitation to participate in the survey was extended to participants identified for the purposive sampling. The survey commences with an acknowledgement of consent, addressing privacy concerns such as anonymity and confidentiality of responses.

3.3.11.2 Confidentiality and Anonymity

The survey questions do not contain any identifying information, either at an individual or organisational level. However, it contained a section about demographic type information which was used to categorise the responses. Thus, this research does not raise any concerns about confidentiality or anonymity since the information will not be available.

3.4 Summary of Research Design

In summary, the purpose and objective of this research is to provide empirical evidence of whether and how differing levels of an agile maturity model are associated with perceived project success. The specific research questions are “How are the different maturity levels of the Agile Maturity Model (AMM) as proposed by (Patel & Ramachandran, 2009a) associated with perceived project success?”. The related sub question being “How are the specific process areas of the different maturity levels in the AMM associated with perceived project success?”. The research is independent of the context of the agile methods being employed or the projects involved. Table 4 shows the summary of the approach taken for each of the research design areas.

Table 4: Research design summary

Research Design Section	Applied Research Approach
Ontology	Objectivist
Epistemology	Positivist
Approach to Theory	Explanation and Prediction
Research Method	Deductive
Inquiry Strategy	Questionnaire
Distribution	Online
Research Instrument Scales	5-point Likert scale
Questions	Adapted from Patel and Ramachandran (2009a) AMM model and Serrador and Pinto (2015) for Perceived Project Success
Time Horizon	Cross sectional
Survey Period	One month
Sampling Method	Purposive and snowball sampling
Data Analysis	Quantitative, using correlation to determine strength and direction of associations

4 RESULTS

The following section details the data analysis of the results obtained from the online survey for this study. Details are provided on how the data was handled and cleansed to be used for the analysis. Descriptive statistics are then used to determine the demographics of the respondents. Reliability and item consistency analysis is used to determine if the responses are consistent, with normality distribution being employed to determine the distribution of the responses. Finally, each of the hypothesis are tested for the association between the independent and dependent constructs, using correlation analysis.

4.1 Data Handling and Cleansing

The responses to the survey were exported from Qualtrics in a comma delimited file (CSV) to enable the data to be analysed in a variety of different applications. The CSV file was imported into Microsoft Excel (version 2007) to perform the initial data cleansing and analysis due to the familiarity with the application. The total number of respondents to the survey was ninety-six. Two of the respondents opted to not partake in the survey by answering “No” to the initial consent question. Of the remaining ninety-four, twenty-five of the surveys did not have all the answers completed and were deemed to have been abandoned. A response was considered incomplete where a respondent did not provide answers to all 45 questions pertaining to the conceptual model. This resulted in a working response set of sixty-nine (R_n = 69) total completed responses. Though a sample of sixty-nine seems low, this does not invalidate the study, as Stutely (2003) advises as a rule of thumb a minimum number of 30 is required for statistical analysis. Furthermore, *“Statisticians have also shown that a sample size of 30 or more will usually result in a sampling distribution for the mean that is very close to a normal distribution”* (Saunders, Lewis, & Thornhill, 2009, p. 218). As shown in Table 5 this gives an opt-out rate of 2.08%, an abandonment rate of 26.60% and a completion rate of 73.40%. The completed responses were then imported into SPSS for further statistical analysis.

Table 5: Survey response and abandon rates

Item	Count	Rate
Number of Completed Responses (R_n)	69	72%
Number of Opt Outs	2	2%
Number of Abandoned Surveys	25	26%
Total Number of Respondents	**96**	**100%**

4.2 Descriptive Analysis

This section describes the background of the respondents in line with the questions asked in the questionnaire, i.e. industry, agile methodology being used, agile experience of the respondent, length of time the project was using an agile methodology, the job title, and role of the respondent.

For the categorical variables industry, agile methodology, job title and agile role, a non-probabilistic one-sample chi-squared test with a null hypothesis that each category occurs with equal probability at a significance level of $p \leq 0.05$ was performed to determine whether the responses were occurring with equal probability.

Tests for normal distribution of the ordinal variables indicating the number of years the project had been using an agile method, coded as "ProjectAgileYears" and the number of years of personal agile experience of the respondent, coded as "AgileExperience" were conducted. The normality test used the Shapiro-Wilk (SW) test with a null hypothesis that the sample is normally distributed. A $p - value < 0.05$ indicates the variable is not normally distributed. The descriptive statistics also included values for median, standard deviation and skewness to show how the responses are distributed. Since the variables are interval in nature, the median is reported.

4.2.1 Responses by Industry

Table 6 depicts how the respondents are distributed across the different industries. With 47 responses, representing 68.12% of R_n, the majority of the responses, were obtained from the financial services sector. These 47 responses were composed of 10 (14%) from Banking, 30 (43%) from Insurance and 7 from "Other" types of financial services.

Table 6: Response distribution by Industry

Industry	Number of Respondents	Percentage of Total Completed Responses
Academic/Education	1	1%
Entertainment	3	4%
Financial Services - Banking	10	14%
Financial Services - Insurance	30	43%
Financial Services - Other	7	10%

Industry	Number of Respondents	Percentage of Total Completed Responses
Medical/Health Services	3	4%
Retail	10	14%
Transportation	1	1%
Other	4	6%
Total	**69**	**100%**

The chi-squared result $X^2(2) = 0.000,\ p \leq 0.05$, as shown in Table 7, indicates the difference in the responses obtained by industry is statistically significant, with a bias towards the financial services industry.

Table 7: One-sample Chi-Square test for responses by Industry

Null Hypothesis	Test	Sig.	Decision
The category of Industry occurs with equal probability	One-Sample Chi-Square Test	.000	Reject the null hypothesis

However, given the non-probabilistic sampling technique utilised in this research, this outcome is expected. Saunders, Lewis and Thornhill (2009) note the common limitations of non-probabilistic sample being bias and self-selection, potentially leading to a homogenous sample (Saunders, Lewis, & Thornhill, 2009). Furthermore, the response distribution achieved is consistent with annual international surveys where financial services and insurance comprise approximately 20% of responses (VersionOne, 2016).

4.2.2 Response by Agile Methodology

Table 8 depicts the distribution of the agile methodology implemented across the projects of the respondents. Consistent with findings from international surveys (VersionOne, 2016) the findings from this study indicates Scrum to be in use in the majority (75%), 52 responses. Interestingly none of the respondents indicated using XP. However, this could be expected as the current trend in the use of agile methods internationally shows that XP is consistently on the decline. In contrast the use of hybrid methodologies, either using Scrum or a Custom Hybrid is consistently in the top three methodologies being employed (VersionOne, 2015; VersionOne, 2016; VersionOne, 2017).

Table 8: Response distribution by agile methodology implemented

Agile Methodology	Number of Respondents	Percentage of Total Completed Responses
Scrum	52	75%
Kanban	4	6%
Scrumban	2	3%
Scrum/XP Hybrid	2	3%
Custom Hybrid	6	9%
Other	3	4%
Total	**69**	**100%**

The chi-squared result $X^2(2) = 0.000,\ p \leq 0.05$, as shown in Table 9, indicates the difference in the responses obtained by agile methodology is statistically significant, with a bias towards the scrum methodology.

Table 9: One-sample Chi-Square test for responses by Agile Methodology

Null Hypothesis	Test	Sig.	Decision
The category of Agile Methodology occurs with equal probability	One-Sample Chi-Square Test	.000	Reject the null hypothesis

However, given the alignment to international trends and findings, the response to this survey is consistent with other observation and thus properly representative of the agile population.

4.2.3 Responses by Years Agile in Use for Project

Table 10 shows the distribution of the number of years projects have been using agile methodologies. Responses are evenly distributed across the available year ranges, however the majority of projects, 50 responses (72%) indicate using agile methodologies for three years or less.

Table 10: Distribution of responses by number of year's project has been using agile

Number of Years Project Using Agile Methodologies	Number of Respondents	Percentage of Total Completed Responses
Less than 1 year	14	20%

Number of Years Project Using Agile Methodologies	Number of Respondents	Percentage of Total Completed Responses
Between 1 and 2 years	16	23%
Between 2 and 3 years	20	29%
Between 3 and 4 years	7	10%
Between 4 and 5 years	11	16%
Between 5 and 6 years	0	0%
More than 6 years	1	1%
Total	**69**	**100%**

The normal distribution and descriptive statistics results as shown in Table 11, indicate the responses are not normally distributed, with a SW result of 0.905 and $p - value < 0.001$. The skewness value of 0.512 indicates the distribution is moderately (Bulmer, 1979) and slightly positively skewed.

Table 11: Descriptive statistics and normal distribution results for project agile time

Median	Std Deviation	Skewness	Shapiro-Wilk	Significance
3	1.421	0.512	0.905	0.000

4.2.4 Responses by Agile Experience of Respondents

Table 12 shows the distribution of responses as per the number of years of experience of the respondents with the agile methodology. Given the publication of the agile manifesto in 2001 (Fowler & Highsmith, 2001) being over fifteen years ago it is unsurprising the majority of the respondents, 43 responses equating to 62%, report to have more than four years of agile experience, comprised of 16 responses between 4 and 5 years, 4 responses between 5 and 6 years and 23 responses with more than 6 years' experience.

Table 12: Response distribution by agile experience of respondents

Experience with Agile Methodologies	Number of Respondents	Percentage of Total Completed Responses
Less than 1 year	6	9%
Between 1 and 2 years	5	7%
Between 2 and 3 years	12	17%

Between 3 and 4 years	3	4%
Between 4 and 5 years	16	23%
Between 5 and 6 years	4	6%
More than 6 years	23	33%
Total	**69**	**100%**

The normal distribution tests conducted for agile experience, as shown in Table 13 indicate the variable is not normally distribution, with SW measure of 0.869 and a $p-value < 0.001$. The skewness value of -0.410 indicates the distribution is fairly symmetrical (Bulmer, 1979) and slightly negatively skewed.

Table 13: Descriptive statistics and normality statistics for agile experience responses

Median	Std Deviation	Skewness	Shapiro-Wilk	Significance
5	2.045	-0.410	0.869	0.000

The responses obtained are consistent with findings from international studies which show the majority of respondents having more than four years of agile experience (VersionOne, 2017). However, a quarter of the respondents specified having less than three years of experience, which indicates that agile methodologies are experiencing a healthy adoption rate, consistent with international surveys (VersionOne, 2017).

4.2.5 Responses by Job Title and Agile Role of Respondents

Table 14 shows the distribution of respondents by job title. The majority of responses, 29% (20 respondents) were from “developers”. The remaining responses were evenly distributed, with the only other notable exception being that of “solution architect” at 10% (7 respondents).

