



Proceedings of the Informing Science + Information Technology Education Conference

An Official Publication
of the Informing Science Institute
InformingScience.org

InformingScience.org/Publications

Online July 7-8, 2020

ASSESSING THE ASSOCIATION BETWEEN AGILE MATURITY MODEL LEVELS AND PERCEIVED PROJECT SUCCESS

Vaughan Henriques University of Cape Town, Cape Town, HNRVAU001@myuct.ac.za
South Africa

Maureen Tanner* University of Cape Town, Cape Town, mc.tanner@uct.ac.za
South Africa

* Corresponding author

ABSTRACT

Background 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 might not fully be compatible with agile implementation, specifically at higher levels of maturity without sacrificing agility. Agile maturity models (AMM), which are aligned to agile principles encourage continuous improvement while maintaining agility.

Aim/Purpose Given the underlying philosophy of the agile manifesto, this study investigates whether an increase in agile maturity is associated with improved perceived project success.

Methodology The study employs a conceptual model based on an existing agile maturity model that is related to perceived project success. Using an objectivist perspective, a quantitative method was employed to analyze the results of an online survey of agile practitioners.

Contribution 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.

Accepted by executive review by Editor Michael Jones | Received: November 19, 2020 | Revised: January 22, February 3, 2020 | Accepted: February 5, 2020.

Cite as: Henriques, V., & Tanner, M. (2020). Assessing the association between agile maturity model levels and perceived project success. In M. Jones (Ed.), *Proceedings of InSITE 2020: Informing Science and Information Technology Education Conference*, pp 117-156. Informing Science Institute. <https://doi.org/10.28945/4519>

(CC BY-NC 4.0) This article is licensed to you under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

Findings	The data analysis found that a 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.
Recommendations 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. 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.
Recommendations for Researchers	The contributions 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.
Future Research	It would be useful to replicate this study whilst following a qualitative approach. The study could also be replicated with a sample consisting of agile project customers.
Keywords	Agile, Scrum, XP, Maturity Model, Agile Maturity Model, Agile Process Improvement, Project Success, Agile Success

INTRODUCTION

As a result of the ever-changing business environment and the need to produce quality software more rapidly (Silva et al., 2015), the agile manifesto was created in 2001. The agile manifesto contains a set of four value statements and twelve principles for agile methodologies (Fowler & Highsmith, 2001). Principle twelve of the agile manifesto encourages teams to strive to improve the process of software delivery continually. In an effort to attain this constant improvement, organizations typically utilize maturity models such as Software Process Improvement (SPI) frameworks. A maturity model is a predefined process improvement model for improving the desired outcome (Fontana et al., 2015).

Nowadays, organizations are increasingly interested in combining maturity and agility within software development projects (Silva et al., 2015). To that effect, this study 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. In general, the intent of a maturity model is the continual improvement of a desired outcome (Fontana et al., 2015). Since the underlying agile philosophy 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.

However, a review of the current literature highlights a lack of research relating to improved maturity in an agile maturity model to project success. While studies exist which relate the maturity levels of Capability Maturity Model Integrated (CMMI) to improved project success, Gren et al. (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 et al., 2015), there is limited guidance 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.

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 organization in which it is being applied.

The purpose of this research is to ascertain whether improving (maturing) in agile discipline is associated with improved perceived project success by providing empirical evidence, independent of the agile method, industry, or organization in which it is being applied. The specific research question is “How are the different maturity levels of the Agile Maturity Model (AMM) as proposed by Patel and Ramachandran (2009a) associated with perceived project success?” The related sub-question is “How are the specific process areas of the different maturity levels in the AMM associated with perceived project success?”

LITERATURE REVIEW

THE AGILE MANIFESTO

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 rapidly (Silva et al., 2015). 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. The agile manifesto does not prescribe a specific methodology but instead provides a set of guiding values and principles.

Though the conceptualization 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 et al., 2008). 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).

AGILE TEAMS INCLUDING BUSINESS REPRESENTATIVE

The agile principles strongly support and promote the concept of a cross-functional, self-organized team (Fowler & Highsmith, 2001). Typically, agile teams consist of five to nine people, having all the roles required to deliver the software (Papadopoulos, 2015), allowing the team to be self-sufficient, which leads to improved project success (Stettina & Hörz, 2015). This is consistent with the ‘whole team’ concept from Extreme Programming (XP) in which “*people with all the skills and perspectives necessary for the project to succeed*” (Beck & Andres, 2004, p. 38) are included.

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). The 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 (Tanner & von Willingh, 2014) and forms the vital link between the business, customer and end-user and the development team (Ambler, 2012).

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, ensuring the delivery of business value. Studies have found a lack of customer feedback can cause a financial loss to business (Hoda et al., 2011). 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.

PERCEIVED PROJECT SUCCESS

Thomas and Fernández (2008) equate the notion of project success to the capturing of Proteus, the mythical sea god of the 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 et al., 2012), there is an agreement of an important distinction to be made between project management success and project success (McLeod et al., 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 a 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.

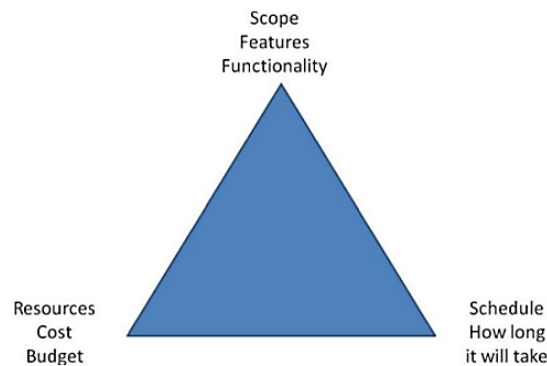


Figure 1: The traditional "Iron-Triangle" of project management (Highsmith, 2004)

A study by Thomas and Fernández (2008) focusing on subjective (perceived) project success was not able to distil a single definition of the concept. This view is further supported by McLeod et al. (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 a 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.

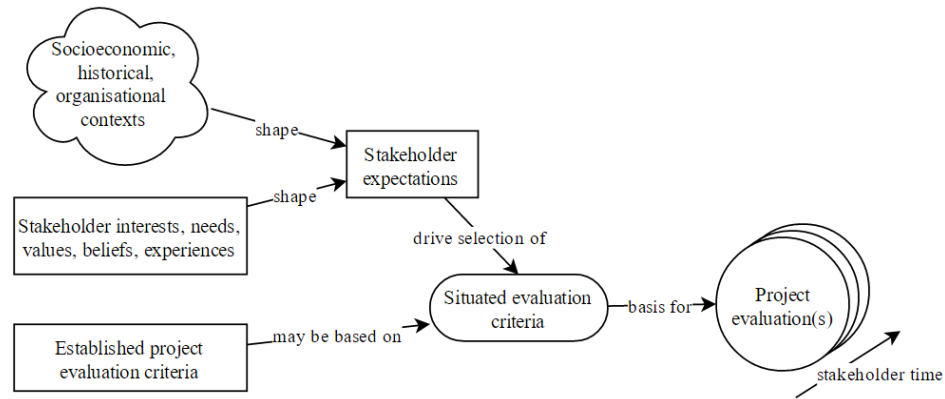


Figure 2: Stakeholder perspective can influence project success evaluation over time (McLeod et al., 2012)

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 focusing 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, 2008, p. 7). Agile methods have altered the traditional view on project success, with the focus shifting more towards stakeholder satisfaction (Leppänen, 2013).

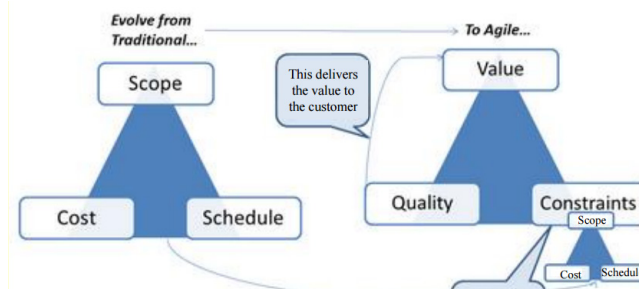


Figure 3: Agile Iron-triangle (Highsmith, 2004)

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. Serrador and Pinto (2015) concluded that the questions relating to stakeholder success are a better indicator of perceived project success. The questions utilized for measuring perceived project success was a combination of three dimensions being the 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 is 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.

