Revisiting the Project Business Case in the Age of Agile [1]

Martin J. Cohen

Abstract

Information systems and new product development projects continue to be challenged in their ability to deliver value and meet time and budgetary constraints, in spite of the fact that Agile development methodologies have increasingly been applied to both. Agile methods de-emphasize the use of up-front documentation in general and comprehensive requirements specifications in particular. This research demonstrates how robust business case content could contribute to Agile project governance and project success.

Contribution to Practice

Projects are the universal mechanism for strategy implementation. This research explains the role of project business case content in project design, selection and governance. Without robust business case content, there can be no organizational accountability or continuity of purpose, especially in Agile development environments.

Business case research is still in the formative stages. While the IS value realization literature refers to content that should be part of an IS project business case, there is no universally accepted practical or theoretical definition for a project business case. Moreover, there is no theoretical or empirical research establishing the importance of specific business case content in Agile or Waterfall projects. This research applies Goal Setting Theory, requirements engineering and Stage Gate concepts to develop a theoretical framework for business case quality. This framework will be validated in future research using existing project case studies to document organizational practices in business case formulation and their influence on software quality achievement and value realization.

Keywords: business case, requirements engineering, project governance, project value realization, agile methods

[1] How to cite this paper: Cohen, M.J. (2022). Revisiting the Project Business Case in the Age of Agile; *PM World Journal*, Vol. XI, Issue IV, April.

1. Introduction

1.1 Structure of this paper

The research questions will be addressed by reviewing the historical motivation for Agile and the principles on which Agile is based. Next, we will examine the strengths and weaknesses of Agile, which have led some organizations to combine Agile methodologies with Stage Gate controls. The status of business case research is presented, focusing on the reasons business cases have not been widely or effectively employed, with particular emphasis on the challenges of writing business cases for information systems projects. A conceptual model for business case creation and a theoretical framework for business case quality are presented. These models are derived from the requirements engineering literature and Goal Setting Theory. Finally, a set of hypotheses are presented for our next phase of business case research.

1.2 Research Questions

- Why is project business case quality especially relevant in Agile projects?
- What is the appropriate role for the project business case in Agile development projects employing Stage Gate controls?
- What business case content is required to maximize software quality in Agile development environments?

2. Literature Search

2.1 Persistence of IS Project Failure

"As the Cheshire cat said to Alice in Wonderland, if you don't know where you are going, then any road will do." (Dodgson, 1865)

"Weakness in the business case is like a chronic disease in the project, leading to problems such as breakdown of management control, inability to monitor progress or expenditure, failure to manage resistance, project collapse, or worst of all, completion of the project without delivery of benefits." (Gambles, 2009)

Information systems expenditures represent about half of U.S. capital investment IS investments and make up the largest area of capital growth area in most industries

(Borenstein & Betancourt, 2005). While failed IS projects can disable a business strategy, successful IS projects can not only enable a business strategy (Davenport, 1989) but create a sustained competitive advantage (Porter, 1996).

Information systems projects have consistently fallen short of expectations (Standish, 2009) even when they have followed standard practices. The following chronology (Table 1.0) documents these failures, which have occurred even as Agile methodologies have gained widespread adoption.

Chronology of information systems project challenges

Table 1.0

(pre-Agile introduction)
The IS project failure in the 1980's and 1990's was consistently recorded over 50% with the largest projects, the most challenged. (Wright, 2011)
A study in Britain's public sector found 20% of IS projects to be total write-offs, with an additional 30% to 40% producing marginal results. (Wilcocks, 1994)
(post -Agile introduction)
83% of corporate projects were deemed unsuccessful, with 33% cancelled. Of those not cancelled, cost and schedule were almost double the budgeted amounts. (Georgiadou, 2003)
$150 billion has been lost annually on failed system and communication technology projects in the US- covering both public and private sectors - with $140 Billion lost in the EU. (Dalcher & Audley, 2003)
84% of British public sector projects were perceived as failures by the Royal Engineering Academy and British Computer Society (2004)
Shenhar (2008) found over 60% of IS projects failing to meet schedule or budgetary objectives, and frequently failing to meet business objectives.
According to the Chaos Report Summary (2009) only 32% of IS projects were delivered within specification and within schedule and budget. 24% of projects were cancelled during implementation; the remaining 44% eventually becoming operational, but with less functionality than originally intended.
Ambler (2013) reported that 62% of agile software projects succeeded, 3% failed, and 35% faced challenges.
The Standish Group (2015) found 16.2% of software projects successful, i.e. coming in on time and budget. The rest were either challenged or impaired. The Standish Group reported that 57.7% faced challenges and 31.1% failed.

2.2 Motivation for Agile Methodology

The majority of software development today still uses the Waterfall methodology, "which uses feedback loops between successive stages of development". (Misra, 2012) Prototyping is incorporated as an independent stage with comprehensive analysis and design, before any actual coding. The Waterfall method addressed many of the limitations of previous methodologies.

When Agile emerged, "it was seen as the solution to many of the problems that Waterfall could not address" (Misra, 2012). As Reagan (2012) explained, "it's hard to alter course when you're being swept down a large waterfall. Too much up-front planning means too much change management downstream."

In 2001, 17 software professionals who had been monitoring 'lightweight' software design met in Snowbird, Utah, to create a common philosophy. They developed the Agile Manifesto, which explains the principles of the Agile philosophy: (Misra, 2012)

2.3 Principles behind the Agile Manifesto

- Satisfy customers with prompt delivery of useful software
- Welcome requirement changes, even if they occur late in the process.
- Deliver functioning software in short cycles.
- Users and developers working together for the project duration.
- Build projects around motivated people.
- Provide development teams the conditions needed for success.
- Information is best conveyed through in-person discussion.
- Functioning software is the ultimate measure of progress.
- Team members should work at a steady pace.
- Agility is achieved by technical excellence and good design.
- Simplicity is achieved by focusing on the essential.
- The best results emerge from self-organized teams.
- The team decides how to be most effective, and adjusts accordingly.

Agile is designed to allow developers to create working code through continuous customer interaction. Once a project is approved and preliminary requirements laid out, Agile focuses solely on execution. Agile development consists of multiple short cycles, or

sprints, each sprint being performed by a designated group of users and developers. The intended result of a sprint is executable code. The objective is to have a release with each iteration. “A sprint typically lasts two to four weeks, with multiple iterations required to bring a product to fruition. Requirements, which may not be totally known at the start, are revealed and validated by iteration, and requirements thought to be important, but turn out not to be, are eliminated.” (Cooper, 2016)

Table 2.0 (below) explains how and why Agile methodologies can improve developmental efficiency and effectiveness.