Table 14: Response distribution of job title responses

Job Title	Number of Respondents	Percentage of Total Completed Responses
Business Analyst	3	4%
CTO	2	3%
Data Analyst	1	1%
Data architect	2	3%
Developer	20	29%

Job Title	**Number of Respondents**	**Percentage of Total Completed Responses**
Development Manager	2	3%
DevOps specialist	1	1%
Director of Technology	1	1%
Head of Process Engineering	1	1%
IT Team Manager	1	1%
Lean Agile Coach	1	1%
Line Manager / Scrum Master	1	1%
PMO Manager	1	1%
Practice Head: Software Quality Engineering	1	1%
Product Owner	2	3%
Programme Manager	1	1%
Project manager	3	4%
QA engineer	1	1%
Quality Assurance & Test Manager	1	1%
Scrum Master	4	6%
Scrum Master/ Project Manager	1	1%
Senior Business Analyst	1	1%
Senior Manager Custom Application Development	1	1%
Senior Staff Engineer	1	1%
Software Practise Engineering - Practise Manager R&D	1	1%
Solution Architect	7	10%
Systems Analyst	2	3%
Tech lead	1	1%

Job Title	Number of Respondents	Percentage of Total Completed Responses
Test Analyst	1	1%
VP of Technology	1	1%
Team leader	1	1%
Agile coach	1	1%
Total	**69**	**100%**

The prevalence of the developer as well as the solution architect is unsurprising as Ambler (2012b) mentions the predominance of both the developers and architects in agile teams, particularly in large corporate environments.

The chi-squared result $X^2(2) = 0.000,\ p \leq 0.05$, as shown in Table 15, indicates the difference in the responses obtained by job title is statistically significant, with a bias towards respondents having a job title of developer.

Table 15: One-sample Chi-Square test for responses by Job Title

Null Hypothesis	Test	Sig.	Decision
The category of Job Title occurs with equal probability	One-Sample Chi-Square Test	.000	Reject the null hypothesis

Table 16 shows the distribution of respondents by the agile role and confirms the difference between roles and job titles in agile teams (Ambler, 2012b), with the developer role being the most common amongst respondents. Notably a number of respondents perform multiple roles. The chi-squared result $X^2(2) = 0.849,\ p \leq 0.05$, as shown in Table 17, indicates the difference in the responses obtained by agile role is not statistically significant and thus no bias towards any specific agile role is present.

Table 16: Distribution of respondents by agile role

Agile Role	Number of Respondents	Percentage of Total Completed Responses
Business representative	10	8%
Scrum Master	16	12%
Line Manager	19	14%

Agile Role	Number of Respondents	Percentage of Total Completed Responses
Developer	27	20%
Business Analyst	8	6%
Tester	10	8%
Designer	11	8%
Architect	21	16%
DevOps	10	8%
Total	**132**	**100%**

Table 17: One-sample Chi-Square test for responses by Agile Role

Null Hypothesis	Test	Sig.	Decision
The category of Job Title occurs with equal probability	One-Sample Chi-Square Test	.849	Retain the null hypothesis

4.3 Item Consistency and Reliability Tests

For each of the constructs the internal consistency of responses was determined using the Cronbach alpha (α) test. A Cronbach $\alpha = 0.7$ is deemed to show the combination of questions posed for a construct is reliable (Nunnally, 1978). Although an α as low as 0.6 is acceptable for exploratory research (Fornell & Larcker, 1981), where a construct was found to have an $\alpha < 0.6$, the test was extended to determine the effect the elimination of a question would have on the reliability measure. Since Cronbach α depicts the lower bound reliability measure (Hair & Hult, 2016), it is suggested further reliability tests be conducted for low α results (Sijtsma, 2009). For these cases, the Spearman correlation (rho) was calculated to determine whether there was a statistically significant correlation between the responses. A statistically significant correlation measure and $\alpha \geq 0.6$ ensures the questions used in the analysis of the construct are aligned to the intent of the construct.

The questions related to a construct were coded as a concatenation of the construct abbreviation, e.g. CA (Customer Availability) and the question number in the survey. As shown in Table 18, the constructs Customer Availability, Requirements Management, Self-Organising Teams, Agile Project Management and Performance Management have a Cronbach $\alpha = 0.7$, Project Planning, Regular Delivery, Test Driven Development and Sustainable Pace have a Cronbach $\alpha = 0.8$ and Perceived Project Success with a Cronbach $\alpha = 0.9$. These constructs all show high internal consistency, however Collaborative Development ($\alpha = 0.5$) and Defect Prevention ($\alpha = 0.4$) show low internal consistency and further analysis is applied as further detailed in the following subsections.

Table 18: Cronbach alpha internal consistency results

Construct	Cronbach Alpha
Customer Availability	0.7
Requirements Management	0.7
Project Planning	0.8
Regular Delivery	0.8
Collaborative Development	0.5
Test Driven Development	0.8
Sustainable Pace	0.8
Self-Organising Team	0.7
Agile Project Management	0.7
Defect Prevention	0.4
Performance Managements	0.7
Perceived Project Management	0.9

4.3.1 Collaborative Development

A Cronbach of $\alpha = 0.5$ was obtained for the Collaborative Development construct and as shown in Table 20, the exclusion of any of the questions does not improve the reliability measure of the construct. Table 19 shows the results of the rho correlation tests for the collaborative development construct. The results show a statistically significant correlation between CD17 and CD18 at a 0.01 (99%) confidence level. However, CD19 shows statistically insignificant correlation with these questions. CD19 is thus excluded from further analysis and the construct is represented by CD17 and CD18.

Table 19: Internal consistency Spearman (rho) correlation for Collaborative Development Construct

Variable	CD17	CD18	CD19
CD17	1.000		

Variable	CD17	CD18	CD19
CD18	0.349**	1.000	
CD19	0.099	0.178	1.000

**** Correlation is significant at the 0.01 level (2-tailed)**

Table 20: Effect on Cronbach alpha measure when deleting questions

Variable	Cronbach's Alpha if Item Deleted
CD17	0.335
CD18	0.225
CD19	0.498

4.3.2 Defect Prevention

A Cronbach of $\alpha = 0.4$ was obtained for the Defect Prevention construct, however as shown in Table 22, the exclusion of any of the questions does not improve the reliability of the construct. Table 21 shows the results of the rho correlation tests for the defect prevention construct. The results show a statistically significant correlation between DP33 and DP34 at a 0.01 (99%) confidence level. However, DP35 shows statistically insignificant correlation with these questions and a possible explanation is the negative phrasing of the question. Roszkowski and Soven (2010) show that negatively phrased questions could adversely affect the response correlation. DP35 is thus excluded from further analysis and the construct was represented by DP33 and DP34.

Table 21: Internal consistency Spearman (rho) correlation for Defect Prevention Construct

	DP33	DP34	DP35
DP33	1.000		
DP34	0.418**	1.000	
DP35	0.113	0.070	1.000

**** Correlation is significant at the 0.01 level (2-tailed)**

Table 22: Effect on Cronbach alpha measure when deleting questions

Variable	Cronbach's Alpha if Item Deleted
DP33	0.115

Variable	Cronbach's Alpha if Item Deleted
DP34	0.246
DP35	0.547

4.4 Construct Normality and Distribution

Each construct was tested for normality using the Shapiro-Wilk test for normality. The construct was also evaluated for distribution by evaluating its descriptive statistics the mean, standard deviation and skewness. The normal distribution of the constructs influenced the statistical method utilised for the hypothesis testing, since certain statistical tests assume a normal distribution of the data. The following sub-sections discuss each construct in reference to Table 23.

Table 23: Descriptive statistics and normality test results

Construct	Mean	Std Deviation	Skewness	Shapiro-Wilk			Distribution
				Statistic	df	Sig	
CA	4.04	0.83	-0.80	0.90	69	0.000	Non-Normal
RM	3.62	0.72	-0.85	0.92	69	0.000	Non-Normal
PP	3.65	0.77	-0.95	0.93	69	0.001	Non-Normal
RD	3.99	0.87	-1.53	0.84	69	0.000	Non-Normal
CD	2.62	0.97	0.36	0.97	69	0.005	Normal
TDD	3.58	0.83	-0.83	0.95	69	0.007	Non-Normal
SP	2.79	1.03	0.35	0.96	69	0.026	Non-Normal
SOT	3.67	0.76	-0.72	0.93	69	0.001	Non-Normal
APM	3.59	0.86	-0.63	0.94	69	0.002	Non-Normal
DP	3.49	0.85	-0.57	0.98	69	0.001	Non-Normal
PerfMng	3.41	0.75	-0.39	0.98	69	0.141	Normal
PPS	3.68	0.83	-0.71	0.90	69	0.000	Non-Normal

4.4.1 Customer Availability (CA)

With a $p - value < 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.83, the average response for the construct is closely distributed around the mean of 4.04. The responses are slightly negatively skewed with a skewness measure of -0.80.

4.4.2 Requirements Management (RM)

With a $p - value < 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.72, the average response for the construct is closely distributed around the mean of 3.62. The responses are slightly negatively skewed with a skewness measure of -0.85.

4.4.3 Project Planning (PP)

With a $p - value = 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.77, the average response for the construct is closely distributed around the mean of 3.65. The responses are slightly negatively skewed with a skewness measure of -0.95.

4.4.4 Regular Delivery (RD)

With a $p - value < 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.87, the average response for the construct is closely distributed around the mean of 3.99. The responses are slightly negatively skewed with a skewness measure of -1.53.

4.4.5 Collaborative Development (CD)

With a $p - value = 0.005$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.97, the average response for the construct is closely distributed around the mean of 2.62. The responses are slightly positively skewed with a skewness measure of 0.36.

4.4.6 Test Driven Development (TDD)

With a $p - value = 0.007$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.83, the average response for the construct is closely distributed around the mean of 3.58. The responses are slightly negatively skewed with a skewness measure of -0.83.

4.4.7 Sustainable Pace (SP)

With a $p - value = 0.026$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 1.03, the average response for the construct is closely distributed around the mean of 2.79. The responses are slightly positively skewed with a skewness measure of 0.35.

4.4.8 Self-Organising Team (SOT)

With a $p - value = 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.76, the average response for the construct is closely distributed around the mean of 3.67. The responses are slightly negatively skewed with a skewness measure of -0.72.

4.4.9 Agile Project Management (APM)

With a $p - value = 0.002$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.86, the average response for the construct is closely distributed around the mean of 3.59. The responses are slightly negatively skewed with a skewness measure of -0.63.

4.4.10 Defect Prevention (DP)

With a $p - value < 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.85, the average response for the construct is closely distributed around the mean of 3.49. The responses are slightly negatively skewed with a skewness measure of -0.57.

4.4.11 Performance Management (PerfMan)

With a $p - value = 0.141$, the SW test is not statistically significant and the construct is thus normally distributed. Having a standard deviation of 0.75, the average response for the construct is closely distributed around the mean of 3.41. The responses are slightly negatively skewed with a skewness measure of -0.39.

4.4.12 Perceived Project Success (PPS)

With a $p - value < 0.001$, the SW test is statistically significant and the construct is thus non-normally distributed. Having a standard deviation of 0.83, the average response for the construct is closely distributed around the mean of 3.68. The responses are slightly negatively skewed with a skewness measure of -0.71.

In summary, the majority of the constructs produced statistically significant results for the Shapiro-Wilk test for normal distribution. Hence the responses for the constructs were regarded as non-normally distributed and non-parametric statistical methods were applied for further analysis (Saunders, Lewis, & Thornhill, 2009).