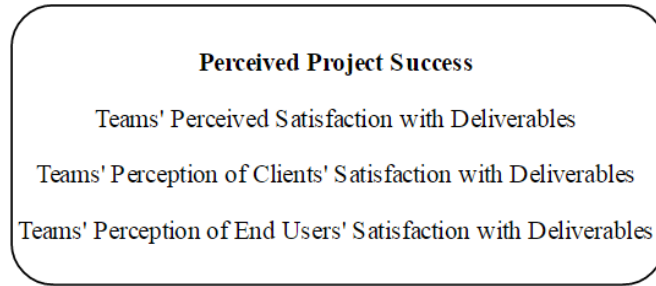


Figure 4: Perceived project success construct – defined as stakeholder satisfaction (Serrador & Pinto, 2015)

MATURITY MODELS

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 et al., 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 et al., 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 et al., 1993).

In the mid-1980s, the Department of Defense (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 et al., 2008). In response, the Software Engineering Institute (SEI) of 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 et al., 2008).

Once publicly available, practitioners produced a number of variants that 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 focusing on three different disciplines, termed constellations in CMMI parlance. The first of these constellations were specifically focused on software development and officially named CMMI-Dev (Kitson et al., 2009). Amongst other changes, the 2010 CMMI version 1.3 release included changes addressing implementation in agile environments (SEI, 2010).

CMMI defines five levels of increasing maturity 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 (SEI, 2010).

As can be seen in Table 1, CMMI level one (initial) is solely reliant on competent individuals with no defined process in place (SEI, 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.

Table 1: CMMI maturity levels, focus, and process areas (The Process Group, 2015)

Level	Focus	Process Area
5 - Optimizing	Continuous Process Improvement	Causal Analysis and Resolution Organizational Performance Management
4 - Quantitatively Managed	Quantitative Management	Organizational Process Performance Quantitative Project Management
3 - Defined	Process Standardization	Decision Analysis and Resolution Integrated Project Management Organizational Process Definition Organizational Process Focus Organizational 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 two (managed) is solely concerned at a project level. Activities take guidance from the governing policies and procedures of the organization and focus on defining and documenting the various activities (SEI, 2010). CMMI level three (defined) focusses on standardizing the various processes across the organization and adhering to these standards (SEI, 2010). 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 (SEI, 2010). The focus can be at a project, process, or organizational level (SEI, 2010). CMMI level five (optimizing) shifts the focus to continuous improvement at an organizational level using Organizational Process Management (OPM) and Causal Analysis and Resolution (CAR). With both OPM and CAR concerned with organizational improvement, the key distinction at this level of maturity is the focus being solely on the organization as a whole (SEI, 2010).

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

Compatibility of CMMI and agile methods

A significant research effort has been invested in attempting to find a level of compatibility between CMMI and agile methods. Consensus exists in 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 organizations could affect the maturity ratings. Fritzsche and Keil (2007) and Łukasiewicz and Miler (2012) attribute this challenge to the nature of the two disciplines with CMMI focusing on the organizational level while agile focuses on the successful delivery of a project. This is consistent with research finding a decline in project success rates (Dingsøyr & Moe, 2014) when organizational 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 three, finding no evidence of support at levels four and five in agile practices.

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 satisfactory compatibility, although acknowledging complete absence in other more organizationally focused process areas.

Research by Sutherland, Jakobsen, and Johnson (2008) found the introduction of the Scrum methodology in a CMMI level five compliant organization had the effect of successfully decreasing the amount of rework required. Whilst at first, this might seem contradictory to literature, it is notable that the introduction of agile methods was after the organization had attained the level five rating (Sutherland et al., 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). 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).

Maturity models for agile environments

The application of agile principle twelve encourages the continuous improvement of the software delivery process, which aligns with the intent of a maturity model. Whilst the CMMI focuses on process maturity; given the people-centric and collaborative nature of agile environments, there is an agreement that a CMMI equivalent agile maturity model should align to agile principles and practices (Fontana, Fontana, et al., 2014; Gren et al., 2015).

A systematic literature review by Henriques and Tanner (2017), focusing on research themes conducted with agile maturity models as the independent variable, found two major themes emerging. The two major themes being, “Agile/CMMI” focusing on “adapting agile practices and principles to fit current software maturity models” (Fontana et al., 2015, p. 89) and “Agile Maturity”, which focus on maturity models based on agile principles, as shown in Figure 5. As depicted, the majority of the articles (59%) focus on the “Agile/CMMI” topic due to the existing investment organizations have in CMMI (Leppänen, 2013).

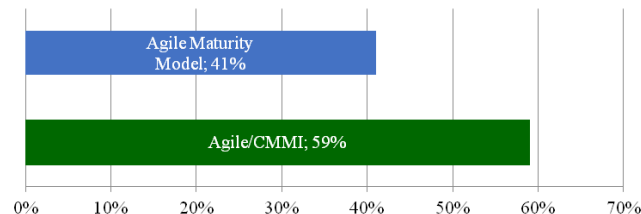


Figure 5: Agile maturity models - research theme distribution (Henriques & Tanner, 2017)

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).

Analysis of proposed agile maturity models

Fontana et al. (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 et al. (2001), Patel and Ramachandran (2009a), and Yin et al. (2011).

Though each of the models provides between three and six maturity levels, there are shortcomings (Fontana et al., 2015; Leppänen, 2013) when viewed relative to the characteristics mentioned in the section on the maturity models or in the scope to which they can be applied, summarized in Table 2.

The maturity models proposed by Benefield (2010), Lui and Chan (2005), Nawrocki et al. (2001) and Yin et al. (2011) are limited to either a Scrum or XP agile methodology. However, agile methods are broader than these methodologies, and an agile maturity model would need to cater to all agile methods by being based on agile principles and not solely on the practices of a specific methodology.

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 et al., 2015; Leppänen, 2013) which are intrinsic characteristics of a maturity model (Leppänen, 2013).

Table 2: Summary of agile maturity models relative to maturity model characteristics and scope (Fontana et al., 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 No success criteria
Qumer and Henderson-Sellers (2008)	No success criteria defined No ability to assess
Yin, da Silva, and Figueiredo (2011)	Limited to Scrum Limited ability to assess

In the analysis of the five-level Agile Maturity Model (AMM) by Patel and Ramachandran (2009a) the conclusion is it “exemplifies how to perform the assessment” (Fontana et al., 2015, p. 91). Thus applying the characteristics of maturity models previously mentioned the requirement to be applicable across any agile method and providing assessment criteria, the working definition of an Agile Maturity Model as proposed by Patel and Ramachandran (2009a) is adopted for this research. Furthermore, it is based on agile practices and principles, and provides predefined focus areas for each maturity level, refer to Figure 6. Each AMM level and focus area are further discussed in the following subsections.