Why Agile methodologies can accelerate product development
(Denning, 2016, Ahimbisibwe, 2015)

Table 2.0

Direct client/developer interaction increases personal motivation and engagement.
Team members report to the client; work goals are defined through user stories.
Sponsoring managers systematically remove impediments to progress, so that the team can evaluate and adapt their own performance to ensure continuous improvement
Rapid prototyping through frequent sprints yields executable code throughout development process.
Validation and testing of output in short releases promotes team member communication and cohesion.
Ability to update requirements throughout development avoids costly rework and rejected deliverables.

2.4 Impact of Agile practices on software quality and project success

Reports on Agile practices from software practitioners have been anecdotal or experiential; empirical studies are scarce. (Misra, et al, 2012) Reports of agile success have been confined to specific cases of Agile implementation. (Cockburn & Williams, 2000, Parrish, et al, 2004)

2.5 Rationale for combining Agile and Stage Gate methodologies

The value of Stage-Gate methodology has been well established: the discipline, staging structure, and clear benchmarks are designed to cull-out fatally flawed projects. (Cooper, 2016)

The Stage Gate process is divided into multiple stages, with deliverables generated from each. Between stages there are quality control checkpoints known as gates. Each gate has criteria to be fulfilled in order to obtain a go decision. Each gate also has a defined output which is the plan of action for the next stage. Gates are managed by gatekeepers, who understand the project and have the authority to commit resources. (Bhatia, et al, 2017)

Leading manufacturing organizations are starting to combine elements of Agile development with their Stage Gating processes for physical products. The trend started in software development, where Agile and Stage-Gate were seen as complimentary. The hybrid model is perceived to accelerate product releases, improve responsiveness to market demands, and facilitate communication among project participants (Cooper, 2016, Bhatia, 2017).

This research suggests that a robust business case can play a critical role in effective Agile/ Stage Gate collaboration by providing Stage Gate the benchmarks and metrics needed to enforce project management controls and identify projects at risk. It is posited that the Stage Gate process cannot be an effective vehicle of Agile project governance without the content provided by a robust business case.

2.6 Agile Project Governance and Value Creation

“The concept of governance is defined by Müller (2009) as the value system, responsibilities, processes and policies that provide organizations a framework for decision-making and managerial action in order to serve the best interests of internal and external stakeholders.” Governance, specifically corporate governance has been applied to industry, politics, economics and education (Lappi, 2017).

Müller (2009) believes project governance is necessary to achieve the best use of resources and identify projects at risk. Agile methodologies provide abundant opportunity to redirect projects that are not delivering promised benefits (Luna, 2014). Projects can be abandoned, scope-enhanced, scope-reduced, or redirected in a way that will make value realization possible, assuming the original project objectives were documented in a robust business case.

2.7 Status of Business Case Research

In many organizations, a business case is seen as a useful mechanism for justifying IS investments. “Business case research, however, is scattered among various literatures, and a clear, holistic definition of what actually constitutes a business case is still missing”. (Maes, 2013). In the view of these authors, “this field of research is in its infancy”. Despite the widely perceived value of business cases, empirical studies of their authorship, usefulness and application are rare. (Nielsen, 2017, Berghout, 2013, Maes, 2014)

“There is no commonly accepted practical or theoretical definition for an IS project business case” or for business case quality (Maes, 2014). “There is no accepted definition of the term ‘business case’ for research purposes.” (Kopmann, 2015). This research will provide a theoretical framework for measuring and defining business case quality.

> While many organizations believe in the importance of an information systems business case document and routinely create one (Ward, 2008), there is no theoretical or empirical research establishing the importance of specific business case content (Maes, 2014) or the motivation for including it.

One reason for perceived deficiencies in business case quality may be that the information technology and project management literatures have presented requirements for business case quality that are necessary, but may not be sufficient for IS project success. (PMBOK, 2013, Lester, 2007, Bradley, 2006, Havelka, 2002)

This research does not assume that organizations establish or enforce standards for business case quality. From a Goal Setting Theory perspective, it is illogical for an organization to initiate a project with unclear goals, uncertain feasibility, and weak rationale. (Lee, 1990, Mento, 1987, Latham, 1986) The information technology value realization literature recognizes that substantial IS investments should be justified by a high-quality business case. (Gambles, 2009, Havelka, 2002, Thomas, 2008) While research shows that this is often not the case in real life, the literature does not shed light on *why* this is so. (Ward, 2008, Jones, 2017, Farbey, 1992, Bacon, 1992)

What is notably absent in virtually all of the published business case definitions (Berghout, 2013, Maes, 2014, 2017) is a recognition of the need for detailed system requirements as a precursor to project assessment. The inadequacy of functional specifications is cited in the IS valuation literature as a primary cause of IS project failure. (Stefanou, 2001)

Post (1992) was one of the first to perceive the need for additional business case research to achieve a better understanding of their influence. There has been little response. “The fragmentation of knowledge on the subject and the lack of a clear definition may stimulate misunderstanding and discourage further research” (Maes, 2013), which this research can help address.

2.8 Ways in which business cases are employed

PMBOK (2013) defines the business case as “a documented economic feasibility study used to establish validity of the benefits of a selected component lacking sufficient definition and that is used as a basis for the authorization of further project management activities.” (p. 530) The business case describes the necessary information from a business standpoint to determine whether or not the project is worth the required investment. Typically, the business need (or rationale) and the cost-benefit analysis are contained in the business case to justify and establish boundaries for the project.” (p. 69) Lester (2007) characterizes business case content by posing these questions:

- “Why is the project required, and what are we trying to achieve?” (justification)
- “What are the deliverables, timeline and the anticipated costs?” (conceptualization)
- “What are the performance and success criteria?” (justification)
- “What are the risks?” (feasibility assessment)

Bradley (2006) views the project business case as a “living document created at the program level to steer program activity towards the eventual achievement of the project vision”. In Bradley’s view, the project business case should include the following:

- Rationale for the proposal, and how it supports the organization’s mission (conceptualization)
- The benefits of the investment, including assumptions and risks (justification)
- The intended future business state (conceptualization)
- The financial assessment (justification)
- Sections on project organization and governance (feasibility)

Kopmann, et al (2015) describe a business case as performing three roles: “describing why and how a project can be beneficial for an organization, providing the underlying

rationale for an organization to invest in a project, and setting the general conditions for the project scope". Hsaio (2008) views the business cases as "a structured overview of specific elements of an investment, such as the objectives, costs and benefits." Krell and Matook (2009) define a "business case as a formal document that summarizes the costs, benefits and impact of an investment". Benaroch, et al (p.835) (2006) define the business case as a "comprehensive document that includes the project description, resources required, benefits and financial plans, and a risk management plan." It provides the necessary information to define solutions for achieving investment objectives. LeFave (2008) believes investments can be prioritized through a business case to identify high-value deliverables.

Most researchers believe that a business case must include the project mission, anticipated costs, benefits and uncertainties. Krell and Matook (2009), Franken et al (2009) and Reimus (1997) found that individual accountability is essential for maintaining commitment. Gregor, et al (2006) see the benefit in determining qualitative and intangible project values in addition to economic factors.