4.5 Maturity Levels

The four levels of the agile maturity model (AMM) (Patel & Ramachandran, 2009a) under investigation in this research, consist of groupings of various constructs as shown in Figure 11. As previously mentioned, level 1 does not contain specifically defined processes (Tarnowski, 2014) and outcomes are not repeatable, with no alignment to agile principles and is thus excluded from this study. To ascertain whether these groupings of variables relate to the maturity level, consistency and reliability tests were conducted on the different groups of responses as indicated. If a statistically significant correlation and a Cronbach alpha reliability of $\alpha \geq 0.7$ was attained, the variables were combined to represent the respective maturity level.

Figure 11: Variable groupings per maturity level

4.5.1 Level 2 Maturity - Explored

Level two maturity consists of the CA, RM and PP variables. As shown in Table 24, the variables show a statistically significant correlation at the 0.01 (99%) and 0.05 (95%) confidence intervals with a reliability measure of Cronbach $\alpha = 0.7$. The level 2 maturity (L2) is thus represented by the average of the CA, RM and PP variables.

Table 24: Internal consistency Spearman (rho) correlation for Level 2 maturity

	CA	RM	PP
CA	1.000		

RM	0.423**	1.000	
PP	0.546**	0.281*	1.000
**** Correlation is significant at the 0.01 level (2-tailed)**			
*** Correlation is significant at the 0.05 level (2-tailed)**			

4.5.2 Level 3 Maturity - Defined

Level three maturity consists of the CD, RD and TDD variables. As shown in Table 25, the variables show a statistically significant correlation at the 0.01 (99%) confidence interval with a reliability measure of Cronbach $\alpha = 0.7$. The level 3 maturity (L3) is thus represented by the average of the CD, RD and TDD variables.

Table 25: Internal consistency Spearman (rho) correlation for Level 3 maturity

	CD	**RD**	**TDD**
CD	1.000		
RD	0.413**	1.000	
TDD	0.542**	0.441**	1.000
**** Correlation is significant at the 0.01 level (2-tailed)**			

4.5.3 Level 4 Maturity - Improved

Level four maturity consists of the APM, SOT and SP variables. As shown in Table 26, the variables show a statistically significant correlation at the 0.01 (99%) confidence interval with a reliability measure of Cronbach $\alpha = 0.7$. The level 4 maturity (L4) is thus represented by the average of the APM, SOT and SP variables.

Table 26: Internal consistency Spearman (rho) correlation for Level 4 maturity

	SP	**SOT**	**APM**
SP	1.000		
SOT	0.340**	1.000	
APM	0.350**	0.483**	1.000
**** Correlation is significant at the 0.01 level (2-tailed)**			

4.5.4 Level 5 Maturity - Sustained

Level five maturity consists of the PerfMng and DP variables. As shown in Table 27, the variables show a statistically significant correlation at the 0.01 (99%) confidence interval with a reliability measure of Cronbach $\alpha = 0.6$. The level 5 maturity (L5) is thus represented by the average of the PerfMng and DP variables.

Table 27: Internal consistency Spearman (rho) correlation for Level 5 maturity

	DP	PerfMng
DP	1.000	
PerfMng	0.427**	1.000

**** Correlation is significant at the 0.01 level (2-tailed)**

4.6 Hypotheses Tests

Correlation analysis was used because the objective of the research is the investigation of the association between the two constructs and the data collected is discrete numeric data (Saunders, Lewis, & Thornhill, 2009). Since the conceptual model being investigated in this research and the corresponding hypotheses are unidirectional, a one-tailed correlation was employed. As concluded in the normality tests (section 4.4) the underlying constructs are not normally distributed and thus a non-parametric correlation method was used to determine the association (Saunders, Lewis, & Thornhill, 2009). Spearman's non-parametric correlation (r_s) was used to determine the strength and direction of the correlation, since it does not assume normality in the underlying data. Significance testing was conducted for each correlation hypothesised, to test the probability of the correlation occurring by chance. For resulting p-value < 0.05 the correlation was deemed to be statistically significant, allowing the relevant null hypotheses to be rejected (Saunders, Lewis, & Thornhill, 2009).

Table 28 shows the summary of the Spearman rho correlation for each of the independent constructs against the dependent construct of perceived project success. As is evidenced, all the constructs show statistically significant correlation at either the 0.01 (99%) or 0.05 (95%) confidence interval. The strongest correlation of 0.626 is observed for performance management (PerfMng) while the weakest correlation of 0.270 is observed for collaborative development (CD).

Table 28: Summary of Spearman rho Correlation of independent constructs to the dependent construct of Perceived Project Success, ordered by descending strength of correlation

Construct	Correlation
Performance Management	0.626**
Requirement Management	0.559**
Self-Organising Team	0.540**
Test Driven Development	0.496**
Regular Delivery	0.491**
Defect Prevention	0.473**
Project Management	0.473**
Customer Availability	0.401**
Project Planning	0.347**
Sustainable Pace	0.340**
Collaborative Development	0.270*

**** Correlation is significant at the 0.01 level (1-tailed).**

*** Correlation is significant at the 0.05 level (1-tailed).**

Table 29 shows the summary of the Spearman rho correlation for each of the maturity level constructs against the dependent construct of perceived project success. As is evidenced, all the constructs show statistically significant correlation at the 0.01 (99%) confidence interval. The strongest correlation of 0.626 is observed at maturity level five (L5) while the weakest correlation of 0.482 is observed at maturity level three (L3).

Table 29: Summary of Spearman rho Correlation of independent constructs to the dependent construct of Perceived Project Success

Construct	Perceived Project Success (PPS)
Level 2 – Explored (L2)	0.507**
Level 3 – Defined (L3)	0.482**
Level 4 – Improved (L4)	0.575**
Level 5 – Sustained (L5)	0.616**

**** Correlation is significant at the 0.01 level (1-tailed).**

5 FINDINGS AND DISCUSSION

The following section includes the interpretation of the results obtained in the previous chapter, relative to the conceptual model (refer Figure 13) used for this study. Each of the hypothesis are analysed and explanation of the findings are presented relative to existing literature. The chapter concludes by providing answers to the research question and sub-question identified in this research.

5.1 Introduction

A maturity model defines the specific focus areas required to improve the outcome of a desired process as the practice develops. The Agile Maturity Model (AMM) as proposed by Patel and Ramachandran (2009a) depicts a five-level maturity model. In combination with principle seven of the agile manifesto*, "Working software is the primary measure of progress"* (Fowler & Highsmith, 2001, p. 35), this research attempts to determine whether there is an association between maturity in an agile implementation and the success of these projects, as shown in the high level conceptual model shown in Figure 12. The study further investigated how the specific activities in each maturity level is associated with the perceived project success construct, giving rise to the conceptual model and corresponding 11 hypotheses as shown in Figure 13. The conceptual model shows the association between the independent constructs of the maturity levels and process areas of the AMM and the dependent construct of perceived project success, primarily as defined by previous work by Serrador and Pinto (2015).

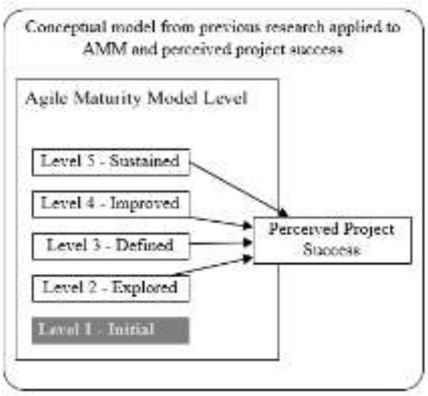

Figure 12: High level conceptual model under research

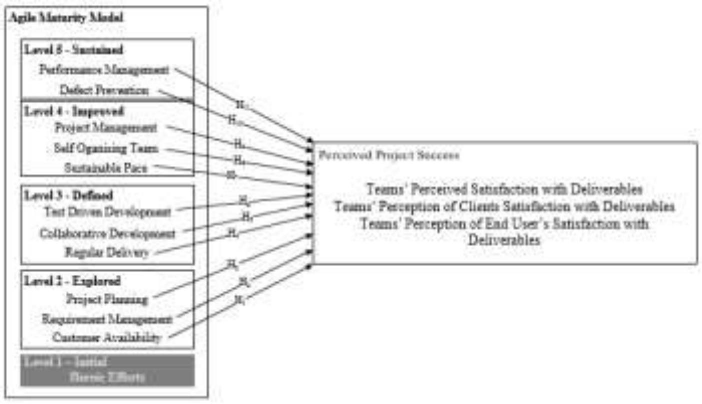

Figure 13: Conceptual model with Hypotheses associating AMM process areas with perceived project success

5.2 Interpretation of Hypotheses Analysis

5.2.1 Customer Availability (H_1)

H_1: Customer availability in an agile team environment is positively associated with the teams' perceived project success.

The Customer Availability construct shows a statistically significant correlation of 0.401 at a confidence level of 0.01 (99%). Thus, the null hypothesis $H_{0\text{-}1}$: Customer availability in an agile team environment is not positively associated with perceived project success, is rejected in favour of H_1. The responses to this survey indicate that customer availability is positively associated with the teams' perceived project success.

This finding is consistent with previous research on customer availability being a critical success factors for agile environments (Chow & Cao, 2008; Tanner & von Willingh, 2014) and further supports principle four of the agile manifesto (Fowler & Highsmith, 2001). Boehm and Turner (2003) emphasise the need for "CRACK" (Collaborative, Representative, Authorised, Committed, and Knowledgeable) business user involvement who is able and empowered to make the proper decisions about requirements for development, in a successful development process. Critically at level two maturity of the AMM, a knowledgeable customer should be present at the start of an iteration to ensure requirements are properly understood, explained and clarified (Patel & Ramachandran, 2009a). Furthermore, findings from this study confirm previous work in which the daily availability and inclusion of the customer in decisions related to the development of the software is positively associated with project success (Abelein & Paech, 2015; Lin & Shao, 2000).

5.2.2 Requirements Management (H_2)

H_2: Requirement management implemented through the use of story cards which are allowed to change is positively associated with the teams' perceived project success.

Responses to this survey show a statistically significant correlation of 0.559 at the 0.01 (99%) confidence level, between the requirements management and perceived project success construct. The null hypothesis $H_{0\text{-}2}$ is rejected in favour of H_2 and the finding from this research is that requirement management is positively associated with the teams' perceived project success.

Consistent with findings by Patel and Ramachandran (2009b) this research indicates that the management of requirements represented in story cards is positively associated with project success. The story cards should be written with sufficient detail and should be allowed to change during the development cycle. Furthermore, the findings from this research is consistent with principle two of the agile manifesto (Fowler & Highsmith, 2001) with Ambler (2014) and Sverrisdottir, Ingason and Jonasson (2014) highlighting the positive influence changing requirements can have on this can have on project success. Baruah (2015) however, cautions against constantly changing requirements and the regularity of the delivery, although concludes agile methods are better suited to handle volatile requirements, which again aligns with the findings from this study.

5.2.3 Project Planning (H_3)

H_3: Project planning activities based on estimates by the implementation team is positively associated with the teams' perceived project success.

The responses for this survey show a statistically significant correlation of 0.347 at a 0.01 (99%) confidence level between project planning and perceived project success. Thus, the null hypothesis H_{0-3} "Project planning activities based on estimates by the implementation team is not positively associated with team's perceived project success" is rejected in favour of H_3.