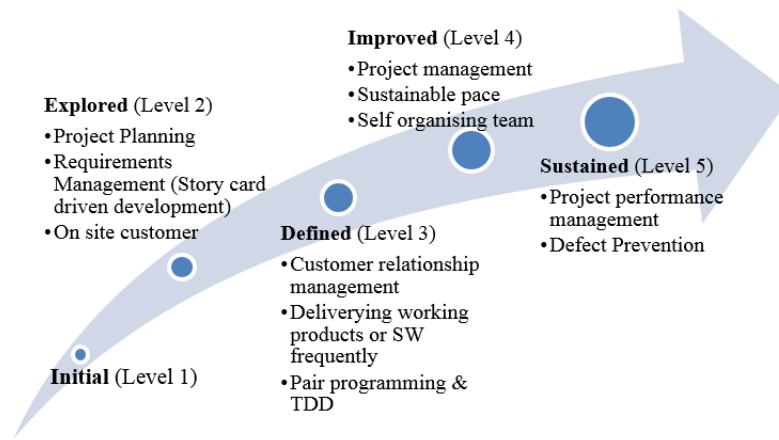


Figure 6: Level AMM for Agile SPI (Patel & Ramachandran, 2009a)

AMM levels and focus areas

Level One: Initial. Level one of the AMM (Initial) is characterized 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 (Explored) activities covers the initial set of focus areas that organizations 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). Lack of proper customer presence and collaboration has been shown to affect prioritization, clarity of requirements, loss of productivity, and “in extreme cases, Business Loss” (Hoda et al., 2011, p. 527). The presence of the customer is vital in the prioritization of requirements (Sverrisdottir et al., 2014), allowing the developer to focus only on functionality important to business, aligned with agile principle ten (Fowler & Highsmith, 2001).

The requirements are 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 standardized 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 et al., 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 et al., 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 et al., 2014). 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 (Defined) shifts the focus to better defining the specific agile implementation (Patel & Ramachandran, 2009a), focusing on the use of technical and technological aspects of the implementation. This level is characterized 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. The AMM at this level of maturity requires “frequent releases which will create a feedback loop” (Patel & Ramachandran, 2009a, p. 10), generally taking the form of a demonstration of the functionality developed during an iteration to the customer. These demonstrations are vital for the customer to provide necessary feedback to the development team and allow the surfacing of any incorrect assumptions made during development, which assist in improving future iterations (Hoda et al., 2011) and rely 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. Unsurprisingly, it remains a contentious issue both in practice and academic research, with findings ranging from showing improved 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 et al., 2008; Hannay et al., 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 (Sanchez et al., 2007). Building on the previous maturity level, the unit tests are derived from the user story.

Level Four: Improved. The foundation having been established in prior maturity levels, level four of the AMM (Improved), focuses on non-technical aspects such as project, team, and people management. It is characterized by a shift toward project management and tracking based on successful delivery (principle seven). Teams are allowed to organize 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) are related to the proper planning, work allocation, and progress tracking. Properly prioritized work lists, known as product backlogs, is the responsibility of the customer to ensure the team works on relatively important work items. A properly prioritized 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 prioritized 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-organizing team leading to an environment where the development team is trusted to get the work done (Stettina & Hörz,

2015). 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). 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 agile practice and implementation is properly established. Level five of the AMM (Sustained) switches the focus to project performance management with the team focusing 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 that in mature agile environments, the availability and commitment of customers can be more influential for success than satisfaction and collaboration (Misra et al., 2009). Furthermore, story cards produced at this maturity level insist 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 favor of future functionality (by agreement with the 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 et al., 2008).

CONCEPTUAL MODEL, CONSTRUCTS AND HYPOTHESES

CONCEPTUAL MODEL

The research questions presented in this study combine concepts from agile principles and practices, maturity models, and perceived projects. Combining these concepts results in the high-level conceptual model shown in Figure 7, which shows the hypothesized relationship between the constructs. The conceptual model for this research relates the levels of the AMM as the independent variable to the dependent variable of perceived project success.

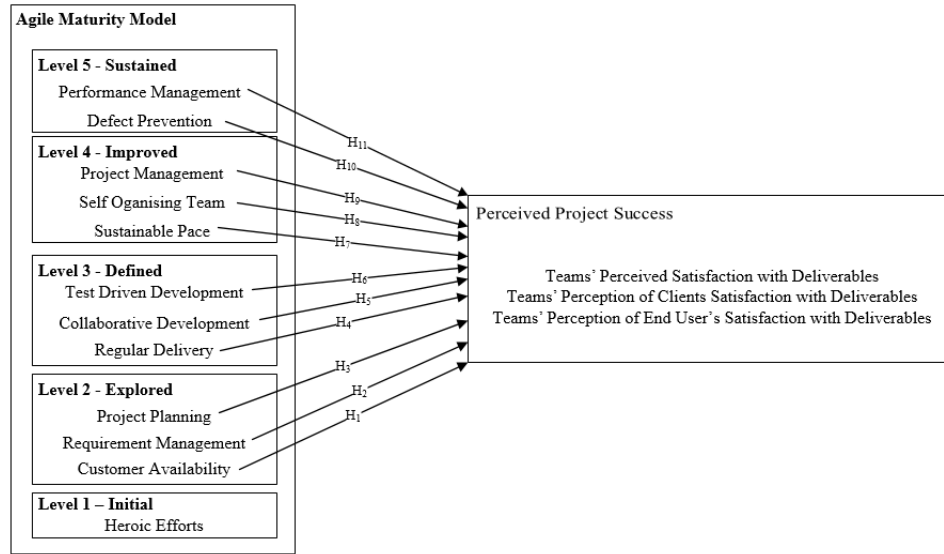


Figure 7: Conceptual model for evaluating the association of focus areas of the AMM (Patel & Ramachandran, 2009a) and perceived project success (Serrador & Pinto, 2015)

The dependent variable for this research will adopt the definition of perceived project success as being stakeholder satisfaction as defined by Serrador and Pinto (2015) and discussed in the section on Perceived Project Success in the literature review.

The AMM, as proposed by Patel and Ramachandran (2009a), forms the basis for the conceptual model presented in Figure 7. More specifically, to answer the sub-question posed in this research, the conceptual model needs to associate each of the focus areas at each maturity level to the perceived project success.

HYPOTHESES

Using the conceptual model gives rise to eleven hypotheses and their corresponding null hypotheses for this study (see Table 3). Evidence from literature in support of the hypotheses is available in section 2.5. Note that the initial level of the AMM is characterized 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.

Table 3: Summary of Hypotheses

AMM Level	Hypotheses	Evidence from literature
Level 1: Initial	N/A	N/A
Level 2: Explored	H ₁ : <i>Customer availability</i> in an agile team environment is positively associated with the teams’ perceived project success. H _{0.1} : Customer availability in an agile team environment is not positively associated with the teams’ perceived project success.	(Patel & Ramachandran, 2009a) (Tanner & von Willingh, 2014) (Hoda et al., 2011)
	H ₂ : <i>Requirement management</i> implemented through the use of story cards which are allowed to change is positively associated with the teams’ perceived project success.	(Patel & Ramachandran, 2009a)

AMM Level	Hypotheses	Evidence from literature
	<p>H_{0.2}: Requirement management implemented through the use of story cards that are allowed to change is not positively associated with the teams' perceived project success</p>	<p>(Sverrisdottir et al., 2014)</p>
	<p>H₃: <i>Project planning</i> activities based on estimates by the implementation team are positively associated with the teams' perceived project success.</p> <p>H_{0.3}: Project planning activities based on estimates by the implementation team are not positively associated with the teams' perceived project success.</p>	<p>(Gandomani et al., 2014)</p> <p>(Mahnič & Hovelja, 2012)</p>
Level 3: Defined	<p>H₄: <i>Regular delivery</i> of software to the customer is positively associated with the teams' perceived project success</p> <p>H_{0.4}: Regular delivery of software to the customer is not positively associated with the teams' perceived project success</p>	<p>(Patel & Ramachandran, 2009a)</p> <p>(Hoda et al. 2011)</p>
	<p>H₅: Using <i>collaborative development</i> techniques such as pair programming, peer reviews, and collective code ownership is positively associated with the teams' perceived project success.</p> <p>H_{0.5}: Using collaborative development techniques such as pair programming, peer reviews, and collective code ownership is not positively associated with the teams' perceived project success.</p>	<p>(Patel & Ramachandran, 2009a)</p>
	<p>H₆: Using <i>test-driven development</i> practices is positively associated with the teams' perceived project success.</p> <p>H_{0.6}: Using test-driven development practices is not positively associated with the teams' perceived project success.</p>	<p>(Crispin, 2006)</p> <p>(Sanchez et al., 2007)</p>
Level 4: Improved	<p>H₇: Implementing <i>sustainable pace practices</i> by limiting working hours to forty hours a week is positively associated with the teams' perceived project success.</p> <p>H_{0.7}: Implementing sustainable pace practices by limiting working hours to forty hours a week is not positively associated with the teams' perceived project success</p> <p>.</p>	<p>(Patel & Ramachandran, 2009a)</p>