Van Putten and Schief (2012) showed how business models, as a (description or) implementation of a company's business strategy can be aligned with business cases, as a conceptualization of an organization's processes. They believe that the business case can/should be used to close the gap between strategy and operations. In their systematic review of the business case literature, Maes, et al (2013) identified proof of concept, prototyping and viability assessment as essential parts of the business case development process.

While some authors believe a robust business case is needed primarily to support the investment decision, other authors believe the business case should support planning and monitoring through the entire project lifecycle. (Ward, et al, 2008) These authors believe that the business case should become a living document through an interactive process. (Franken, et al, 2009, Gregor, et al, 2006) Maes, Van Grembergen, De Haes (2013) propose that benefit expectations, costs and risks should continue to be formalized during and even after project implementation. A recent study, however, found that only one third of organizations used their IT project business case for the entire project duration. Another third created it solely to gain project approval, and the final third never produced one (Einhorn, 2019).

Khajavinia (2009) believes that a business case document should reference (or include) current process descriptions, because they "form the foundation for additional project work". Khajavinia also believes that "software engineers and business decision makers live in separate worlds, using their own terminology and decision criteria. Software engineers do not seem to understand nor are they involved in organizational value creation objectives. Building a business case, he believes, is one way to bridge the gap between business and software engineering and to increase the quality of software development."

2.9 Challenges in writing business cases for IS projects

The project management literature assumes that a project would not be initiated without some form of justification. (PMBOK, 2013) Although a physical document is procedurally required by many organizations, there is survey evidence that business case quality varies greatly, and in some cases, no justification is written. (Ward, 2008)

Oliver and Romm (2002) demonstrated that many of the largest IS projects are approved without any formal justification. Moreover, the level of strategic justification of IS technology is generally not proportionate to the financial expenditure or risk involved. Powell (1992) and Symons (1990) reported that formal evaluation methods take on a diminished role as the strategic importance of an IS project increases, leading directly to low business case quality.

One of the assumptions of this research is that business case quality is important to project success, because it motivates the appropriate project team competencies and performance goals. In their case study research, Berghout and Tan (2013) saw evidence that the strength of the business case led to more precise cost estimation and appropriate investment decisions. Norris (1996) reported that obscure statements of IS project motivation without assignment of responsibility for value delivery, was a primary cause of project failure.

Organizations maintained that a complete business case was crucial to obtaining value from IS investments in research conducted by AMR, ASUG and SAP. (Swanton, 2010) Two out of three European organizations believed that a business case is key to obtaining value out of IS investments, and 96 percent of organizations surveyed were required to produce some sort of document. (Ward, 2008)

One reason for low business case quality is that the financial justification of IS projects can be more complex than other project types, given the challenges of predicting and quantifying the costs and benefits. As a result, the overwhelming majority of IS project justifications do not employ financial analysis. (Bacon, 1992) Non-financial costs and benefits are even harder to estimate in IS projects because of the difficulty of linking cause and effect (Borenstein, 2005). As a result, these steps are often ignored.

Another reason for low business case quality might be the inadequacy of organizational metrics. To the extent that organizations utilize financial and non-financial metrics for process and business unit performance, these same metrics can and should be applied to project proposals. Unfortunately, comprehensive organizational metrics are more the exception than the rule (Bloom, 2012)

Conceptualizing an IS application can be more time and resource consuming than building the application itself. (Stefanou, 2001, Hecht, 1999) This is partially the result of high-level programming languages, which makes it possible to generate code more efficiently, but not necessarily with more forethought. As a result, this process may not be given the time and attention it deserves, and may result in business cases of low quality. Interviews with 130 Fortune 1000 CIO's revealed that 41% of companies do not have central oversight of their IS budget; 46% do not fully document their applications and infrastructure; 47% do not centrally track projects; 57% do not have project success criteria; and 68% do not track project benefits.(Jeffrey, 2004) Clearly there is a gap in the ability of organizations to practice sound IS portfolio management, which this research can help address. The absence of sound IS portfolio management could create organizational conditions leading to low business case quality.

If an IS project business case does not include a valid justification, does not conceptualize the project deliverables in detail, and does not set forth a viable implementation methodology, i.e., it is of low quality, it is posited that all of the known causes of project failure may be set into motion. Pinto and Slevin (1987) demonstrate the relationship between strategy and tactics in their 'Strategy Tactics Effectiveness Matrix'. "Organizations that employ strong implementation methodologies without strategic foundation are likely to implement the wrong solution and/or solve the wrong problem. While the customer might initially accept the deliverables, they are likely to be misapplied or misused".

Table 3 (below) explains how specific business case deficiencies can prevent full project value realization.

Impact of business case deficiencies on project value realization

Table 3.0

Business case deficiency	Organizational impact
No practical or theoretical business case definition (Kopmann, 2015)	Inability to establish guidelines for business case research or professional standards for use
Lack of theoretical or empirical research establishing the importance of specific business case content (Maes, 2013)]	Inability to apply industry/ organization-wide content requirements
Inadequate requirements for project assessment (Maes, 2013, Berghout, 2013, Maes, 2017)	Naïve, superficial evaluation of project proposals; divergent project goals among stakeholders
Inadequate functional specifications (Stefanou, 2001, Hecht, 1999)	Primary cause of IS project failure due to diminished product usability
Dearth of financial analysis (Bacon, 1992)	Suboptimal, non-economic use of organizational resources
Vague statements of IS project benefits (Norris, 1996)	Uncertain allocation of responsibility for managing project benefit delivery
Unrealistic/unarticulated project goals, badly defined requirements, unmanaged risks, use of immature technology (Charette, 2005, Wallace, 2004a)	Primary causes of IS project failure.

2.10 Requirements engineering defined

Business Requirements define the boundaries of a project determining what is inside or outside of scope. (Rahmesh, 2011) ***Requirements engineering*** is defined in the literature as an "iterative process of eliciting, structuring, specifying, analyzing and managing software requirements." (Sommerville, 2011) "The requirements engineering process is a structured set of activities which are carried out to derive, validate and maintain requirements documents" (Raja, 2009) According to Weinberg (2011) "requirements are made for a common purpose; to convert vague desires into explicit and unambiguous statements of what customers want".

After requirements elicitation, structuring and specification, the last phase in RE is validation. “Requirements validation identifies ambiguous requirements and resolves conflicts.” (Raja, 2009) “The requirements validation process ensures that the requirements in the software requirements specification (SRS) are consistent and complete, before it is used as a basis for further system development.” (Nguyen, 2016)
“IEEE Computer Society (2012) defines ***verification*** as the process of providing objective evidence that the system… and its associated products conform to requirements (e.g., for correctness, completeness, consistency, and accuracy)”.