The results confirm that the use of agile estimation techniques as input into the project plan for the iteration, relying on the collective knowledge of the implementation team allows for more accurate and reliable estimation (Surowiecki, 2004), which in the context of this study, is found to have a positive association with the teams' perceived project success. Using the teams' input into the iteration plan and the presence of the customer during the initial estimation, allows for transparency, clarification and expectation management, resulting in a more realistic and achievable plan (Patel & Ramachandran, 2009b). This confirms previous findings that input from the development team in project planning, based on the granular decomposition of the work required allows for more accurate, reliable and realistic estimates (Turner, 2014) and thus improved project success.

5.2.4 Regular Delivery (H_4)

H_4: Regular delivery of software to the customer is positively associated with the teams' perceived project success

The regular delivery construct shows a statistically significant correlation of 0.491 at a 0.01 (99%) confidence level with the perceived project success construct. Thus, the null hypothesis H_{0-4} "Regular delivery of software to the customer is not positively associated with team's perceived project success" is rejected in favour of H_4.

The positive association is found between regular delivery and perceived project success is confirmation of the underlying philosophy of an agile implementation is embodied in principle one, *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software"* (Fowler & Highsmith, 2001, p. 30) and further embodied in principle seven of the agile manifesto *"Working software is the primary measure of success"* (Fowler & Highsmith, 2001, p. 35). Furthermore, these results align with existing consensus in literature that regular delivery is a critical success factor for agile implementations (Chow & Cao, 2008; França, da Silva, & de Sousa Mariz, 2010) with França, da Silva and de Sousa Mariz (2010) finding it to have the strongest correlation with project success.

5.2.5 Collaborative Development (H_5)

> *H_5: Using collaborative development techniques such as pair programming, peer reviews and collective code ownership is positively associated with the teams' perceived project success*

Though the agile manifesto encourages collaborative practices as per principle six *"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation"* (Fowler & Highsmith, 2001, p. 35), collaborative practices such as pair programming is not found to be a critical success factor in agile implementations in literature (Chow & Cao, 2008). However, the responses obtained in this research show a statistically significant correlation of 0.270 at a confidence level of 0.05 (95%) between collaborative development practices and perceived project success. Thus, the null hypothesis H_{0-5} is rejected in favour of H_5.

However, this construct was found to have the weakest correlation of all the constructs. Though collaborative techniques have been found to result in improved code quality and increased business knowledge (Bipp, Lepper, & Schmedding, 2008), but is said to have limited success, working only for new and complex problems when the proper mix of skills, personality and expertise are involved (Hannay, Dybå, Arisholm, & Sjøberg, 2009; Lui & Chan, 2006). More specifically, collaborative development techniques have been found to be less effective amongst experienced developers (Hannay, Dybå, Arisholm, & Sjøberg, 2009; Lui & Chan, 2006) and since the majority of the respondents for this study have more than six years of agile experience (refer section 4.2.4) the weak correlation found is possibly due to prevalence of more experienced respondents, aligning to existing findings (Hannay, Dybå, Arisholm, & Sjøberg, 2009; Lui & Chan, 2006). Due to the limitations of this study (refer section 6.2) it is not feasible to perform more in-depth analysis to explain the results obtained in further detail however, it could be a topic for future research.

5.2.6 Test Driven Development (H_6)

H_6: Using test-driven development practices is positively associated with the teams' perceived project success

A statistically significant correlation of 0.496 at a confidence level of 0.01 (99%) between test driven development and perceived project success was found. Thus, the null hypothesis $H_{0\text{-}6}$ is rejected in favour of H_6.

As discussed in section 2.2, a key component of perceived project success in an agile environment is quality (Serrador & Pinto, 2015), which is *"the most subject to variation in perception by multiple project stakeholders"* (Prabhakar, 2009, p. 7). Test driven development activities have been shown to improve the quality of the implemented software (Crispin, 2006; Sanchez, Williams, & Maximilien, 2007), whilst ensuring frequent delivery (Patel & Ramachandran, 2009a). Whilst an increase in the quality has been found to improve project success (Serrador & Pinto, 2015), the frequent delivery aligns with principle three of the agile manifesto *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."* (Fowler & Highsmith, 2001, p. 35). The finding of this study is thus aligned with previous literature.

5.2.7 Sustainable Pace (H_7)

H_7: Implementing sustainable pace practices by limiting working hours to forty hours a week is positively associated with the teams' perceived project success

The sustainable pace construct shows a statistically significant correlation of 0.340 at a 0.01 (99%) confidence level. Thus, the null hypothesis $H_{0\text{-}7}$: "Implementing sustainable pace practices by limiting working hours to forty hours a week is not positively associated with team's perceived project success" is rejected in favour of H_7.

Principle eight of the agile manifesto mentions *"constant pace indefinitely"* (Fowler & Highsmith, 2001, p. 31), which is the basis of this construct in the AMM. The practice of limiting the working week to forty hours is one of the principles in extreme programming (Sauter, 2006) and has been shown to be a critical success factor in agile implementation (Chow & Cao, 2008; Tanner & von Willingh, 2014). The finding from this research thus aligns with critical success factor research (Chow & Cao, 2008; Tanner & von Willingh, 2014). With a correlation of 0.340, the sustainable pace construct is found to have the second lowest correlation to perceived project success. A possible explanation for the low correlation is the negatively phrased questions used in the wording of one of the questions for this construct, which has sometimes been found to potentially adversely affect responses (Roszkowski & Soven, 2010).

Though Chow and Cao (2008) stress sustainable pace practice as a critical success factor, França, da Silva, and de Sousa Mariz (2010) found sustainable pace to not correlate with project success. A possible explanation for the misalignment between the findings from this study and that of França, da Silva and de Sousa Mariz (2010), is this study does not suffer from the same generalisation restriction mentioned by França, da Silva and de Sousa Mariz (2010). The generalisation restriction in the study by França, da Silva and de Sousa Mariz (2010) is primarily due to the size and localisation of the survey participants. Though the current study has a similar response size, the sample does not suffer from the same localisation restriction.

5.2.8 Self-Organising Team (H_8)

H_8: Self-organising teams which are allowed to select the work items and organise themselves to deliver the functionality is positively associated with the teams' perceived project success.

The self-organising team construct shows a statistically significant correlation of 0.540 at a confidence level of 0.01 (99%). Thus, the null hypothesis $H_{0\text{-}8}$ is rejected in favour of the hypothesis H_8. The responses obtained in this survey depict a positive association between the independent construct of a self-organising team and the dependent construct of perceived project success.

The observation from this study is consistent with studies showing teams which operate in a *"classic craftsman environment"* (Boehm & Turner, 2003, p. 7), being able to select their own work is critical to the success of an agile implementation. The results of this study therefore concur with existing consensus that self-organising teams are both an important critical success factor for agile implementations (Chow & Cao, 2008) and critical in the success of the project (Misra, Kumar, & Kumar, 2009; Tanner & von Willingh, 2014).

5.2.9 Agile Project Management (H_9)

H_9: Agile project management activities using customer prioritised backlogs and tracking mechanisms based on value delivery is positively associated with the teams' perceived project success

The responses to the survey indicates a statistically significant correlation of 0.473 at a confidence level of 0.01 (99%). Thus, the null hypothesis H_{0-9} is rejected in favour of H_9. Principles seven *"Working software is the primary measure of progress"* (Fowler & Highsmith, 2001, p. 35) of the agile manifesto equates project success to the delivery of software of value, to the customer.

Agile Project Management (APM) within the AMM, focuses the management efforts on metrics to track the delivery of business value (Patel & Ramachandran, 2009a) and improving the quality of software (Verheyen, 2014). Ambler (2011) concludes agile projects are considered successful when deemed to be delivering value to the business, which is tracked by using agile methods such as burn-down charts (Papadopoulos, 2015) and story completion rates being used to measure continuous value delivery and risk reduction (Verheyen, 2014). While not explicitly mentioned as a critical success factor, Chow and Cao (2008) lists the *"Lack of agile progress tracking mechanism"* (Chow & Cao, 2008, p. 963) as a failure factor for agile implementations. Furthermore, Stettina and Hörz (2015) found a properly prioritised backlog to be associated with project success. Thus, the key components of the APM construct, namely the use of prioritised backlogs and tracking mechanisms based measuring value, as found in the results from this study are thus congruent with previous work is a critical component of perceived project success (Ambler, 2011; Chow & Cao, 2008; Stettina & Hörz, 2015).

5.2.10 Defect Prevention (H_{10})

H_{10}: Implementing defect prevention and root cause analysis in favour of future functionality is positively associated with the teams' perceived project success

The defect prevention construct was found to have a statistically significant correlation of 0.473 at a confidence level of 0.01 (99%) with the perceived project success construct. Thus, the null hypothesis H_{0-10} is rejected in favour of H_{10}.

The defect prevention construct focusses on the quality aspects of agile delivery, in favour of future functionality (Patel & Ramachandran, 2009a). As discussed in section 2.2 (Perceived Project Success) and highlighted by Highsmith (2004), Leppänen (2013), quality, though subjective (Prabhakar, 2009; Serrador & Pinto, 2015) is a good indicator of project success. Thus, the observation from this study is aligned with previous findings.

Furthermore, considered in conjunction with the "Test Driven Development" construct, which focusses on quality, a fairly high inter-item correlation of 0.564 is found, showing that the respondents consistently associate the focus on quality with perceived project success. However, the defect prevention construct focuses on the sacrificing future functionality in favour of quality activities (Patel & Ramachandran, 2009a). Given the limitation of this study (refer section 6.2) it is not possible to test whether the observed association holds consistently for all agile roles represented amongst respondents.

5.2.11 Performance Management (H_{11})

H_{11}: Project performance management activities focussing on customer involvement and satisfaction is positively associated with the teams' perceived project success.

The responses obtained for this research indicates a statistically significant correlation of 0.626 at a 0.01 (99%) confidence level, between performance management and perceived project success. Thus, the null hypothesis H_{0-11} is rejected in favour of H_{11}.

The focus of the performance management construct is customer satisfaction and continuous customer involvement (Patel & Ramachandran, 2009a) and was found to have the strongest correlation to perceived project success. Consistent customer involvement in the daily development activities is repeatedly found to be a critical success factor for successful agile implementations (Chow & Cao, 2008; Tanner & von Willingh, 2014), which aligns with the findings of this study.

Though Baruah (2015) cautions against changing requirements too often due to the customers' presence, this study finds this construct to have the strongest positive correlation with perceived project success. This observation could be explained by the distribution of respondents by agile role (refer Table 16). There is a high representation of either business representatives, in the form business representative (10 respondents, 8%), business analyst (8 respondents, 6%) or line management (19 respondents, 14%), which is further evidenced by the observed mean of 4.04 (refer section 4.4.1).

The presence and constant involvement of the customer in successful agile delivery is embedded in agile principle four *"Business people and developers must work together daily throughout the project.* (Fowler & Highsmith, 2001, p. 35)*"*. The observed association between the consistent involvement of the customer with the daily development activities is thus both congruent with the agile manifesto and previous studies highlighting the importance of customer involvement (Hoda, Noble, & Stuart, 2011; Sverrisdottir, Ingason, & Jonasson, 2014).