AMM Level	Hypotheses	Evidence from literature
	<p>H₈: <i>Self-organizing teams</i> that are allowed to select the work items and organize themselves to deliver the functionality is positively associated with the teams' perceived project success.</p> <p>H_{0,8}: Self-organizing teams that are allowed to select the work items and organize themselves to deliver the functionality is not positively associated with the teams' perceived project success.</p>	<p>(Stettina & Hörz, 2015)</p> <p>(Boehm & Turner, 2003)</p>
	<p>H₉: <i>Agile project management activities</i> using customer prioritized backlogs and tracking mechanisms based on value delivery are positively associated with the teams' perceived project success.</p> <p>H_{0,9}: Agile project management activities using customer prioritized backlogs and tracking mechanisms based on value delivery are not positively associated with the teams' perceived project success.</p>	<p>(Chow & Cao, 2008)</p> <p>(Stettina & Hörz, 2015)</p>
Level 5: Sustained	<p>H₁₀: Implementing <i>defect prevention</i> and root cause analysis in favor of future functionality is positively associated with the teams' perceived project success.</p> <p>H_{0,10}: Implementing defect prevention and root cause analysis in favor of future functionality is not positively associated with the teams' perceived project success.</p>	<p>(Patel & Ramachandran, 2009a)</p> <p>(Patel & Ramachandran, 2009b)</p> <p>(Nagappan et al., 2008).</p>
	<p>H₁₁: <i>Project performance management</i> activities focusing on customer involvement and satisfaction are positively associated with the teams' perceived project success.</p> <p>H_{0,11}: Project performance management activities focusing on customer involvement and satisfaction are not positively associated with the teams' perceived project success.</p>	<p>(Patel & Ramachandran, 2009b)</p>

RESEARCH METHODOLOGY

The objective of this study was to analyze the association between the independent variables represented in the levels of the AMM and the perceived project success, independent of external factors. Thus, an objectivist ontological stance using a positivist view was adopted for the research. Using a deductive approach to prove or disprove the hypotheses, the goal of this research relates to theory type IV of the taxonomy mentioned by Gregor (2006) and is both explanatory and predictive. Saunders et al. (2009) note a key constraint of a deductive approach to ensure rigor is *“that the researcher should be independent of what is being observed.”* (Saunders et al., 2009, p. 125). Thus, to maintain rigor and remove the researcher from the agile environments being researched, a quantitative approach was used. To further maintain the independence between the researcher and the participants, a question-

naire was employed as the inquiry strategy, using an online survey for ease of distribution and the ability to reach a wider audience.

RESEARCH INSTRUMENT

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 categorize the respondent without identifying the individual or the organization, consistent with the objectivist ontology.

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 was used specifically since Cummins and Gullone (2000) advise that a broader selection of options allows 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 was used to allow a neutral response option. The absence of a neutral response option could influence the results obtained since respondents would be forced to adopt a stance (Croasmun & Ostrom, 2011). The forced response would tend to more socially acceptable answers, thus skewing the results (Garland, 1991).

The wording of the survey questions was 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’*” and to avoid “*automatic, cognitive processing*” (Podsakoff et al., 2003, p. 884). Thus, where possible, each construct will include a negatively worded statement.

As previously mentioned, the constructs representing agile maturity levels were adopted from the AMM based on the model by Patel and Ramachandran (2009a), which includes qualitative assessment criteria for each process area. Since the study was quantitative, statements developed adapted the assessment criteria from the AMM to be more suitable to a quantitative study and a Likert scale. Various statements, in line with constructs of the framework, were formulated accordingly. The questionnaire is included in the Appendix.

TARGET POPULATION & SAMPLE

The objective and questions posed in this study are independent of any specific agile methodology. The AMM utilized 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 (2012) 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 practicing any agile methodology and cannot be limited to any specific job title.

With the focus on maturity levels, organizations representing the five levels of maturity needed 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 organization. Based on CMMI, Shrum and Phillips (2004) provide guidance suggesting organizations can take up to seven years to obtain a higher level of maturity, which was used as guidance in terms of the population being approached.

Given no comprehensive database exists which defines the sample frame of agile practitioners, a combination of non-probabilistic sampling approaches was used (Saunders et al., 2009). Purposive sampling was initially utilized, 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 et al., 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 et al., 2009).

Non-probabilistic sampling suffers from common limitations. Most notably, these include bias and self-selection; potentially leading to a homogenous sample (Saunders et al., 2009). However, given the objective is to obtain responses from agile practitioners, the initial use of purposive sampling should mitigate this limitation.

RESULTS

The responses to the survey were exported from Qualtrics into Microsoft Excel (version 2007) to perform the initial data cleansing and analysis. The total number of respondents to the survey was ninety-six. Two of the respondents opted not to 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. 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 et al., 2009, p. 218). The completed responses were then imported into SPSS for further statistical analysis.

DESCRIPTIVE ANALYSIS

This section describes the background of the respondents in line with the questions asked in the questionnaire, i.e. industry, the agile methodology being used, the 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-square 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.

Responses by industry

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 (see Table 4). The response distribution achieved is consistent with annual international surveys where financial services and insurance comprise approximately 20% of responses (VersionOne, 2016). The chi-squared result $X^2(2) = 0.000$, $p \leq 0.05$ indicates the difference in the responses obtained by industry is statistically signifi-

cant, with a bias towards the financial services industry. Given the non-probabilistic sampling technique utilized, this outcome is expected.

Table 4: 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%
Medical/Health Services	3	4%
Retail	10	14%
Transportation	1	1%
Other	4	6%
Total	69	100%

Responses by Agile methodology

Consistent with findings from international surveys (VersionOne, 2016), the findings from this study indicate Scrum to be in use in the majority (75%), 52 responses (see Table 5). Interestingly none of the respondents indicated using XP. This is expected as the current trend in the use of agile methods internationally shows that XP is consistently on the decline (VersionOne, 2016). The chi-squared result $X^2(2) = 0.000$, $p \leq 0.05$ indicates the difference in the responses obtained by agile methodology is statistically significant, with a bias towards the scrum methodology. However, given the alignment to international trends and findings (VersionOne, 2016), the response to this survey is consistent with other observations and thus properly representative of the agile population.

Table 5: 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%

Responses by job title and agile role of respondents

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) (see Table 6). The prevalence of developers as well as solution architects is unsurprising as Ambler (2012) mentions the predominance of both the developers and architects in agile teams, particularly in large corporate environments. The chi-squared result $X^2(2) = 0.000$, $p \leq 0.05$ indicates the difference in the responses obtained by job title is statistically significant, with a bias towards respondents having a job title of “developer”.

In contrast, the distribution of respondents by the agile role being fulfilled confirms the difference between roles and job titles in agile teams (Ambler, 2012), with the developer role being the most common amongst respondents. Notably, a number of respondents perform multiple roles. The chi-squared result $X^2(2) = 0.849$, $p \leq 0.05$ 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 6: Response distribution of job title and agile role 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%
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%
Test Analyst	1	1%
VP of Technology	1	1%
Team leader	1	1%
Agile coach	1	1%
Total	69	100%

Responses by years Agile in use for project

The responses of the number of years projects have been using agile methodologies are evenly distributed across the available year ranges. The majority of respondents, 50 responses (72%) indicate using agile methodologies for three years or less (see Table 7). The normal distribution and descriptive statistics results 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 and slightly positively skewed.