“IEEE Computer Society (2012) defines ***validation*** as the process of providing evidence that the system…. solves the right problem (e.g., correctly model physical laws, implement business rules, and use the proper assumptions), and satisfy intended use and user needs”. (Ryan, 2017)

The efficiency and effectiveness of a requirements engineering process can be characterized by the level of scrap, creep, and churn created. (Rahmesh, 2011)

> ***Scrap***: Requirements are dropped due to budget cuts or business/legal constraints.
> ***Creep***: Project scope takes on additional features or functionality.
> ***Churn***: Requirements fluctuate throughout the project.

A key assumption of this research (Figure 1.0) is that comprehensive requirements elicitation is an essential first step to producing a high-quality business case. When requirements and requirements engineering processes are deficient, an unsuccessful or prematurely terminated project is typically the result (Standish, 1999). There is clear evidence that problems in requirements engineering negatively impact project outcomes (Somerville, 1996) in spite of the modest effort needed for a successful RE process (Niazi, 2008). Alexander and Stevens (2002) suggest that only 5% resources are needed to support the requirements engineering effort. This might be 25% of project duration (or 3 months, depending on project size).

Requirements elicitation can take many forms. Findings show that the most used individual techniques are: interviews, scenarios, and questionnaires. Among group techniques, workshops stand out (Pacheco, 2018).

Given that each elicitation method has advantages, it is hypothesized that projects using multiple elicitation techniques will compensate for the limitations of each. It is also posited

that projects using multiple elicitation techniques will encounter fewer problems in verification and validation.

While a complete software requirement specification (SRS) document would not normally be part of a project business case, the more specifications included, it is hypothesized, the more useful the business case will be in guiding the organization towards software quality and project success.

2.11 Requirements engineering in Agile environments

A key assumption of this research is that business case quality is equally important in both Waterfall and Agile environments. In Agile, requirements emerge during the development process. “There are no conceptual models forming the basis of acquired domain knowledge. Rather, knowledge is documented in the form of code, test cases, and programmers’ collective memory” (Rubin, 2011). These authors believe that continually updated conceptual models would benefit communication between stakeholders, essential to the Agile process.

The fundamental hypothesis of Waterfall is that systems can be fully designed, planned and specified before any code is written. The fundamental hypothesis of Agile is that the best software emerges from small teams, continuous improvement, and timely feed-back. (Rubin, 2011) Agile development, however, requires a “clear and complete definition of the entire system before it can be divided and incrementally built” (Stoica, 2013). Maiden & Jones (2012) believe that “Agile requirements techniques support some activities at the expense of others. They believe there is little support in Agile for discovering and analyzing business goals or developing innovative ideas due to the constraints of short-term sprints”.

Cao and Ramesh (2008) and Grundy (2013) reported that Agile projects often select inappropriate architectures, refactor code, and fail to confirm modifications with key stakeholders prior to implementation. Agile requirements engineering is an iterative, incremental approach that does not help in defining project budget or schedule (Matharu, 2015) due to the absence of a-priori requirements planning (Inayat, 2015).

Customers define functional requirements throughout the Agile project life cycle. According to Cardinal (2013) neglecting nonfunctional requirements (scalability, capacity, availability, reliability, recoverability, maintainability, serviceability) may lead to major

rework throughout the project, because nonfunctional requirements are not included in the agile mini project definition phase. Farid and Mitropoulos (2013) claim that Agile project management does not adequately represent activities that deal with nonfunctional requirements. Aljallabi and Mansour (2015) acknowledged that neglecting nonfunctional requirements is one of Agile's limitations.

2.12 Requirements prioritization in Agile

"Most Agile organizations use simple techniques such as user stories to define high-level requirements. These short, abstract descriptions serve mainly as anchors for further discussions with customers." (Cao, 2008)

Agile implements "the highest- priority functionality first, so that customers can realize immediate business value". Organizations establish new priorities as team member "understanding of the project evolves", adding requirements as needed. In Waterfall projects, requirements are prioritized only once, with business value, risk, cost and interdependencies all factored in. In Agile, prioritization is driven primarily by business value as the customer perceives it. (Cao, 2008)

3. Requirements for business case quality

Limitations in Agile methodologies that can be mitigated by application of robust business case and stage gate controls

Table 4.0

Agile documentation deficiencies	Resultant risks	Potential risk mitigation though robust business case and stage gate controls
Project definition begins during 'fuzzy front end', subsequent to project approval (Bhatia, 2017)	Unrealistic project scope, cost and budget	Organizational consensus on project scope, cost and budget prior to development activities
No comprehensive system requirements specifications (Cao, 2008)	Scope instability/ scope creep	Project mapping /directional guidance through clear project conceptualization
Limited documentation of non-functional system requirements (Farid, 2013, Elghariani, 2016)	Schedule delays and loss of product usability	Recognition and accomplishment of system performance expectations

Limited process documentation due to short deliverable timelines and direct user/coder interaction (Reagan, 2012)	Ambiguity of process and uncertain functionality of deliverables	Improved benefits realization and deliverable acceptance

Table 4.0 illustrates how business case content and stage gate controls can correct for deficiencies in Agile requirements engineering, prioritization and governance.

Conceptual Model for Business Case Creation

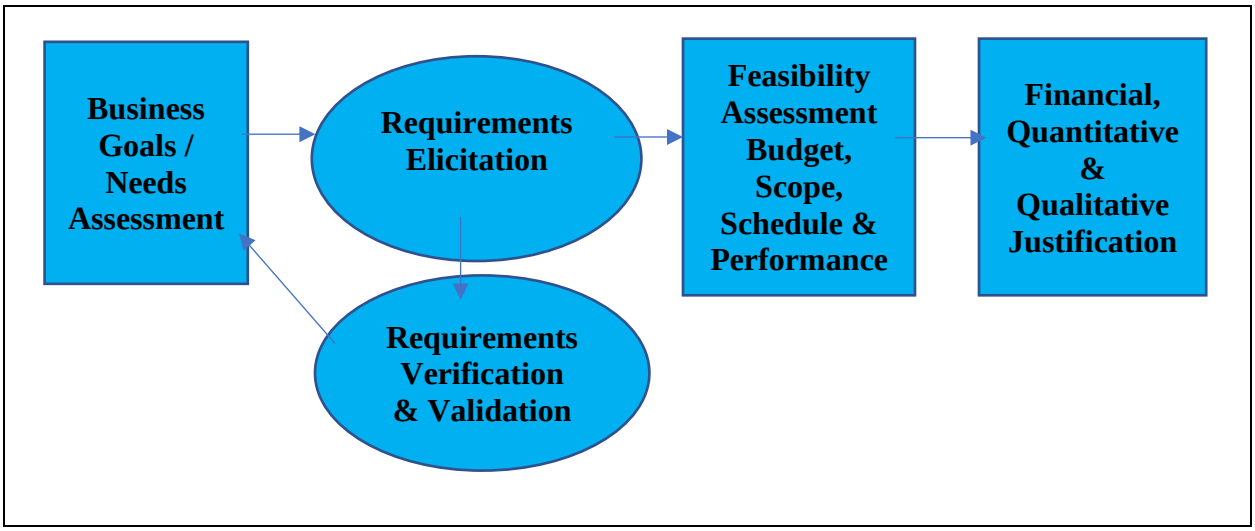

Figure 1.0

The underlying assumption behind the business case creation model (Fig 1.0) is that the requirements engineering process should begin before the project is submitted for approval. This collaboration between business analysts and software designers is necessary to write a high-quality business case, i.e. one that has the potential to achieve organizational objectives.