5.3 Research Questions

All the independent constructs show a statistically significant positive correlation with the independent construct of perceived project success. The secondary research questions posed for this research, namely "How are the specific process areas of the different maturity levels in the AMM associated with perceived project success?", is answered diagrammatically by Figure 14.

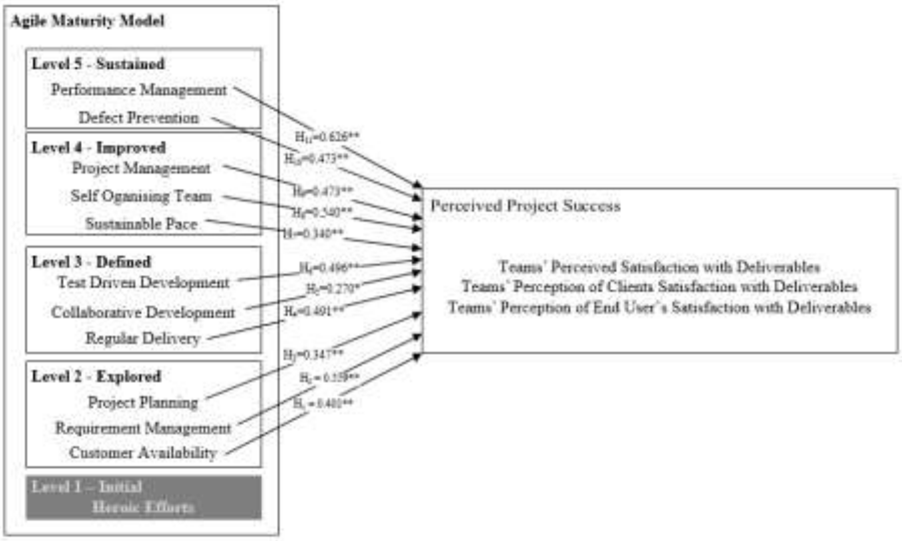

Figure 14: Conceptual model showing correlation found between independent constructs perceived project success

The only construct observed to have a statistically significant correlation > 0.6 is performance management, from maturity level five of the AMM. Given that the focus of this construct is on customer satisfaction (Patel & Ramachandran, 2009a), the underlying philosophy of agile methods, as encapsulated in principle one of the agile manifesto *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software"* (Fowler & Highsmith, 2001, p. 35) is strongly observed in the responses. The performance management construct (PerfMan) in this research specifically focuses on when functionality is accepted as being delivered, stressing all acceptance criteria to have been met as the sole measure of delivery, aligned with agile principle seven *"Working software is the primary measure of progress"* (Fowler & Highsmith, 2001, p. 35). Critical success factor research emphasises regular delivery as a key factor but does not relate it to the satisfaction of the customer (Chow & Cao, 2008). Performance management is thus observed to be a focus area for practitioners to achieve improved perceived project success.

The construct found to have the second strongest statistically significant correlation is Requirements Management (RM), from maturity level two. This construct specifically focuses on understanding what functionality the customer wants to be developed within an iteration. Viewed in conjunction with the performance management construct, it is evident that requirements management shows a statistically significant correlation of 0.456 at a confidence level of 0.01 (99%) with performance management. Thus, practically the ability to deliver what the customer wants (Performance Management construct) is associated with knowing what the customer wants (Requirements Management construct). This is further confirmed by critical factor research (Chow & Cao, 2008; Tanner & von Willingh, 2014). The observed correlation is possibly skewed by the bias in the responses to the Scrum methodology, since requirements management in the context of scrum relies heavily on the presence of the customer during development for clarification of requirements (Baruah, 2015).

The RM is observed to have the strongest statistically significant correlation of 0.504 at a confidence level of 0.01 (99%), with the Regular Delivery construct (RD), from maturity level three of the AMM. Whilst RM focuses on knowing what the customer wants and PerfMan on the satisfaction of the customer, RD in the context of this research focused on the capability of this delivery and itself shows a strong statistically significant correlation with Perceived Project Success (PPS). The observed correlation between RD and PPS is consistent with findings from previous research (França, da Silva, & de Sousa Mariz, 2010). RD in turn shows the strongest statistically significant correlation with the Customer Availability construct (CA), from maturity level two of the AMM, which once again has been shown to be a critical success factor for agile implementations (Tanner & von Willingh, 2014). The observation is aligned with previous findings which show a strong correlation between customer availability and project success (Hoda, Noble, & Stuart, 2011).

The construct found to have the third strongest statistically significant correlation is Self-Organising Team (SOT), from maturity level four of the AMM. This observation is consistent with previous studies which found SOT to be in the top three constructs in terms of strength of correlation to project success (Misra, Kumar, & Kumar, 2009). Interestingly in the observed results, SOT shows the strongest statistically significant correlation with the previously mentioned constructs of PerfMan (0.541), CA (0.540) and RD (0.524) at a confidence level of 0.01 (99%).

The primary research question "How are the different maturity levels of the Agile Maturity Model (AMM) as proposed by Patel and Ramachandran (2009a) associated with perceived project success?" is answered by reviewing the results found in Table 29 and depicted in Figure 15.

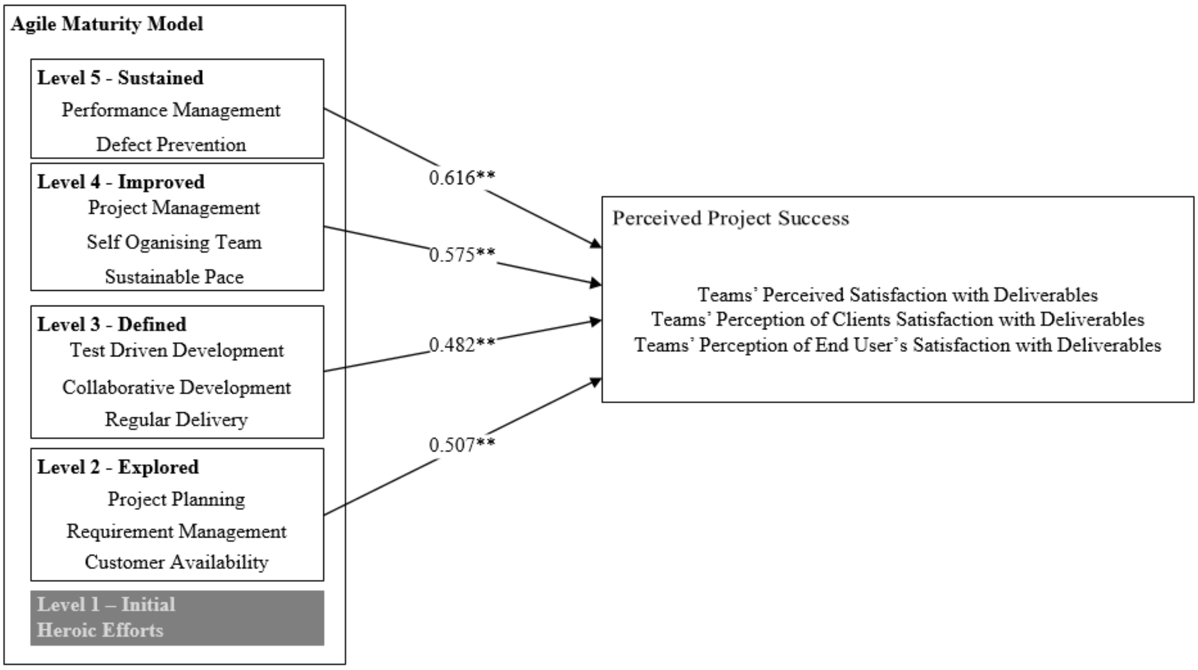

Figure 15: Maturity levels correlation to perceived project success

Interestingly all the maturity levels are found to have statistically significant correlations to the perceived project success construct. Notably the correlation varies from $r = 0.507, p \leq 0.01$ at level two, $r = 0.482, p \leq 0.01$ at level three, $r = 0.575, p \leq 0.01$ at level four with a peak of $r = 0.616, p \leq 0.01$ at maturity level five, as shown in Figure 16. The solid line depicts the specific correlations found for each of the levels whilst the dotted line shows the trend observed across the different maturity levels. As observed, besides the decrease in the strength of the correlation at level three, the correlation increases as the maturity level increases.

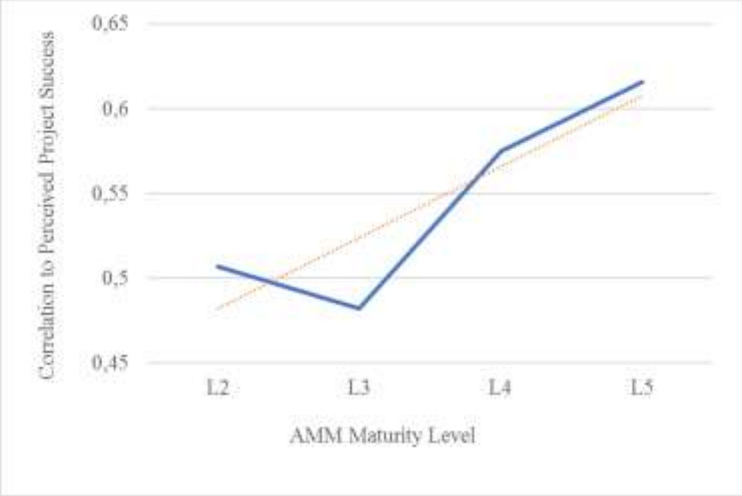

Figure 16: Correlation between maturity level and perceived project success

Notably the observed correlation at level three deviates from the overall observed behaviour of increasing correlation as maturity levels increase. However, it should be noted that the collaborative development construct (CD), which forms part of the level three maturity, as previously discussed in section 4.3.1 was found to have low internal consistency and reliability measures. The decrease in this correlation could thus be influenced by the reliability of the CD and if it was ignored, the observed correlation would change to a value of 0.570 at a confidence level of 0.01 (99%), which would result in a more linear progression between the strength of the correlation and the maturity level, as shown in Figure 17.

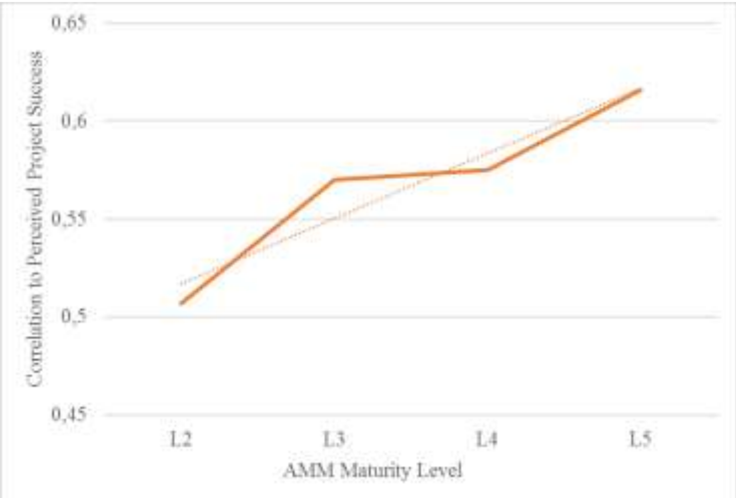

Figure 17: Correlation between maturity level and perceived project success when Collaborative Development construct is omitted

The increasing strength in the correlation as well as the higher correlation at maturity level five, stands in contrast to studies relating CMMI maturity and agile success. This is as expected though, since these previous studies are in consensus CMMI maturity levels above three are difficult to achieve without sacrificing agility (Fritzsche & Keil, 2007; Łukasiewicz

& Miler, 2012; Marçal, et al., 2008; Potter & Sakry, 2009). The results obtained in this study thus both confirm the AMM by Patel and Ramachandran (2009a) and highlight the findings of previous research that the observed behaviour of an agile maturity model differs from the traditional maturity levels in the CMMI.