Table 7: 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%
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%

Responses by Agile experience of respondents

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 (see Table 8). The normal distribution tests conducted for agile experience indicate the variable is not normally distributed, with an SW measure of 0.869 and a $p - value < 0.001$. The skewness value of -0.410 indicates the distribution is fairly symmetrical and slightly negatively skewed. The responses obtained are consistent with findings from international studies (VersionOne, 2016), which show the majority of respondents having more than four years of agile experience (VersionOne, 2017).

Table 8: 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%

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 re-

search (Fornell & Larcker, 1981), where a construct was found to have $\alpha < 0.6$, the test was extended to determine the effect of 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 (ρ) 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. The constructs Customer Availability, Requirements Management, Self-Organizing 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 $\alpha = 0.9$ showing high internal consistency. 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.

Collaborative development

A Cronbach of $\alpha = 0.5$ was obtained for the Collaborative Development construct. The exclusion of any of the questions does not improve the reliability measure of the construct (refer to Table 10). Table 9 shows the results of the ρ 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 a statistically insignificant correlation with these questions. CD19 is thus excluded from further analysis and the construct is represented by CD17 and CD18.

Table 9: Internal consistency Spearman (ρ) correlation for Collaborative Development Construct

Variable	CD17	CD18	CD19
CD17	1.000		
CD18	0.349**	1.000	
CD19	0.099	0.178	1.000
** Correlation is significant at the 0.01 level (2-tailed)			

Table 10: Effect on Cronbach alpha measure when deleting questions

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

Defect prevention

A Cronbach of $\alpha = 0.4$ was obtained for the Defect Prevention construct; however as shown in Table 12, the exclusion of any of the questions does not improve the reliability of the construct. Table 11 shows the results of the ρ 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 a 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 11: 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 12: Effect on Cronbach alpha measure when deleting questions

Variable	Cronbach's Alpha if Item Deleted
DP33	0.115
DP34	0.246
DP35	0.547

CONSTRUCT NORMALITY AND DISTRIBUTION

Each construct was tested for normality using the Shapiro-Wilk (SW) test for normality to determine the statistical approach to be applied. With *p – value* < 0.001, the constructs of CA, RM, RD, and PPS are non-normally distributed. The constructs PP, SOT, and DP were found to have *p – value* = 0.001, and thus also have a non-normal distribution. The remaining constructs of CD, *p – value* < 0.005, TDD, *p – value* < 0.007, SP, *p – value* < 0.026, APM, *p – value* < 0.002 and PM, *p – value* < 0.141 were all found to have a *p – value* > 0.001 and thus the responses were normally distributed. Since there is a mixture of normally and non-normally distributed responses, non-parametric statistical methods were used for further analysis (Saunders et al., 2009).

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 et al., 2009). Since the conceptual model being investigated in this research and the corresponding hypotheses are unidirectional, a one-tailed correlation was employed. 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 hypothesized correlation 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 et al., 2009).

Table 13 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 PerfMng while the weakest correlation of 0.270 is observed for CD.

Table 14 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 a 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 13: Summary of Spearman rho Correlation of independent constructs to the dependent construct of Perceived Project Success

Construct	Correlation
Performance Management	0.626**
Requirement Management	0.559**
Self-Organizing 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 14: Summary of Spearman rho Correlation of independent constructs to the dependent construct of Perceived Project Success

Construct	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).	

FINDINGS AND DISCUSSION

HYPOTHESES ANALYSIS

Hypothesis 1: Customer availability

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_{0-1} is rejected in favor 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 factor for agile environments (Chow & Cao, 2008; Tanner & von Willingh, 2014) and further supports principle four of the agile manifesto (Fowler & Highsmith, 2001). 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 are positively associated with project success (Abelein & Paech, 2015).

Hypothesis 2: Requirements management

Responses to this survey show a statistically significant correlation of 0.559 at 0.01 (99%) confidence level between the requirements management and perceived project success construct. The null hypothesis $H_{0.2}$ is rejected in favor 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 perceived project success. Furthermore, the findings from this research are consistent with principle two of the agile manifesto (Fowler & Highsmith, 2001) with Ambler (2014), and Sverrisdottir et al. (2014) highlighting the positive influence changing requirements can have on perceived project success.

Hypothesis 3: Project planning

The responses for this survey show a statistically significant correlation of 0.347 at 0.01 (99%) confidence level between project planning and perceived project success. Thus, the null hypothesis $H_{0.3}$ is rejected in favor of H_3 .

The results confirm that the use of agile estimation techniques 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.

Hypothesis 4: Regular Delivery

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 $H_{0.4}$ is rejected in favor of H_4 .

The positive association found between regular delivery and perceived project success is confirmation of the underlying philosophy of an agile implementation, embodied in principle one and seven of the agile manifesto (Fowler & Highsmith, 2001). Furthermore, these results align with existing consensus in literature that regular delivery is a critical success factor for agile implementations (Chow & Cao, 2008; França et al., 2010), with França et al. (2010) finding it to have the strongest correlation with project success.

Hypothesis 5: Collaborative development

Though the agile manifesto encourages collaborative practices as per principle six (Fowler & Highsmith, 2001), 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 favor 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 et al., 2008), working only for new and complex problems when the proper mix of skills, personality and expertise are involved (Hannay et al., 2009). More specifically, collaborative development techniques have been found to be less effective amongst experienced developers (Hannay et al., 2009) and since the majority of the respondents for this study have more than six years of agile experience the weak correlation found is possibly due to prevalence of more experienced re-

spondents, aligning to existing findings (Hannay et al., 2009). Due to the limitations of this study 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.

Hypothesis 6: Test-driven development

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 $H_{0.6}$ is rejected in favor of H_6 .

As previously discussed, 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, 2008, p. 7). Test-driven development activities have been shown to improve the quality of the implemented software (Sanchez et al., 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.

Hypothesis 7: Sustainable pace

The sustainable pace construct shows a statistically significant correlation of 0.340 at a 0.01 (99%) confidence level. Thus, the null hypothesis $H_{0.7}$ is rejected in favor 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 Willingham, 2014). The finding from this research thus aligns with critical success factor research (Chow & Cao, 2008; Tanner & von Willingham, 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 et al. (2010) found sustainable pace to not correlate with project success. A possible explanation for this apparent misalignment of findings, is this study does not suffer from the same generalization restriction. The generalization restriction in the study by França et al. (2010) is primarily due to the size and localization of the survey participants. Though the current study has a similar response size, the sample does not suffer from the same localization restriction.

Hypothesis 8: Self-organizing team

The self-organizing team construct shows a statistically significant correlation of 0.540 at a confidence level of 0.01 (99%). Thus, the null hypothesis $H_{0.8}$ is rejected in favor of the hypothesis H_8 . The responses obtained in this survey depict a positive association between the independent construct of a self-organizing team and the dependent construct of perceived project success.

The observation from this study is consistent with studies showing teams that 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 (Chow & Cao, 2008) and is shown to be critical in the success of the project (Tanner & von Willingham, 2014).

Hypothesis 9: Agile project management

The responses to the survey indicate 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 favor 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.

Hypothesis 10: Defect prevention

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 favor of H_{10} .

The defect prevention construct focusses on the quality aspects of agile delivery, in favor of future functionality (Patel & Ramachandran, 2009a). As discussed previously and highlighted by Highsmith (2004) and Leppänen (2013), quality, though subjective (Prabhakar, 2008; 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 favor of quality activities (Patel & Ramachandran, 2009a). Given the limitation of this study it is not possible to test whether the observed association holds consistently for all agile roles represented amongst respondents.

Hypothesis 11: Performance management

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 favor 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 (Tanner & von Willingh, 2014), which aligns with the findings of this study. The strength of the correlation could be explained by the distribution of respondents by agile role. 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.

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 involvements 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 et al., 2011; Sverrisdottir et al., 2014).

DISCUSSION OF THE FINDINGS

All the constructs show a statistically significant positive correlation with the 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 8.

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 is strongly observed in the responses. The PerfMan construct 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 emphasizes 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 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).