Applying requirements engineering methodologies to the elicitation of user needs is the first step towards formulating system requirements that accurately address organizational and project goals. Only after requirements have been elicited in sufficient detail can their feasibility be determined. Financial and qualitative justification should be based on verified, validated and feasible system requirements.

3.1 Business Case Quality Defined

The construct *Business Case Quality* is defined in this research (Table 5.0) as the extent to which a project has been conceptualized, justified, and demonstrated to be feasible This definition goes beyond the traditional PMBOK (2013) concept of project mission or vision, and is motivated by the attributes of effective goal setting.

Project Conceptualization is based on the ***clarity*** and ***specificity*** of project goals. It entails a clear description of the deliverables and functionalities that will be provided. The conceptualization should also define clear performance and success criteria for measuring project value delivery (Thomas, 2008), and should include:

- Project scope
- Software architecture
- Requirements description
- Requirements prioritization
- Implementation process definition
- Vendor/product/technology selection and acceptance criteria

Project Feasibility Assessment is based on ***goal difficulty*** and ***attainability*** It serves to confirm the economic, technological, and operational plausibility of the application [36] as well as the feasibility of completing the project deliverables via the prescribed methodology, and within the project's budgetary and schedule constraints, and should include the following:

- Scenario planning to determine impact of most likely outcomes
- Sensitivity Analysis and contingency planning for key project uncertainties
- Enumeration and validation of technical and business assumptions
- Description and evaluation of the risks and uncertainties in the process, product, or solution.
- Illustration and validation (proof of concept) of the project goals and deliverables. This may require modeling, testing, or prototyping deliverables to ensure customer acceptance.

A project feasibility assessment "considers whether the proposed solution is operationally, technologically, or economically viable and likely to occur in a timely manner" (Montague, 2016, p. 15). A feasibility study is conducted to ascertain an IS project's viability, and obtain data for planning (McConnell, 1998). In a feasibility study, performance is compared against pre-determined technical, operational and financial benchmarks (Pressman, 2004). This allows for project cancellation at a lower cost (stage gate process), or, if the project is allowed to proceed, assists in the re-estimation of resource requirements. Despite the recognition afforded feasibility studies, the topic is large ignored in popular software engineering textbooks (Greer, 2009).

Project Justification is based on the ***value*** or ***relevance*** of the project goals, and the connection of those goals to the recommended solution. It explains how and why the project will provide the organization a competitive, cost, or qualitative advantage. It should include a comparison of the recommended process or product with potential technology alternatives and/or solution providers, including do-nothing options. IS success dimensions include system, information and service quality, and the strategic, informational, and transactional value added. Mirani and Lederer (1998) point out that a single information system can provide all three types of benefits. IS projects are typically initiated based on one or more of the following organizational events:

- New regulatory requirement
- Opportunity to lower transaction costs
- Legacy system or platform replacement
- New product or business opportunity

Each of the three business case components, conceptualization, justification, and feasibility assessment (Table 5.0) are essential in themselves, and dependent on the other two. For example, one cannot justify or define an implementation methodology for an objective which has not been conceptualized. Similarly, one cannot validate a financial justification without a demonstrable and verifiable implementation methodology. One could have, for example, a well-defined (conceptualized) objective with many potential benefits (justification) for which no cost-effective implementation methodology exists. In other words, every set of benefits must be proportionate to the cost. Unfortunately, the true cost of implementation and the true cost/benefit ratio cannot be determined until the project has been conceptualized at a fairly detailed level, which, in the case of information systems projects, typically does not occur until well past project initiation. This is yet

another reason why a high-quality IS project business case is so important, yet so difficult to achieve.

3.3 Goal Setting Theory - theoretical foundation for business case quality

Goal Setting Theory (Locke, 1981) suggests that comprehensive, well-articulated, justifiable, and achievable project goals, i.e. higher quality business cases, would lead to better project outcomes. It is hypothesized, therefore, that the level of project value achievement can be anticipated by the 'quality of the goals' expressed in the IS project business case. Goal Setting Theory plays a central role in this research, because it provides the theoretical framework for what constitutes business case quality ***Goal clarity, relevance, and attainability*** correspond to the ***conceptualization, justification, and feasibility*** components proposed for business case quality (Table 5.0).

Goal Setting Theory addresses the importance of clear, relevant, and challenging goals in guiding and achieving high individual, group and organizational performance. This theory has been applied to tasks where individuals have control over their performance, as is the case in IS business case formulation. "Specific and difficult goals (vs. 'do best' goals) lead to a higher level of performance than vague, non-quantitative goals (Locke, 1990) The direct effect of goal specificity, divorced from goal difficulty or goal level on performance is to reduce performance variance. This is because goal specificity reduces interpretive leeway as to the exact meaning of the goal" (Locke, 1990)

Expectancy Theory (Vroom, 1964) maintains that irrelevant, unworthy, and unattainable goals will not motivate individuals toward valued rewards, or, it is hypothesized, lead to business cases of high quality. While hard goals may lead to greater persistence and effort than easy goals (Locke, 1990), unachievable goals will surely lead to project failure. Therefore, the harder the goals, i.e. the greater the degree of project novelty and complexity, the more important it would be to validate project feasibility. Clear goals motivate the execution of task driven strategies. (Latham, 1975) Challenging goals stimulate planning; challenging goals also increase the likelihood that strategy will be applied. (Earley, 1987, 1988)

Kernan and Lord (1990) reported that general (non-specific) goals led people to commit the same amount of resources to a problem regardless of prior degree of failure. Explicit goals, however, enable people to use feedback information intelligently. They commit fewer resources to courses of action that lead to extreme failure. Non-specific goals also

allow people "to give themselves the benefit of the doubt concerning the adequacy of their performance" (Locke, 1990).

Latham, Winters, and Locke (1994) found that effort applied to goal setting influences goal commitment and self-efficacy; participation in formulating task strategies influences task strategy quality; and self-efficacy and task strategy quality (in turn) influence performance. This would suggest that core project teams that participate in the formulation of project goals would be more likely to write a high-quality business case.