Thus, in answering the primary research question it is found from the responses obtained, achieving higher levels of agile maturity in the AMM can be associated with an improved perception of project success. Since a maturity model describes the evolution of a process over time, with each successive level of maturity equating to an improvement in the desired outcome (Fontana, Meyer, Reinehr, & Malucelli, 2015), the results of this survey support the AMM as a maturity model for agile implementations.

In providing an answer to the research sub-question “How are the specific process areas of the different maturity levels in the AMM associated with perceived project success?”, the observation shows the activities are interspersed across maturity levels two, four and five but show strong correlation with activities from other maturity levels. The activities in the higher levels of maturity show a reliance on the customer availability construct (CA). Even though CA is found to have a statistically significant correlation with PPS, it was found to be one of the weaker correlations amongst respondents (refer Table 28).

5.4 Summary of Findings

In summary, the results obtained in this study show positive correlation between each of the independent constructs and the dependent construct of perceived project success. Previous research has either found each of the independent constructs to be critical success factors for agile implementations or fundamental to the principles and philosophy of the agile manifesto. Table 30 shows each the hypothesis, with the associated correlation and the existing supporting literature for the finding.

Though a contentious issue with no consensus in literature, collaborative development was found to be positively associated with perceived project success. The independent constructs with the strongest correlations “Performance Management”, “Requirements Management” and “Self-Organising Teams” to perceived project success either have an element of customer involvement and/or collaboration which align strongly with previous findings highlighting the criticality of customer involvement in successful agile implementations.

Table 30: Summary of hypotheses, acceptance/rejection and support literature

Hypothesis	Accept/Reject Hypothesis (Correlation and Significance)	Supporting Literature
H_1: Customer availability in an agile team environment is positively associated with the teams' perceived project success.	Accept (0.401; 0.01)	(Boehm & Turner, 2003) (Chow & Cao, 2008) (Fowler & Highsmith, 2001) (Tanner & von Willingh, 2014)
H_2: Requirement management implemented through the use of story cards which are allowed to change is positively associated with the teams' perceived project success	Accept (0.559; 0.01)	(Ambler, 2014) (Fowler & Highsmith, 2001) (Patel & Ramachandran, 2009b) (Sverrisdottir, Ingason, & Jonasson, 2014)
H_3: Project planning activities based on estimates by the implementation team is positively associated with the teams' perceived project success.	Accept (0.347; 0.01)	(Patel & Ramachandran, 2009b) (Surowiecki, 2004) (Turner, 2014)
H_4: Regular delivery of software to the customer is positively associated with the teams' perceived project success	Accept (0.491; 0.01)	(Chow & Cao, 2008) (Fowler & Highsmith, 2001) (França, da Silva, & de Sousa Mariz, 2010)
H_5: Using collaborative development techniques such as pair programming, peer reviews and collective code ownership is positively associated with the teams' perceived project success	Accept (0.270; 0.05)	(Bipp, Lepper, & Schmedding, 2008) (Chow & Cao, 2008) (Hannay, Dybå, Arisholm, & Sjøberg, 2009) (Fowler & Highsmith, 2001) (Lui & Chan, 2006)

Hypothesis	Accept/Reject Hypothesis (Correlation and Significance)	Supporting Literature
H_6: Using test-driven development practices is positively associated with the teams' perceived project success	Accept (0.496; 0.01)	(Crispin, 2006) (Fowler & Highsmith, 2001) (Sanchez, Williams, & Maximilien, 2007) (Serrador & Pinto, 2015)
H_7: Implementing sustainable pace practices by limiting working hours to forty hours a week is positively associated with the teams' perceived project success	Accept (0.340; 0.01)	(Chow & Cao, 2008) (Fowler & Highsmith, 2001) (Sauter, 2006) (Tanner & von Willingh, 2014)
H_8: Self-organising teams which are allowed to select the work items and organise themselves to deliver the functionality is positively associated with the teams' perceived project success	Accept (0.540; 0.01)	(Boehm & Turner, 2003) (Chow & Cao, 2008) (Misra, Kumar, & Kumar, 2009) (Tanner & von Willingh, 2014)
H_9: Agile project management activities using customer prioritised backlogs and tracking mechanisms based on value delivery is positively associated with the teams' perceived project success	Accept (0.473; 0.01)	(Ambler, 2011) (Chow & Cao, 2008) (Fowler & Highsmith, 2001) (Serrador & Pinto, 2015) (Verheyen, 2014)
H_{10}: Implementing defect prevention and root cause analysis in favour of future functionality is positively associated with the teams' perceived project success	Accept (0.473; 0.01)	(Highsmith, 2004) (Leppänen, 2013) (Prabhakar, 2009) (Serrador & Pinto, 2015)
H_{11}: Project performance management activities focussing	Accept	(Chow & Cao, 2008)

Hypothesis	**Accept/Reject Hypothesis (Correlation and Significance)**	**Supporting Literature**
on customer involvement and satisfaction is positively associated with the teams' perceived project success	(0.626; 0.01)	(Fowler & Highsmith, 2001) (Hoda, Noble, & Stuart, 2011) (Sverrisdottir, Ingason, & Jonasson, 2014) (Tanner & von Willingh, 2014)

6 CONCLUSION

Though the most commonly accepted maturity model, the Capability Maturity Model Integrated (CMMI) (Leppänen, 2013) has been found to be incompatible with agile methods, specifically at higher levels of maturity. Consensus in existing research indicates maturity levels greater than three detract from the agility being sort from implementing agile methods (Fritzsche & Keil, 2007). As such a number of researchers have proposed an agile maturity model aligned to the principles in the agile manifesto (Ambler, 2010; Benefield, 2010; Fontana, Fontana, da Rosa Garbuio, Reinehr, & Malucelli, 2014; Lui & Chan, 2005; Nawrocki, Walter, & Wojciechowski, 2001; Packlick, 2007; Patel & Ramachandran, 2009a; Qumer & Henderson-Sellers, 2008; Sidky, Arthur, & Bohner, 2007). The underlying philosophy of the agile manifesto is embodied in *principle one "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software"* (Fowler & Highsmith, 2001, p. 35). Given the alignment of agile maturity models with the agile principles it can be reasonably assumed achieving higher levels of maturity could be associated with an improvement in the successful delivery of projects.

However, as shown found a number of researchers, project success in the agile context is not strictly governed by the traditional measures of scope, time and budget of the iron triangle. Instead as shown by Serrador and Pinto (2015) agile project success is a more subjective, measured in terms of quality and value to a stakeholder, termed perceived project success. This research was specifically conducted to ascertain whether an association exists between the maturity levels of the Agile Maturity Model (AMM) (Patel & Ramachandran, 2009a) and the Perceived Project Success (Serrador & Pinto, 2015).

Using an objectivist perspective, a quantitative method was employed to analyse the results of an online survey, for which sixty-nine valid responses was obtained. In the context of the individual activities of each of the maturity levels, there is strong alignment between the findings from this study and previous research. Most evident is the alignment of the activities with prior research focussing on critical success factors for agile implementations. The strongest correlations are found between Performance Management (0.626), Requirements Management (0.559) and Self-Organising Teams (0.540). Interestingly these activities occur at different maturity levels within the AMM but all the element of either customer involvement and/or collaboration with the customer embedded. The strong influence of customer involvement in the findings is aligned with previous research in this area. The area of collaborative development remains a contentious issue, however the findings from this study support the school of thought that collaborative development efforts are positively associated with perceived project success.

The data analysis found varying statistically significant positive correlation exists between maturity levels and perceived project success. The strongest correlation was found at the highest maturity level, with relatively weaker correlation at the lower levels of maturity. It can thus be concluded that a higher level of maturity in the AMM is positively associated with perceived project success. The significant contribution from this research is the validation of the conceptual model relating the activities and maturity levels of the AMM as the independent variables to the dependent variable of perceived project success.

6.1 Implication of Findings

The implication of this study for academics is the confirmation of the maturity model developed by Patel and Ramachandran (2009a). This study also shows the association between the individual activities within the maturity levels as well as the maturity levels and the perceived project success, addressing a gap in literature relating these concepts. Also evident from the findings is the strong alignment between the results obtained and previous critical success factor research.

For practitioners, the study has practical implications in highlighting that performance management, requirements management, regular delivery and customer availability are key areas to focus on to establish and continually improve the success of agile implementations. Given the previously mentioned alignment between this research and critical success factor research, this study further assists practitioners in systematically identifying the critical agile activities, such as the use of story cards, continuous delivery and the presence of a knowledgeable customer. With the activities arranged in a proper maturity model the results of this study can guide practitioners as to the order in which activities should be introduced into an environment.

6.2 Limitations of Study

Though the results of this study provide answers to the gaps in literature and has implications for both academics and practitioners, it is not without limitations. The primary limitation of the research is the bias towards more technical roles within an agile team. Although purposeful and snowball sampling was employed the majority of the respondents were from a development role with a limited number of respondents indicating themselves to be business representatives. This could have an influence on the perceived project success being reported in the study.

A further limitation is the limited number of responses obtained. With sixty-nine valid responses, the possible statistical analysis is limited and does not offer the researcher the opportunity to, for example, segment the responses by agile role or find moderating variables, such as industry or experience, within the responses. With a greater sample size, it would be possible to further analyse the results thereby obtaining a richer set of results.

6.3 Direction for Future Research

To address some of the limitations mentioned it would be useful to perform this study, using the same conceptual model as a basis for a qualitative study using case studies. This would allow the researcher the opportunity to strictly define the population as well as obtaining input from the technical and business representatives on the agile teams. Furthermore, this would allow the researcher to provide comparisons on whether the perception of project success is consistent between the technical and business team members. Another method of testing this model would be to include the traditional iron-triangle success measures of scope, schedule and resources to ascertain whether a significant difference exists between the two constructs. Finally, the agile maturity model selected for this research is one of a number currently under discussion in literature. It would be possible to re-use the conceptual model used in this research and substitute a different agile maturity model, to determine if similar results will be obtained.

7 REFERENCES

Abelein, U., & Paech, B. (2015). Understanding the Influence of User Participation and Involvement on System Success – a Systematic Mapping Study. *Empirical Software Engineering*, *20*(1), 28-81.