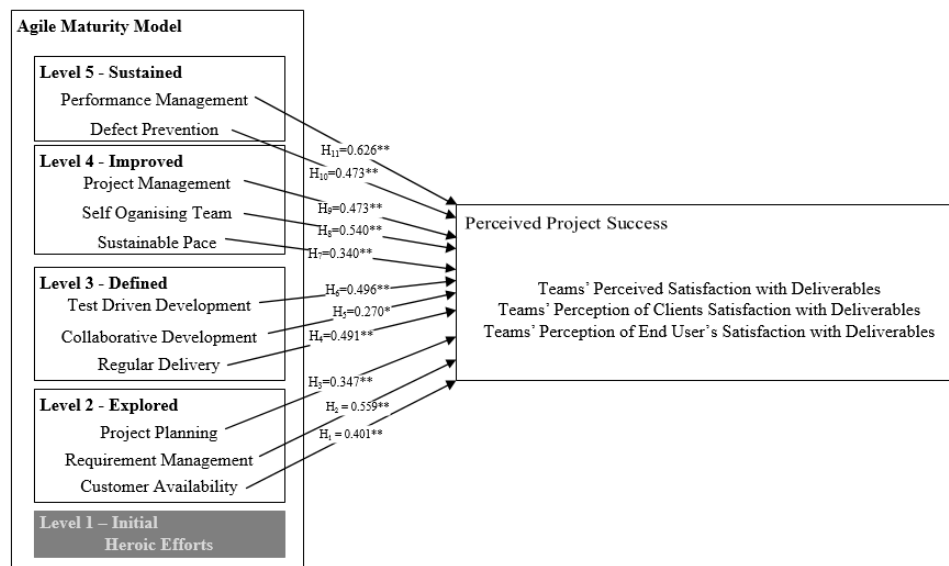


Figure 8: Conceptual model showing correlation found between constructs and perceived project success

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 et al., 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 et al., 2011).

The construct found to have the third strongest statistically significant correlation is Self-Organizing 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 et al., 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 14 and depicted in Figure 9. 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 10. 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.

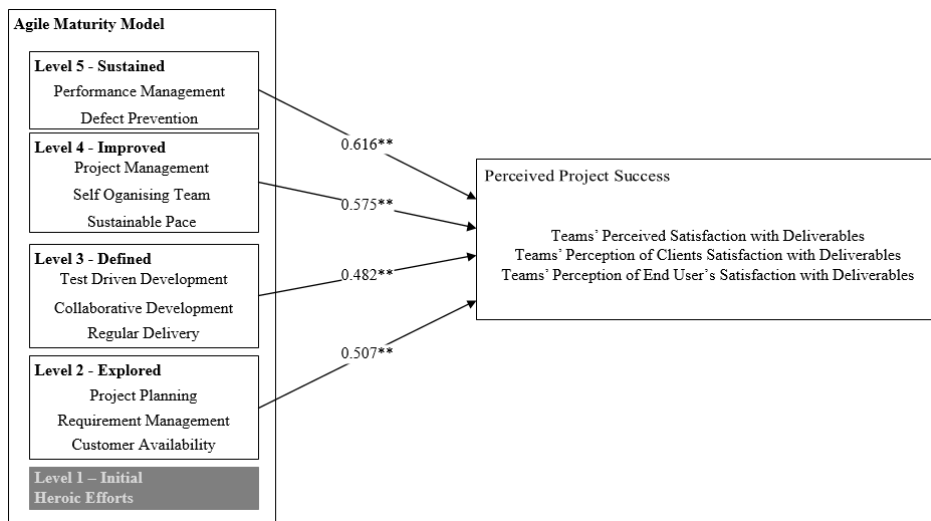


Figure 9: Maturity levels correlation to perceived project success

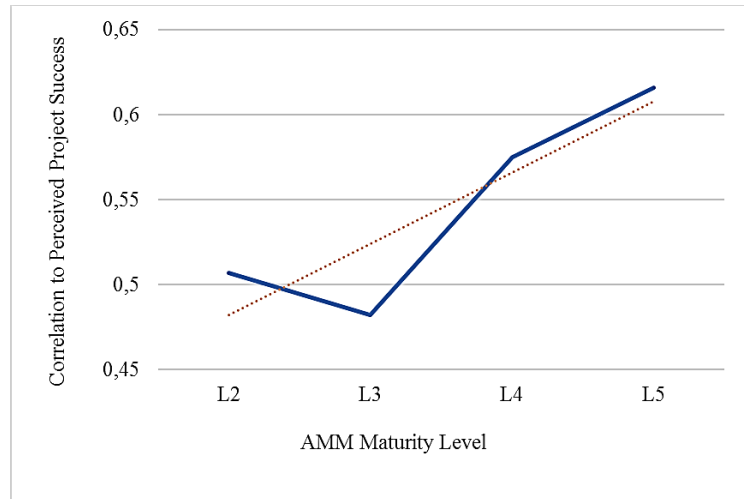


Figure 10: Correlation between maturity level and perceived project success

Notably the observed correlation at level three deviates from the overall observed behavior 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 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 11.

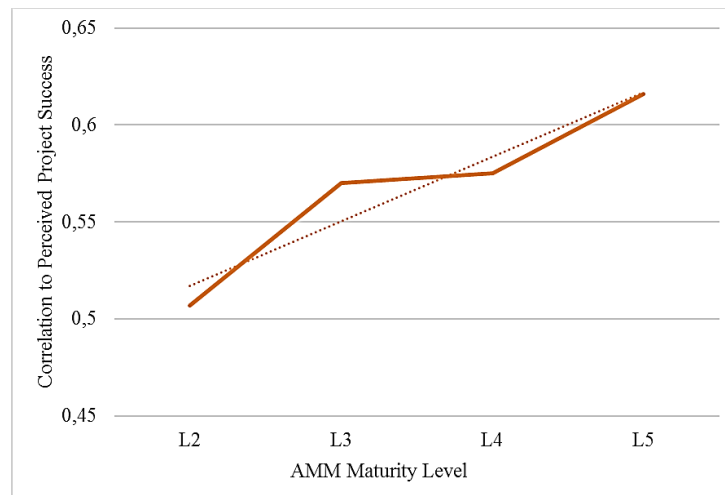


Figure 11: Correlation between maturity level and perceived project success when Collaborative Development construct is omitted

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 behavior 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 et al., 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.

CONCLUSION

Though the most commonly accepted maturity model, the Capability Maturity Model Integrated (CMMI) (Leppänen, 2013) has been found to be less compatible with agile methods, particularly at higher levels of maturity. 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. 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).

The sixty-nine responses were analyzed using correlation analysis and found to strongly align with prior research focusing on critical success factors for agile implementations. The strongest correlations are found between Performance Management (0.626), Requirements Management (0.559) and Self-Organizing 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 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 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. For practitioners, the study has practical implications in systematically identifying the critical agile activities, such as the use of story cards, continuous delivery and the presence of a knowledgeable customer to improve perceived project success. 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.

The primary limitation of the research is the bias towards more technical roles within an agile team, with the majority of the respondents from a development role with a limited number of respondents indicating themselves to be business representatives. 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.

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. This would allow the researcher the opportunity to strictly define the population as well as obtaining input from the technical and business representatives from the same agile teams. Additionally, this would allow the researcher to provide comparisons on whether the perception of project success is consistent between the technical and busi-

ness team members. 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.