Kondrasuk (1981) reviewed 185 studies of goal-setting at the organizational or unit level. 91% of the 185 studies showed positive or contingently positive results. "The core findings of Goal Setting Theory are based on data from over 40,000 subjects in eight countries; 88 different tasks; laboratory and field settings; experimental and correlational designs; time spans ranging from one minute to three years; studies of assigned, self-set and group goals; and data from group, individual, and organizational levels of analysis." (Locke, 1990,p. 62)

Table 5.0 applies Goal Setting Theory directly to the content requirements needed for business case quality.

Theoretical Framework for Business Case Quality

Table 5.0

Goal Setting Theory Construct	**Goal Clarity**	**Goal Relevance**	**Goal Attainability**
Business Case Component	**Project Conceptualization**	**Project Justification**	**Project Feasibility Validation**
	Organizational goals	Quantitative and qualitative valuation of organizational goals	Achievability of organizational goals with successful project delivery
	Project goals	Relevance and value of project goals to organizational goals	Achievability of project goals within timeline and budget

Business Case Content Requirements			
	Value proposition of project to organization	Quantification and qualification of benefits	Costs, risks and obstacles to benefit delivery
	Specificity of project deliverables	Incremental return on investment for each deliverable	Likelihood of delivery within timeline and budget
	Prioritization of project deliverables	Criticality and valuation of deliverables	Costs, risks and obstacles to benefit delivery
	Proposed process documentation	Criticality of each proposed process component to project value delivery	Attainability of each proposed process component
	Essential system functionalities	Criticality of each functionality to project value delivery	Costs, risks and obstacles to benefit delivery

4. Conclusions

Agile software development projects would benefit from explicit and comprehensive business case content, perhaps even to a greater degree than Waterfall projects. Waterfall projects require detailed system requirements as a first implementation step, which can serve to address deficiencies in the original business case. Any documentation created in Agile projects is post-facto, and can, therefore, have no beneficial effect on project selection or design. Functionalities created in Agile projects may or may not be consistent with the original project intent. If the value proposition of a project is not clearly documented up front, and agreed to by critical stakeholders, scope creep and loss of focus can result.

The purpose of this paper is not to question the value or validity of Agile methods – quite to the contrary. The purpose of this paper is to identify those elements of business case content that could facilitate and strengthen the application of Agile methods. Without a well-defined business case, there is no basis for holding the sponsor or development team accountable for achieving project objectives, budget, or timetable, whatever the development methodology employed.

This research has provided a theoretical framework for business case composition. This framework can be used as a roadmap for project design, project selection, and project value realization, not only in information systems projects, but product development projects as well, which suffer from an even higher failure rate than information systems. Coupled with stage gate controls, this theoretical framework can create conditions that will facilitate Agile adoption and value realization.

5. Hypotheses to be tested in future research

1. Business cases with well-articulated project goals and success criteria will be associated with higher software quality.
2. Business cases with clearly defined project scope will be associated with higher software quality
3. Business cases with requirements developed through formal requirements prioritization will be associated with higher software quality.
4. Business cases based on a complete requirements feasibility assessment will be associated with higher software quality.
5. Business cases with clearly articulated financial, quantitative and qualitative justifications will be associated with higher software quality.
6. Business cases containing a conceptual model of the proposed solution will be associated with higher software quality.
7. Business cases created by organizations with well-documented internal processes will be associated with higher business case quality and higher software quality.
8. Requirements elicitation, verification, and validation activities that are utilized for business case development prior to project approval will have an incrementally greater effect on software quality beyond those activities that occur after project approval.
9. Requirements elicitation, verification and validation activities that take place after project approval but prior to project initiation will have an incrementally greater effect on software quality beyond those activities that occur after project initiation.
10. Business cases employing multiple requirements elicitation, verification and validation techniques will be associated with higher software quality than those business cases using a single technique.

REFERENCES

Alexander, I., & Stevens, R. (2002). Writing better requirements. Addison-Wesley.

Aljallabi, B. M., & Mansour, A. (2015). Enhancement approach for non-functional requirements analysis in agile environment. In Computing, control, networking, electronics and embedded systems engineering (ICCNEEE), 2015 international conference (pp. 428-433). Khartoum: IEEE

Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, *28*(1), 7–33. https://doi.org/10.1108/JEIM-08-2013-0060

Ambler, S. (2013). 2013 IT project success rates survey results. Retrieved from http://www.ambysoft.com/surveys/success2013.html Ambler, S., & Lines, K. (2012). Disciplined agile delivery: A practitioner

Bacon, C. J., (1992), The use of decision criteria in selecting information systems/technology investment, *MIS Quarterly*, 16(3) 355-50

Benaroch, M., Lichtenstein, Y. and Robinson, K. (2006) Real options in information technology risk management: an empirical validation of risk option relationships, *MIS Quarterly*, Vol. 30, No. 4, pp. 827-864

Berghout, E., & Tan, C. W. (2013). Understanding the impact of business cases on IT investment decisions: An analysis of municipal e-government projects. *Information and Management*. https://doi.org/10.1016/j.im.2013.07.010

Bhatia A, Cheng J, Salek S, Chokshi V and Jetter A (2017) Improving the effectiveness of fuzzy front end management: expanding stage-gate methodologies through agile, Proceedings of PICMET '17: Technology Management for Interconnected World

Bloom, N., Sadun, Raffaella, Van Reenan, J. (2012) How three essential practices can address even the most complex global problems, *Harvard Business Review*, November, pp. 77 – 82

Borenstein, D. and Betancourt, P. R. B., (2005), A multi-criteria model for the justification of IT investments, *INFOR*, V. 3, pp. 1-21

Bradley, Gerald, J. (2006), Benefit realization management: practical guide to achieving benefits through change, Gower Publishing Limited, pp. 201-201, 275

Cao, L. and Ramesh, B, (2008) “Agile Requirements Engineering Practices: An Empirical Study,” IEEE Software, vol. 25, no. 1, , pp. 60–67.

Cardinal, M. (2013). Executable specifications with scrum: A practical guide to agile requirements discovery. New York, NY: Addison-Wesley.

The Chaos Report Summary (2009), The Standish Group International, Boston, MA

Charette, R. N. (2005) Why software fails, *IEEE Spectrum*, September

Cockburn, A., and Williams, L. 2000. "The Costs and Benefits of Pair Programming." In Extreme Programming and Flexible Processes in Software Engineering. Boston, MA: Addison-Wesley, pp. 223–247.

Cooper, R. G. (2016). Agile-stage-gate hybrids. *Research Technology Management*, *59*(1), 21–29. https://doi.org/10.1080/08956308.2016.1117317

Dalcher, D. & Audley, G. (2003). Avoiding IS/IT implementation failure. Technology Analysis and Strategic Management, 15 (4), 403-7.

Davenport, T., (1989), The case of the soft software proposal, Harvard Business Review, May-June, pp. 12-24.