AgileAlliance. (2015). *Agile Alliance*. Retrieved from Collective Ownership: https://www.agilealliance.org/glossary/collective-ownership/

Agrawal, M., & Chari, K. (2007). Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects. *IEEE Transactions on Software Engineering, 33*(3), 145-156.

Ambler, S. (2010). *The agile maturity model (AMM)*. Retrieved April 22, 2016, from Dr Dobb's: http://www.drdobbs.com/architecture-and-design/the-agile-maturity-model-amm/224201005

Ambler, S. (2011). *Agile Adoption Strategies: November 2011 Survey Results*. Retrieved April 10, 2016, from Ambysoft: http://www.ambysoft.com/surveys/agileStateOfArt201111.html

Ambler, S. (2012a). *Agility at Scale Survey: Results from the Summer 2012 DDJ State of the IT Union Survey*. Retrieved March 28, 2016, from Effective Practices for Software Solution Delivery: http://www.ambysoft.com/surveys/stateOfITUnion201209.html

Ambler, S. (2012b). *Roles on Agile Teams: From Small to Large Teams.* Retrieved June 22, 2016, from Ambysoft: http://www.ambysoft.com/essays/agileRoles.html

Ambler, S. (2014). *Agile Requirements Change Management*. Retrieved September 21, 2016, from Agile Modeling: http://agilemodeling.com/essays/changeManagement.htm

Baruah, N. (2015). The 2015 International Conference on Soft Computing and Software Engineering: Requirement Management in Agile Software Environment. *Procedia Computer Science. 62*, pp. 81-83. Elsevier B.V.

Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional.

Benefield, R. (2010). Seven dimensions of agile maturity in the global enterprise: a case study. *Proceedings of the 43rd Hawaii International Conference on System Sciences*, (pp. 1-7). Honolulu.

Bipp, T., Lepper, A., & Schmedding, D. (2008). Pair programming in software development teams – An empirical study of its benefits. *Information and Software Technology, 50*, 231-240.

Boehm, B., & Turner, R. (2003). People factors in software management: lessons from comparing agile and plan-driven methods. *Management Basics, CROSS TALK-The Journal of Defense Software Engineering*.

Boehm, B., & Turner, R. (2005). Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software, 22*(5), 30-39.

Buchanan, E., & Hvizdak, E. (2009). Online Survey Tools: Ethical and Methodological Concerns of Human Research Ethics Committees. *Journal of Empirical Research on Human Research Ethics*, 37-48.

Bulmer. (1979). *Principles of Statistics.* Courier Corporation.

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems and Software, 81*, 961-971.

Creswell, J. (2009). The Selection of a Research Design. In *Research design: qualitative, quantitative and mixed methods approaches.* Thousand Oaks, California: Sage Publications.

Crispin, L. (2006). Driving Software Quality: How Test-Driven Development Impacts Software Quality. *IEEE Software*, 70-71.

Croasmun, J., & Ostrom, L. (2011). Using likert-type scales in the social sciences. *Journal of Adult Education, 40*(1), 19-22.

Cummins, R., & Gullone, E. (2000). Why we should not use 5-point Likert scales: The case for subjective quality of life measurement. *Proceedings, Second International Conference on Quality of Life in Cities*, (pp. 74-93).

Dingsøyr, T., & Moe, N. B. (2014). Towards Principles of Large-Scale Agile Development: A Summary of the workshop at XP2014 and a revised research agenda. *Conference Paper in Lecture Notes in Business Information Processing.*

Evans, J. R., & Mathur, A. (2005). The value of online surveys. *Internet Research, 15*(2), 195-219.

Fontana, R. M., Fontana, I. M., da Rosa Garbuio, P. A., Reinehr, S., & Malucelli, A. (2014). Processes versus people: How should agile software development maturity be defined? *The Journal of Systems and Software*, *97*, 140–155.

Fontana, R. M., Meyer, V. J., Reinehr, S., & Malucelli, A. (2015). Progressive Outcomes: A Framework for maturing in agile software. *The Journal of Systems and Software, 102,* 88-108.

Fontana, R., Reinehr, S., & Malucelli, A. (2014). Maturing in agile: what is it about? *Proceedings of the 15th International Conference, XP 2014*, (pp. 94-109). Rome.

Fornell, C., & Larcker, D. (1981). Evaluating structural equation models with unobservable. *Marketing Research Journal, 18*(1), 39-50.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35.

França, C., da Silva, F., & de Sousa Mariz, L. (2010). An Empirical Study on the Relationship between the Use of Agile Practices and the Success of Scrum Projects. *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, (p. 37).

Fritzsche, M., & Keil, P. (2007). Agile Methods and CMMI: Compatibility or Conflict? *e-Informatics Software Engineering Journal, 1*(1), 9-26.

Galin, D., & Avrahami, M. (2006). Are CMM Program Investments Beneficial? Analyzing Past Studies. *IEEE Software, 23*(6), 81-87.

Gandomani, T. J., Wei, K. T., & Binhamid, A. K. (2014). A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique. *International Journal of Software Engineering and Its Applications, 8*(11), 173-182.

Garland, R. (1991). The Mid-Point on a Rating Scale: Is it Desirable? *Marketing Bulletin*, 66-70.

Geoghegan, L., & Dulewicz, V. (2008). Do Project Managers' Leadership Competencies Contribute to Project Success? *Project Management Journa, 39*(4), 58–67.

Glazer, H., Dalton, J., Anderson, D., & Konrad, M. D. (2008). *CMMI or agile: Why not embrace both.* Technical Report, Carnegie Mellon University, Software Engineering Institute.

Gonçalves, L., & Linders, B. (2013). *Getting Value out of Agile Retrospectives.*

Gregor, S. (2006). The Nature of Theory In Information Systems. *MIS Quarterly*, *30*(3), 611-642.

Gren, L., Torkar, R., & Feldt, R. (2015). The prospects of a quantitative measurement of agility: A validation study on an agile maturity model. *The Journal of Systems and Software, 107*, 38-49.

Hair, J., & Hult, G. (2016). *A primer on partial least squares structural equation modeling (PLS-SEM).* Sage Publications.

Hall, M. (2010). *Fieldwork in Tourism: Methods, Issues and Reflections.* Routledge.

Hannay, J., Dybå, T., Arisholm, E., & Sjøberg, D. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology, 51*, 1110-1122.

Hastie, S., & Wojewoda, S. (2015). *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. Retrieved March 20, 2016, from InfoQ: http://www.infoq.com/articles/standish-chaos-2015

Heffner, R. (2006). *A Practical Roadmap for Transitioning to CMMI v1.2.* Northrop Grunman.

Henriques, V., & Tanner, M. (2017). A systematic literature review of agile and maturity model research. *Interdisciplinary Journal of Information, Knowledge, and Management, 12,* 53-73. Retrieved from https://www.informingscience.org/Publications/3666

Highsmith, J. (2004). The Agile Revolution. In J. Highsmith, *Agile Project Management.*

Hoda, R., Noble, J., & Stuart, M. (2011). The impact of inadequate customer collaboration on self-organizing Agile teams. *Information and Software Technology*, *53*, 521-534.

Humble, J., & Russel, R. (2009). The agile maturity model applied to building and releasing software. *ThoughtWorks White Paper*. Thoughtworks Web Publishing.

Jiang, J. J., Klein, G., Hwang, H.-G., Huang, J., & Hung, S.-Y. (2004). An exploration of the relationship between software development process maturity and project performance. *Information & Management, 41*, 279-288.

Johnson, C. (2002). The Benefits of PDCA. *Quality Progress, 35*(5), 120.

Joosten, D., Basten, D., & Mellis, W. (2011). Measurement Of Information System Project Success In Organizations – What Researchers Can Learn From Practice. *European Conference on Information Systems 2011 Proceedings*, *Paper 177.*

Jugdev, K., & Müller, R. (2005). A Retrospective Look at Our Evolving Understanding of Project Success. *Project Management Journal*, *36*(4), 19-35.

Kitson, D., Vickroy, R., Walz, J., & Wynn, D. (2009). *An Initial Comparative Analysis of the CMMI Version 1.2 Development Constellation and the ISO 9000 Family*. Carnegie Mellon University.

Kumar, M., Singh, S. K., & Dwivedi, D. R. (2015). A Detail Study of Agile Software Developnment with Extreme Programming. *International Journal of Advanced Research in Computer Science and Software Engineering, 5*(10), 715-729.

Laanti, M. (2014). Characteristics and Principles of Scaled Agile. *Conference Paper in Lecture Notes in Business Information Processing.*

Leppänen, M. (2013). A Comparative Analysis of Agile Maturity Models. In J. C. Rob Pooley (Ed.), *Information Systems Development, Reflections, Challenges and New Directions* (pp. 329-343). Springer New York.

Lin, W., & Shao, B. (2000). The Relationship between User Participation and System Success: A Simultaneous Contingency Approach,. *Information and Management*, *37*(6), 283-295.

Lui, & Chan. (2005). A road map for implementing extreme programming. In *Unifying the Software Process Spectrum; International Software Process Workshop* (Vol. 3840, pp. 474-481). Beijing.

Lui, K. M., & Chan, K. (2006). Pair programming productivity: Novice–novice vs. expert–expert. *International Journl of Human-Computer Studies, 64*, 915-925.

Łukasiewicz, K., & Miler, J. (2012). Improving agility and discipline of software development with the Scrum and CMMI. *IET Software*, *6*(5), 416-422.

Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *The Journal of Systems and Software*, *85*, 2086-2095.

Mandarino, P. (2012). *Leadership in an Agile environment*. Retrieved April 10, 2016, from ThoughtWorks: https://www.thoughtworks.com/insights/blog/leadership-agile-environment

Marçal, A. S., de Freitas, B. C., Soares, F. S., Furtado, M. E., Maciel, T. M., & Belchior, A. D. (2008). Blending Scrum practices and CMMI project management process areas. *Innovations Systems and Software Engineering*, *4*, 17-29.

McHugh, O., Conboy, K., & Lang, M. (2012). Agile practices: The impact on trust in software project teams. *Software, IEEE, 29*(3), 71-76.

McLeod, L., Doolin, B., & MacDonell, S. G. (2012). A Perspective-Based Understanding of Project Success. *Project Management Journal, 43*(5), 68-86.

Miles, M., & Huberman, A. (1994). *Qualitative data analysis: an expanded sourcebook.* Sage Publications.

Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software. *The Journal of Systems and Software, 82,* 1869–1890.

Mitchell, M., & Jolley, J. (2012). Measuring and Manipulating Variables: Reliability and Validity. In M. Mitchell, & J. Jolley, *Research Design Explained* (8th ed., pp. 143-193). Jon-David Hague.

Nagappan, N., Maximilien, E., Bhat, T., & Williams, L. (2008). Realizing quality improvement through test driven development: results and experiences of four industrial teams. *Empirical Software Engineering, 13*(3), 289-302.

Nawrocki, J., Walter, B., & Wojciechowski, A. (2001). Toward maturity model for extreme programming. *Euromicro Conference, 2001. Proceedings. 27th* (pp. 233-239). IEEE.

Nunnally, J. C. (1978). *Psychometric Theory (2nd ed.).* New York: McGraw-Hill.