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. <https://doi.org/10.1007/s10664-013-9278-4>
- Ambler, S. W. (2010, April 01). *The agile maturity model (AMM)*. Retrieved April 22, 2016, from Dr Dobb's: <http://www.drdoobs.com/architecture-and-design/the-agile-maturity-model-amm/224201005>
- Ambler, S. W. (2011, November). *Agile adoption strategies: November 2011 survey results*. Ambysoft. <http://www.ambysoft.com/surveys/agileStateOfArt201111.html>
- Ambler, S. W. (2012). *Roles on Agile teams: From small to large teams*. Ambysoft. <http://www.ambysoft.com/essays/agileRoles.html>
- Ambler, S. W. (2014). *Agile requirements change management*. Agile Modeling: <http://agilemodeling.com/essays/changeManagement.htm>
- Baruah, N. (2015). Requirement management in agile software environment. *Procedia Computer Science*, 62, 81-83. <https://doi.org/10.1016/j.procs.2015.08.414>
- 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, HI: IEEE Press. <https://doi.org/10.1109/HICSS.2010.337>
- Boehm, B., & Turner, R. (2003). People factors in software management: Lessons from comparing agile and plan-driven methods. *CROSS TALK – The Journal of Defense Software Engineering*, 10(12), 4–8. <https://pdfs.semanticscholar.org/17e7/447865cf9971d78b6b4198cc51f0567ccd8a.pdf>
- Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30-39. <https://doi.org/10.1109/MS.2005.129>
- Bipp, T., Lepper, A., & Schmedding, D. (2008). Pair programming in software development teams – An empirical study of its benefits. *Information and Software Technology*, 50(3), 231-240. <https://doi.org/10.1016/j.infsof.2007.05.006>
- Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961-971. <https://doi.org/10.1016/j.jss.2007.08.020>
- Crispin, L. (2006). Driving software quality: How test-driven development impacts software quality. *IEEE Software*, 23(6), 70-71. <https://doi.org/10.1109/MS.2006.157>
- Croasmun, J., & Ostrom, L. (2011). Using likert-type scales in the social sciences. *Journal of Adult Education*, 40(1), 19-22. <https://files.eric.ed.gov/fulltext/EJ961998.pdf>
- Cummins, R., & Gullone, E. (2000). Why we should not use 5-point Likert scales: The case for subjective quality of life measurement. *Proceedings of the Second International Conference on Quality of Life in Cities* (pp. 74-93). Singapore: National University of Singapore. <https://mafiadoc.com/why-we-should-not-use-5-point-likert-scales-the-5a4e731e1723dd593f91bcfe.html>
- Dingsøyr, T., & Moe, N. B. (2014). Towards principles of large-scale agile development. In T. Dingsøyr, N. B., Moe, R. Tonelli, S. Counsell, C. Gencel, & K. Petersen (Eds.), *Agile methods: Large-scale development, refactoring, testing, and estimation* (pp. 1-8). XP 2014 International Workshops. Rome, Italy: Springer. <https://doi.org/10.1007/978-3-319-14358-3>

- 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. <https://doi.org/10.1016/j.jss.2014.07.030>
- Fontana, R. M., Reinehr, S., & Malucelli, A. (2014). Maturing in agile: What is it about? *Proceedings of the 15th International Conference on Agile Software Development* (pp. 94-109). Rome: Springer, Cham. https://doi.org/10.1007/978-3-319-06862-6_7
- Fontana, R. M., Meyer, V. J., Reinehr, S., & Malucelli, A. (2015). Progressive outcomes: A Framework for maturing in agile software. *Journal of Systems and Software*, 102, 88-108. <https://doi.org/10.1016/j.jss.2014.12.032>
- Fornell, C., & Larcker, D. (1981). Evaluating structural equation models with unobservable. *Marketing Research Journal*, 18(1), 39-50. <https://doi.org/10.1177/002224378101800104>
- Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35. <http://users.jyu.fi/~mieijala/kandimateriaali/Agile-Manifesto.pdf>
- França, A. C. C., da Silva, F. Q., & de Sousa Mariz, L. M. (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* (pp. 1-4). <https://doi.org/10.1145/1852786.1852835>
- Fritzsche, M., & Keil, P. (2007). Agile methods and CMMI: Compatibility or conflict? *e-Informatica Software Engineering Journal*, 1(1), 9-26. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.2676&rep=rep1&type=pdf>
- Garland, R. (1991). The mid-point on a rating scale: Is it desirable? *Marketing Bulletin*, 2(1), 66-70. http://scorevoting.net/MB_V2_N3_Garland.pdf
- 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. http://www.academia.edu/download/35919030/16_2.pdf
- Glazer, H., Dalton, J., Anderson, D., & Konrad, M. D. (2008). *CMMI or agile: Why not embrace both!* Technical Report. Pittsburgh, PA: Carnegie Mellon University, Software Engineering Institute. https://kithub.cmu.edu/articles/CMMI_or_Agile_Why_Not_Embrace_Both_/6572513/files/12057554.pdf
- Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, 30(3), 611-642. <https://doi.org/10.2307/25148742>
- Gren, L., Torkar, R., & Feldt, R. (2015). The prospects of a quantitative measurement of agility: A validation study on an agile maturity model. *Journal of Systems and Software*, 107, 38-49. <https://doi.org/10.1016/j.jss.2015.05.008>
- Hannay, J., Dybå, T., Arisholm, E., & Sjøberg, D. (2009). The effectiveness of pair programming: A meta-analysis. *Information and Software Technology*, 51, 1110-1122. <https://doi.org/10.1016/j.infsof.2009.02.001>
- Hair, J., & Hult, G. (2016). *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage Publications.
- Henriques, V., & Tanner, M. (2017). A systematic literature review of agile maturity model research. *Interdisciplinary Journal of Information, Knowledge, and Management*, 12, 53-73. <https://doi.org/10.28945/3666>
- Hoda, R., Noble, J., & Stuart, M. (2011). The impact of inadequate customer collaboration on self-organizing agile teams. *Information and Software Technology*, 53(5), 521-534. <https://doi.org/10.1016/j.infsof.2010.10.009>
- Highsmith, J. (2004). *Agile project management: Creating innovative products*. Addison-Wesley. https://doi.org/10.1007/978-3-540-27777-4_45
- Johnson, C. (2002). The benefits of PDCA. *Quality Progress*, 35(5), 120. <https://search.proquest.com/openview/6fb24b731a9c0c8bafd90096fd751e76/1>

- Joosten, D., Basten, D., & Mellis, W. (2011). Measurement of information system project success in organizations – What researchers can learn from practice. *Proceedings of the 2011 European Conference on Information Systems* (Paper 177). <https://aisel.aisnet.org/ecis2011/177>
- Jugdev, K., & Müller, R. (2005). A retrospective look at our evolving understanding of project success. *Project Management Journal*, 36(4), 19-35. <https://doi.org/10.1177/875697280503600403>
- 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, Software Engineering Institute. <https://doi.org/10.21236/ADA501760>
- Laanti, M. (2014, May). Characteristics and principles of scaled agile. In T. Dingsøyr, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, & K. Petersen (Eds.), *Agile methods. Large-scale development, refactoring, testing, and estimation* (pp. 9-20). XP 2014. Springer. https://doi.org/10.1007/978-3-319-14358-3_2
- Leppänen, M. (2013). A comparative analysis of agile maturity models. In R. Pooley, J. Coady, C. Schneider, H. Linger, C. Barry, & M. Lang (Eds.), *Information systems development: Reflections, challenges and new directions* (pp. 329-343). Springer. https://doi.org/10.1007/978-1-4614-4951-5_27
- Lukasiewicz, K., & Miler, J. (2012). Improving agility and discipline of software development with the Scrum and CMMI. *IET Software*, 6(5), 416-422. <https://doi.org/10.1049/iet-sen.2011.0193>
- Lui, K. M., & Chan, K. C. (2005). A road map for implementing extreme programming. In M. Li, B. Boehm, & L. J. Osterweil (Eds.), *Unifying the software process spectrum* (pp. 474-481). Springer. https://doi.org/10.1007/11608035_38
- Lui, K. M., & Chan, K. C. (2006). Pair programming productivity: Novice–novice vs. expert–expert. *International Journal of Human-Computer Studies*, 64(9), 915-925. <https://doi.org/10.1016/j.ijhcs.2006.04.010>
- Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9), 2086-2095. <https://doi.org/10.1016/j.jss.2012.04.005>
- Marçal, A. S. C., de Freitas, B. C. C., Soares, F. S. F., Furtado, M. E. S., Maciel, T. M., & Belchior, A. D. (2008). Blending scrum practices and CMMI project management process areas. *Innovations Systems and Software Engineering*, 4(1), 17-29. <https://doi.org/10.1007/s11334-007-0040-1>
- McLeod, L., Doolin, B., & MacDonell, S. G. (2012). A perspective-based understanding of project success. *Project Management Journal*, 43(5), 68-86. <https://doi.org/10.1002/pmj.21290>
- Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software. *Journal of Systems and Software*, 82(11), 1869–1890. <https://doi.org/10.1016/j.jss.2009.05.052>
- 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. <https://doi.org/10.1007/s10664-008-9062-z>
- Nawrocki, J., Walter, B., & Wojciechowski, A. (2001). Toward maturity model for extreme programming. *Proceedings of the 27th EUROMICRO Conference, 2001* (pp. 233-239). Warsaw, Poland: IEEE Press. <https://doi.org/10.1109/EURMIC.2001.952459>
- Nunnally, J. C. (1978). *Psychometric theory* (2nd ed.). McGraw-Hill.
- Papadopoulos, G. (2015). Moving from traditional to agile software development methodologies also on large, distributed projects. *Procedia – Social and Behavioral Sciences*, 175, 455-463. Madrid, Spain: Elsevier. <https://doi.org/10.1016/j.sbspro.2015.01.1223>
- 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. https://www.academia.edu/18695406/Agile_maturity_model_AMM_A_Software_Process_Improvement_framework_for_agile_software_development_practices
- Patel, C., & Ramachandran, M. (2009b). Story card based agile Software development. *International Journal of Hybrid Information Technology*, 2(2), 125-140. <https://pdfs.semanticscholar.org/febf/71395e436087b449723fa567d121fcc9600.pdf>

- Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model, version 1.1. *IEEE Software*, 10(4), 18-27. <https://doi.org/10.1109/52.219617>
- 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. <https://doi.org/10.1037/0021-9010.88.5.879>
- Potter, N., & Sakry, M. (2009). Implementing scrum (agile) and CMMI together. *The Process Group-Post Newsletter*, 16(2), 1-6. <http://www.processgroup.com/pgpostmar09.pdf>
- Prabhakar, G. P. (2008). What is project success: A literature review. *International Journal of Business Management*, 3(9), 3-10. <https://doi.org/10.5539/ijbm.v3n9p3>
- The Process Group (2015). *Maturity levels 2 & 3 goals and practices*. <http://www.processgroup.com/condensed-cmmi1p3-dev-v1.pdf>
- Qumer, A., & Henderson-Sellers, B. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11), 1899-1919. <https://doi.org/10.1016/j.jss.2007.12.806>
- 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. <https://doi.org/10.1080/02602930802618344>
- Sanchez, J. C., Williams, L., & Maximilien, E. (2007). On the sustained use of a test-driven development practice at IBM. *Proceedings of 2007 Agile Conference (AGILE 2007)* (pp. 5-14). Washington, DC: IEEE. <https://doi.org/10.1109/AGILE.2007.43>
- Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research methods for students* (5th ed.). Pearson Education Limited.
- Sauter, V. (2006). *Extreme programming*. University of Missouri-St. Louis. http://www.umsl.edu/~sauterv/analysis/f06Papers/Hutagalung/#xp_practices
- Serrador, P., & Pinto, J. K. (2015). Does agile work? – A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.jiproman.2015.01.006>
- Shrum, S., & Phillips, M. (2004). *CMMI® overview for executives*. Carnegie Mellon University, Software Engineering Institute. <https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2004/cmmi/CMMIT1WedAM/1295SandyShrum.pdf>
- Sijtsma, K. (2009). On the use, the misuse, and the very limited usefulness of Cronbach's alpha. *Psychometrika*, 74(1), 107. <https://doi.org/10.1007/s11336-008-9101-0>
- Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Vasconcelos, A. P. L., Kamei, F. K., & de Lemos Meira, S. R. (2015). Using CMMI together with agile software development: A systematic review. *Information and Software Technology*, 58, 20-43. <https://doi.org/10.1016/j.infsof.2014.09.012>
- Software Engineering Institute (SEI) (CMMI Product Team) (2010). *CMMI® for Development, version 1.3*. Technical Report (CMU/SEI-2010-TR-033). Carnegie Mellon University, Software Engineering Institute. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf
- Stettina, C. J., & Hörz, J. (2015). Agile portfolio management: An empirical perspective on the practice in use. *International Journal of Project Management*, 33(1), 140-152. <https://doi.org/10.1016/j.jiproman.2014.03.008>
- Stutely, M. (2003). *Numbers guide: The essentials of business numeracy*. 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 (HICSS 2008)* (pp. 1-9). Waikoloa, HI: IEEE Press. <https://doi.org/10.1109/HICSS.2008.384>

- 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, 257-267. <https://doi.org/10.1016/j.sbspro.2014.03.030>
- Tanner, M., & von Willingh, U. (2014). Factors leading to the success and failure of agile project implemented in traditionally waterfall environments. In *Proceedings of the 2014 Management, Knowledge, and Learning International Conference – Human capital without borders: Knowledge and learning for the quality of life*. (pp. 693-701). Portoroz, Slovenia. <https://www.academia.edu/download/38840758/ML14-618.pdf>
- Tarnowski, M. (2014). *CMMI for development – CMMI-DEV*. Plays-In-Business.com. <http://www.plays-in-business.com/cmmi-for-development-cmmi-dev/>
- Thomas, G., & Fernández, W. (2008). Success in IT projects: A matter of definition? *International Journal of Project Management*, 26(7), 733–742. <https://doi.org/10.1016/j.ijproman.2008.06.003>
- Tongco, M. D. C. (2007). Purposive sampling as a tool for informant selection. *Ethnobotany Research and Applications*, 5, 147-158. <https://doi.org/10.17348/era.5.0.147-158>
- Turner, R. (2014). *Gower handbook of project management*. Gower Publishing Ltd.
- Verheyen, G. (2014). *Measuring success, measuring value*. <https://www.scrum.org/Blog/ArtMID/1765/ArticleID/11/Measuring-Success-Measuring-Value>
- VersionOne (2016). *10th annual state of agile report*. <https://www.versionone.com/?s=state+of+agile>
- VersionOne (2017). *11th annual state of agile report*. <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(1), 58-70. <https://doi.org/10.1016/j.infsof.2010.08.004>
- Yin, A., da Silva, M. M., & Figueiredo, S. (2011). Scrum maturity model. *Proceedings of the ICSEA*, (pp. 20-29)

APPENDIX

DEMOGRAPHIC INFORMATION

Specify the industry of the project	<i>Other: Please specify</i>					
Specify the agile method being used for the project	<i>Other: Please specify</i>					
Approximately how long has the project been using agile?	<Dropdown of predefined selection options, including "Other" option>		<Dropdown of predefined selection options, including "Other" option>		<Dropdown of predefined selection options, including "Other" option>	
	Less than 1 year	Between 1 and 2 years	Between 2 and 3 years	Between 3 and 4 years	Between 4 and 5 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	More than 6 years
What is your current job title?						
What role/roles do you currently fulfil within the agile team?						

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?					

BIOGRAPHIES



Vaughan Henriques obtained a B Eng (Hons) degree in Electronics Engineering from Canterbury University in 1996 and subsequently a B Com (Hons) degree from the University of Cape Town specializing in Information Systems in 2003 and is presently pursuing a Master's degree in Information Systems at the University of Cape Town. Having 19 years of industry experience spanning various sectors ranging from mining, media, telecoms and financial services, in various roles including software developer, development manager and architecture, his research interests include team motivation, software development methods, IT architecture and design.



Assoc Professor Maureen Tanner has been teaching systems analysis and design at the Department of Information Systems of the University of Cape Town since 2009. Her research interests lie in Agile software development related issues (for both collocated and distributed teams), UML, software engineering and social aspects of social engineering, global software development, virtual teams, and team collaboration.