Denning, S. (2016). Understanding the three laws of Agile. *Strategy and Leadership*, *44*(6), 3–8. https://doi.org/10.1108/SL-09-2016-0074

Dodgson, C. L. (Lewis Carroll) (1865), *Alice's Adventures in Wonderland*, Macmillan

Earley, P.C. and Perry, B.C. (1987) Work plan availability and performance: An assessment of task strategy priming on subsequent task completion, *Organizational Behavior and Human Decision Processes*, 39, 279-302

Earley, P.C. Connolly, T. and Lee, C. (1988) Task strategy interventions in goal setting: The importance of search and strategy development, Department of Management and Policy, University of Arizona, unpublished manuscript

Einhorn F, Marnewick C and Meredith J (2019) Achieving strategic benefits from business IT projects: The critical importance of using the business case across the entire project lifetime, *International Journal of Project Management*, 37, 989-1002

Elghariani, K., & Kama, N. (2016). Review on Agile requirements engineering challenges. In *2016 3rd International Conference on Computer and Information Sciences, ICCOINS 2016 - Proceedings* (pp. 507–512). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICCOINS.2016.7783267

Farid, W. M., & Mitropoulos, F. J. (2013). Visualization and scheduling of non-functional requirements for agile processes. In Southeastcon, 2013 Proceedings of IEEE (pp. 1-8). Southeastcon: IEEE.

Farbey, B., Land, F. and Targett, D. (1992), Evaluating investments in IT, *Journal of Information Technology*, 7, 109 – 22

Franken, A., Edwards, C. and Lambert, R. (2009) Executing strategic change: understanding the critical management elements that lead to success, *California Management Review,* Vol. 51, No. 3, pp. 49-71

Gambles, I., (2009) Making the business case: proposals that succeed for projects that work, Ashgate Publ. Farnham, UK, (e-book),

Georgiadou E (2003). Software process and product improvement: a historical perspective. *Cybernetics and Systems Analysis*, 39 (1), 125 – 42

Greer, D., & Conradi, R. (2009). Software project initiation and planning – an empirical study. *IET Software*, *3*(5), 356. https://doi.org/10.1049/iet-sen.2008.0093

Gregor, S., Martin, M., Fernandez, W., Stern, S., & Vitale, M (2006) The transformational dimension in the realization of business value from information technology. *The Journal of Strategic Information Systems*, 15(3) 249-270

Havelka, D., Lee, S. (2002) Critical success factors for information requirements gathering, Information Strategy: The Executive's Journal, summer, pp. 36-46

Hecht, B (1999), Choose the right ERP software www.datamation.com

Hsiao R. (2008), Knowledge sharing in a global professional service firm, *MIS Quarterly Executive*, Vol. 7, No. 3, pp. 399-412

IEEE Computer Society (2012), IEEE-STD-1012- IEEE Standard for System and Software Verification and Validation, New York: IEEE Computer Society

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements, engineering practices, and challenges. Computers in Human Behavior, 51 , 915-929.

Jeffery M and Leliveld I (2004) Best practices in IT portfolio, *MIT Sloan Management Review*, Vol. 45, No. 32, pp. 41-49

Jones, K. (2017) Building the business case for HR initiatives, *Workforce Solutions Review*, January-March, pp. 35-36

Kernan, M.G. and Lord, R.G. (1990) The effects of explicit goals and specific feedback on escalation processes, *Journal of Applied Social Psychology*

Khajavinia, R. (2009.). *THE BASIS FOR BUILDING A BUSINESS CASE IN SOFTWARE DEVELOPMENT, A CASE STUDY*.

Kondrasuk, J. N. (1981) Studies in MBO effectiveness, *Academy of Management Review*, 6, 419-30

Kopmann, J. Kock, A., Killen, C. Gemunden, H.G. (2015) Business case control in project portfolios – an empirical investigation of performance consequences and moderating effects, *IEEE Transactions on Engineering Management*

Krell, K. & Matook, S., (2009) Competitive advantage from mandatory investments: an empirical study of Australian firms, *The Journal of Strategic Information Systems,* Vol. 18, No. 1, pp. 31-45

Lappi, T., & Aaltonen, K. (2017). Project governance in public sector agile software projects. *International Journal of Managing Projects in Business*, *10*(2), 263–294. https://doi.org/10.1108/IJMPB-04-2016-0031

Latham, G.P. and Baldes, J.J. (1975) The "practical significance" of Locke's theory of goal setting, *Journal of Applied Psychology*, 60, 122-124

Latham, G.P. and Lee, T.W. (1986), 'Goal setting'. In Locke, E. (Ed.) Generalizating from Laboratory to Field Settings, Lexington Books, Lexington, MA. pp. 101-117

Latham G, Winters D and Locke E (1994) Cognitive and motivational effects of participation: a mediator study, *Journal of Organizational Behavior*; Chichester Vol. 15, Issue 1:49

Lee, T.W., Locke, E.A. and Latham, G.P. (1990), 'Goal setting theory and job performance'

LeFave, R. Branch, B., Brown, C., & Wixom, B. (2008) How Sprint Nextel reconfigured IT resources for results, *MIS Quarterly Executive*, Vol. 7, No. 4, pp. 171-179

Lester, A., (2007) Project management, planning and control: managing engineering construction, and manufacturing projects to PMI, APM, and BSI standards, Amsterdam: Elsevier/Butterworth- Heinemann, e-book

Locke, E.A. and Latham, G. (1990), A theory of goal setting and task performance. Englewood Cliffs, NJ: Prentice Hall

Locke, E.A., Shaw, K.N., Saari, L.M. and Latham, G.P. (1981). 'Goal setting and task performance: 1969:1980', *Psychological Bulletin*, 90, 125-152

Luna, J.H.de O, A., Kruchten, P., E.Pedrosa, M. L. G. do, Almeida Neto, H. R. d., & Moura, H. P. d. M. (2014). State of the Art of Agile Governance: A Systematic Review. *International Journal of Computer Science and Information Technology*, *6*(5), 121–141. https://doi.org/10.5121/ijcsit.2014.6510

Maes, K., De Haes, S., & Grembergen, W. Van. (2017 A Business Case Process for IT-Enabled Investments: Its Perceived Effectiveness from a Practitioner Perspective. https://doi.org/10.4018/978-1-5225-0861-8.ch001

Maes, K., De Haes, S., & Grembergen, W. Van. (2013.). Using a Business Case Throughout an Investment: An Exploratory Case Study on a Business Case Process, (1).

Maes, K., Van Grembergen, W., & De Haes, S. (2014). Identifying Multiple Dimensions of a Business Case : A Systematic Literature Review. *The Electronic Journal Information Systems Evaluation*, *17*(1), 47–59.