Packlick. (2007). The agility maturity map – a goal oriented approach to agile improvement. *Agile Conference 2007,* (pp. 266-271).

Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Proceedings of the 3rd International Conference on Strategic Innovative Marketing. 175,* pp. 455-463. Madrid, Spain: Elsevier.

Patel, C., & Ramachandran, M. (2009a). Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices. *International Journal of Software Engineering, 2*(1), 3-28.

Patel, C., & Ramachandran, M. (2009b). Story Card Based Agile Software Development. *International Journal of Hybrid Information Technology, 2*(2), 125-140.

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). *Capability Maturity Model, Version 1.1.* Software Eingineering Institute. Pittsburg: IEEE Software.

Podsakoff, P., MacKenzie, S. B., Lee, J. Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology, 88*(5), 879-903.

Potter, N., & Sakry, M. (2009). Implementing Scrum (Agile) and CMMI Together. *The Process Group Post, 16*(2).

Prabhakar, G. (2009). What is project success: a literature review. *Internation Journal of Business Management, 3*(9), 3-10.

ProcessGroup. (2015). *Maturity levels 2 & 3 Goals and Practices.* The Process Group. Retrieved June 29, 2016, from http://www.processgroup.com/condensed-cmmi1p3-dev-v1.pdf

Przemysław, L. (2013). Time, Budget, And Functionality?—IT Project Success Criteria Revised. *Information Systems Management, 30*(3), 263-275.

Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *The Journal of Systems and Software, 81*, 1899-1919.

Raymond, L., & Bergeron, F. (2008). Project management information systems: An empirical study of their impact on project managers and project success. *International Journal of Project Management*, *26*, 213–220.

Rönkkö, M., Peltonen, J., & Frühwirth, C. (2011). Examining the Effects of Agile Methods and Process Maturity on Software Product Development Performance. In *Software Business* (pp. 85-97). Springer Berlin Heidelberg.

Roszkowski, M. J., & Soven, M. (2010). Shifting gears: consequences of including two negatively worded items in the middle of a positively worded questionnaire. *Assessment & Evaluation in Higher Education, 35*(1), 117-134.

Sanchez, J. C., Williams, L., & Maximilien, E. (2007). On the Sustained Use of a Test-Driven Development Practice at IBM. *Agile Conference (AGILE), 2007*, (pp. 5-14).

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Students* (Fifth ed.). England: Pearson Education Limited.

Sauter, V. (2006). *Extreme Programming.* Retrieved June 12, 2016, from University of Missouri-St. Louis: http://www.umsl.edu/~sauterv/analysis/f06Papers/Hutagalung/#xp_practices

Schweigert, T., Vohwinkel, D., Korsaa, M., Nevalainen, R., & Biro, M. (2014). Agile maturity model: analysing agile maturity characteristics from the SPICE perspective. *Journal of Software: Evolution and Process, 26*(5), 513-520.

SEI. (2010). *CMMI V1.3 Planned Improvements.* Carnegie Mellon University, Software Engineering Institute.

Serrador, P., & Pinto, J. K. (2015). Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management, 33*, 1040–1051.

Shrum, S., & Phillips, M. (2004). CMMI Overview for Executives. Software Engineering Institute.

Sidky, A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in Systems and Software Engineering, 3*(3), 203-216.

Sijtsma, K. (2009). On the use, the misuse, and the very limited usefulness of Cronbach's alpha. *Psychometrika, 74*(1), 107.

Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management, 33*, 140-152.

Stutely, M. (2003). *Numbers Guide: The Essentials of Business Numeracy*. London: Bloomberg Press.

Surowiecki, J. (2004). *The Wisdom of Crowds.* Doubleday; Anchor.

Sutherland, J., Jakobsen, C. R., & Johnson, K. (2008). Scrum and CMMI Level 5: The Magic Potion for Code Warriors. *Proceedings of the 41st Hawaii International Conference on System Sciences* (pp. 1-9). IEEE.

Sverrisdottir, H. S., Ingason, H. T., & Jonasson, H. I. (2014). The role of the product owner in scrum - comparison between theory and practices. *Procedia - Social and Behavioral Sciences. 119*, pp. 257-267. Elsevier Ltd.

Tanner, M., & von Willingh, U. (2014). Factors leading to the success and failure of agile project implemented in traditionally waterfall environments. *Human Capital Without Borders,* (pp. 693-701).

Tarnowski, M. (2014). *CMMI for Development — CMMI-DEV*. Retrieved March 06, 2016, from Plays-In-Business.com: http://www.plays-in-business.com/cmmi-for-development-cmmi-dev/

Team, C. P. (2010). *CMMI for Development, Version 1.3.* Technical Report (CMU/SEI-2010-TR-033), Software Engineering Institute.

Thomas, G., & Fernández, W. (2008). Success in IT projects: A matter of definition? *International Journal of Project Management, 26*, 733–742.

Tongco, M. D. (2007). *Purposive sampling as a tool for informant selection.*

Turner. (2014). *Gower Handbook of Project Management.* Gower Publishing, Ltd., 2014.

van Selm, M., & Jankowski, N. (2006). Conducting Online Surveys. *Quality and Quantity, 40*(3), 435-456.

Verheyen, G. (2014). *Measuring Success, Measuring Value*. Retrieved April 10, 2016, from Scrum.org: https://www.scrum.org/Blog/ArtMID/1765/ArticleID/11/Measuring-Success-Measuring-Value

VersionOne. (2015). *9th Annual State of Agile Report.* Retrieved from Version One: https://www.versionone.com/?s=state+of+agile

VersionOne. (2016). *10th Annual State of Agile Report.* Retrieved March 10, 2016, from VersionOne: https://www.versionone.com/?s=state+of+agile

VersionOne. (2017). *11th Annual State of Agile Report.* Retrieved March 10, 2016, from VersionOne: https://www.versionone.com/?s=state+of+agile

Vlaanderen, K., Jansen, S., Brinkkemper, S., & Jaspers, E. (2011). The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology, 53*, 58-70.

What is Agile? (n.d.). Retrieved February 29, 2016, from Agile Alliance: https://www.agilealliance.org/agile101/what-is-agile/

Williams, B. (2010). Exploratory factor analysis: A five-step guide for novices. *Australian Journal of Paramedicine, 8*(3), 1-13.

Yin, A., da Silva, M. M., & Figueiredo, S. (2011). Scrum maturity model. *Proceedings of the ICSEA*, (pp. 20-29).

8 ADDENDUM A: QUESTIONNAIRE

8.1 Background and Consent

A maturity model is a continuous improvement framework aimed at improving a process. Since agile methodologies are aimed at improved software delivery, an agile maturity model should result in improved project success. This survey aims to gather responses to ascertain whether this assertion is valid. In answering the questions, you may use the context of either the current environment or a past environment in which you (have) work(ed). This research has been approved by the Commerce Faculty Ethics in Research Committee. The questions contain no identifying information and all responses will remain confidential and anonymous. Your participation in this research is voluntary. The questionnaire will take approximately 5 to 8 minutes to complete.

- ☐ I hereby consent to partake in this survey.

8.2 Demographic Information

Specify the industry of the project	*<Dropdown of predefined selection options, including "Other" option>*		*Other: Please specify*				
Specify the agile method being used for the project	*<Dropdown of predefined selection options, including "Other" option>*		*Other: Please specify*				
Approximately how long has the project been using agile?	Less than 1 year	Between 1 and 2 years	Between 2 and 3 years	Between 3 and 4 years	Between 4 and 5 years	Between 5 and 6 years	More than 6 years

How much personal experience do you have using agile?	Less than 1 year	Between 1 and 2 years	Between 2 and 3 years	Between 3 and 4 years	Between 4 and 5 years	Between 5 and 6 years	More than 6 years
What is your current job title?							
What role/roles do you currently fulfil within the agile team?							

8.3 Questionnaire

Customer Availability		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	The customer is present in at the beginning of a development cycle to explain the business requirements					
2	The customer is available daily to answer questions					
3	The customer is NOT knowledgeable in the business domain of the requirements being developed					
Requirements Management		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
4	Requirements are presented using stories (story cards)					
5	The user requirements contain sufficient detail to know what to deliver to satisfy the customer					
6	Detailed tasks can be created from the requirements					

7	Changes are NOT allowed to the user requirements once development has commenced					
Project Planning		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
8	The implementation team estimates the work required for the functionality to be developed in the iteration					
9	Estimation techniques such as planning poker are used					
10	Estimation is done at the start of the iteration cycle					
11	The customer is present during estimation					
12	The estimates provided are used as input in planning the work for the iteration					
13	Availability of team members for the iteration is NOT taken into account when doing the iteration planning					
Regular Delivery		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
14	Functionality developed is demonstrated to the customer at regularly intervals					
15	When functionality is demonstrated the customer provides feedback					
16	Feedback on functionality previously demonstrated is used in future iterations					
Collaborative Development		**Never**	**Rarely**	**Sometimes**	**Often**	**All of the Time**
17	Implementation is done using pair programming					
18	Coding peer reviews are conducted					
19	Developers are allowed to alter any part of the source code or system required to complete a development task					
Test-Driven Development		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
20	Test cases are created before the corresponding code is developed					
21	Test cases are derived from user requirements					

22	All tests cases must pass before promoting code to production					
23	All newly developed code must have accompanying test code					
Sustainable Pace		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
24	Management limit the number of hours worked weekly to a maximum of 40 hours					
25	On average I do not work more than 40 hours a week					
26	I do not work more than 40 hours a week for two consecutive weeks					
Self-Organising Team		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
27	The team look at areas for improvement affecting the successful delivery of functionality					
28	The development team, including the customer, select the work being undertaken for an iteration					
29	The development team are allowed to implement the simplest solution to meet the requirement					
Agile Project Management		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
30	Progress within an iteration is tracked using measures such as burn-down charts or stories/features completed or similar measure					
31	The team only undertake work which can be completed in an iteration					
32	Only work which is of high priority for the customer is undertaken					
Defect Prevention		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
33	If/when bugs are found in production the team (including the customer) allocates time to diagnose and fix the root cause of the problem					
34	If/when bugs are found in production the team (including the customer) test cases are implemented to avoid the future reoccurrence of the bug					

35	If/when bugs are found in production the team (including the customer) the scope of the current iteration is NOT sacrificed in favour of resolving the problem					
Performance Management		**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
36	Functionality is only accepted for development if the acceptance criteria are explicitly stated					
37	Functionality is only considered to be completed once all acceptance criteria have been met					
38	The customer is available throughout the day to clarify requirements and provide feedback					
39	We DO NOT keep track of the number of production bugs being reported within a development iteration					
Perceived Project Success		**Very Dissatisfied**	**Dissatisfied**	**Neutral**	**Satisfied**	**Very Satisfied**
40	How do you rate the project team's satisfaction with the quality of the project deliverables?					
41	How do you rate the customer's satisfaction with the quality of the project deliverables?					
42	How do you rate the end users' satisfaction with the quality of the project deliverables?					
43	How do you rate the project team's satisfaction with the value delivered in the project?					
44	How do you rate the customer’s satisfaction with the value delivered in the project?					
45	How do you rate the end users' satisfaction with the value delivered in the project?					