Maiden, N., & Jones, S. (2012). *Agile Requirements—Can We Have Our Cake and Eat It Too?* Retrieved from www.jisc.ac.uk/whatwedo/

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical study of agile software development methodologies: A comparative analysis. ACM SIGSOFT, Software Engineering Notes, 40 (1), 1-6

McConnell, S. (1998) Feasibility studies, *IEEE Software*, Vol. 15, pp. 119-120

Mento, A.J., Steel, R.P. and Karren, R.J. (1987), A meta-analytic study of the effects of goal setting on task performance: 1966-1984', *Organizational Behavior and Human Decision Processes*, vol. 39, pp. 52-83

Mirani, R., Lederer, A., (1998) An instrument for assessing the organizational benefits of IS projects, *Decision Sciences*, Vol. 29, No. 4, pp. 803-838

Misra, S., Kumar, V., Kumar, U., Fantazy, K., & Akhter, M. (2012, October 5). Agile software development practices: Evolution, principles, and criticisms. *International Journal of Quality & Reliability Management*. https://doi.org/10.1108/02656711211272863

Montague, R. A. (2016) Developing successful business cases, *Industrial Management*, March/April 2016, pp. 10-16

Müller, R. (2009),Project Governance , Gower, Farnham

Nguyen, T. H., Grundy, J. C., & Almorsy, M. (2016). Ontology-based automated support for goal–use case model analysis. *Software Quality Journal*. https://doi.org/10.1007/s11219-015-9281-7

Niazi M, Cox K and Verner J (2008), A measurement framework for assessing the maturity of requirements engineering process, *Software Quality Journal*. https://doi.org/10.1007/s11219-007-9033-4

Nielsen, P. A., & Persson, J. S. (2017). Useful business cases: Value creation in IS projects. *European Journal of Information Systems*. https://doi.org/10.1057/s41303-016-0026-x

Norris, G.D. (1996) Post-investment appraisal, L. Willcocks (Ed.), *Investing in information systems evaluation and management*, Chapman & Hall, UK, pp. 193-223, Chapter 9

Oliver, D. and Romm, C. (2002), Justifying enterprise resource planning adoption, *Journal of Information Technology*, Vol 17, pp. 199-213

Pacheco, C., García, I., & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. https://doi.org/10.1049/iet-sen.2017.0144

Parrish, A.; Smith, R.; Hale, D., and Hale, J. 2004. "A Field Study of Developer Pairs: Productivity Impacts and Implications," Software, IEEE (21:5), pp. 76–79.

Pinto, J.K. and Slevin, D. P (1987), Balancing Strategy and Tactics in Project Implementation, *Sloan Mgmt. Review*, Fall, pp. 33-41

PMBOK, 5th edition, (2013), A guide to the project management book of knowledge, Project Management Institute, Newtown Square, PA.

Porter, M. (1996) What is Strategy? *Harvard Business Review*, Nov.-Dec., pp. 61 – 78

Post B (1992) A business case framework for group support technology, *Journal of Management Information Systems*, Vol. 9, No. 3, pp. 7-26

Powell, P. (1992), Information technology evaluation: is it different?, *Journal of the Operational Research Society,* 43 (1), 29-42

Pressman, R. (2004) *Software engineering, a practitioner's approach*, McGraw Hill

Rahmesh, S. P., & Madhavan, B. (2011.). *Analysis Scope Creep, scrap & Churn are NOT SINS in Requirements Engineering*. *Symbiosis Centre for Management and HRD* (Vol. 3).

Raja, U. A. (2009). Empirical studies of requirements validation techniques. In *2009 2nd International Conference on Computer, Control and Communication, IC4 2009*. https://doi.org/10.1109/IC4.2009.4909209

Reagan, B. 2012. Going Agile: CA Technologies, Clarity PPM Division's transformative journey. Digital Celerity, San Francisco, CA, September 22. http://www.slideshare.net/ DCsteve/going-agile-with-ca-clarity-ppm-agile-vision

Reimus, B. (1997) The IT system that couldn't deliver, *Harvard Business Review*, Vol. 75, No. 3, pp. 22-26

Royal Academy of Engineering and the British Computer Society (2004) "Report of a working group"

Rubin E and Rubin H, (2011) "Supporting agile software development through active documentation" *Requirements Engineering*, Vol. 16:117–132

Shenhar, A. (2008), "Unleashing the power of project management", Industrial Management , Vol. 50 No. 1, pp. 14-18.

Slevin, D. P., & Pinto, J. K. (1987). BALANCING STRATEGY AND TACTICS IN PROJECT IMPLEMENTATION. *Sloan Management Review*, *29*(1), 33–41.

Sommerville, I. (1996). Software engineering (5th ed.). Addison-Wesley.

Sommerville, I. (2011). Software engineering (9th ed.). Pearson Education Inc.

Standish-Group. (1999). Chaos: A recipe for success. Standish Group International. Standish-Group. (2003). Chaos—the state of the software industry.

Standish Group International (2014). Chaos Summary for 2014 . Tom Clancy. Standish Group International (2015). Chaos Summary for 2015 . Retrieved from https://www.infoq.com/articles/standish-chaos-2015 Stamelos, I. (2010), Software Project Management

Stefanou, C.J., (2001), A framework for the ex-ante evaluation of ERP software, *European Journal of Information Systems*, vol. 10, pp. 204-215

STOICA, M., MIRCEA, M., & GHILIC-MICU, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*. https://doi.org/10.12948/issn14531305/17.4.2013.06

Swanton B and Draper L (2010), How do you expect to get value from ERP if you don't measure it, *AMR Research*

Symons, V. (1990) Evaluation of information systems: IS development in the processing company, *Journal of Information Technology*, 5, 194-204

Thomas, G., Fernandez, W. (2008) Success in IT projects: A matter of definition? *International Journal of Project Management*, Vol. 26, pp. 733-742

Vroom, V.H., (1964), Work and Motivation, Wiley

Wallace, L., Keil, M., Rai, A. (2004a) Understanding software project risk: a cluster analysis, *Information & Management* 42(1), pp. 115-125

Ward, J., Daniel, E. Peppard, J. (2008) Building better business cases for IT investments, *MIS Quarterly Executive*, Vol. 7 No. 1

Weinberg GM (2011), Exploring Requirements: Quality before Design, Dorset House Publishing

Wilcocks R (1994) *Software*, New Scientist; London Vol. 142, Issue 1931, 45

Wright, K. (2011). *A SURVEY OF INFORMATION SYSTEMS DEVELOPMENT PROJECT PERFORMANCE*.

About the Author

Martin J. Cohen

USA

Martin J. Cohen serves as Adjunct Professor at Stevens Institute of Technology Schools of Engineering and Business, as well as Adjunct Associate Professor at Pace University Lubin School of Business, both in the United States. He holds an MBA degree from the Harvard Business School. His research focusses on entrepreneurship, project selection and project value realization. He has held consulting, management and project management positions in financial services, manufacturing and supply chain organizations. He can be contacted at martin.cohen4@verizon.net