Running head: UNDER WHAT MARKET CONDITIONS ARE AGILE METHODS

Under what Market Conditions are Agile Methods the Optimal Approach to Software Development

David F. Rico

Abstract

The purpose of this document is to serve as a research proposal, which recommends a study to examine the effects of using agile methods on software project performance in dynamic and uncertain market conditions. Agile methods are an approach to software engineering, which are characterized by early customer involvement, flexible development processes, and iterative product releases. This proposal describes the research problem, research theory, literature review, and research methodology for studying the effects of using agile methods. A survey of 25 software projects from 15 Internet firms is suggested to test the hypotheses that better software project performance (e.g., fewer resources, faster time-to-market, and higher customer satisfaction) may be linked to the use of agile methods. The results of this study may help executives and managers better understand the effects of using agile methods by Internet firms on software project performance.

Keywords. Agile methods, early customer involvement, flexible development processes, iterative product releases.

Introduction

The purpose of this document is to serve as a research proposal, which recommends a study to examine the effects of using agile methods on software project performance in dynamic and uncertain market conditions. Agile methods are approaches to software engineering, which are characterized by early customer involvement, flexible development processes, and iterative product releases (Beck, 1999a; Highsmith, 2002; MacCormack, Verganti, & Iansiti, 2001). Traditional approaches to software engineering often include formal software project management, requirements analysis, architecture, design, computer programming, and testing (Beck). Traditional approaches may be linked to unsatisfied customers as a result of rigid, long, and sequential processes that are unresponsive to changing product requirements and market conditions (Beck). This proposal recommends a study to examine the effects of using agile methods by Internet firms in order to help executives and managers better understand the relationship of using agile methods to software project performance.

The scope of this study will be limited to the analysis of early customer involvement, flexible development processes, and iterative product releases, because these characteristics of agile methods may have an impact on software project performance (Beck, 1999a; MacCormack, Verganti, & Iansiti, 2001). There are other tenets of agile methods such as pair programming (Erdogmus & Williams, 2003), which are beyond the scope of this study due to time and resource constraints. This study will also be limited to organizations that make Internet products and services, such as intranets, extranets, web portals, and e-commerce systems because of the pervasive use of these solutions by information technology producers (Highsmith, 2002, p. 12-15; Torkzadeh & Dhillon, 2002) and the suitability of agile methods for these purposes (Highsmith, p. 34-36; MacCormack, 2001).

The relevance, significance, and importance of understanding the relationships between the use of agile methods and software project performance are quite high. The top 500 U.S. firms spend $140 billion per year on information technology (“Masters of Technology,” 2004) and much of the $400 billion U.S. defense budget is devoted to information technology as well (Fulghum & Wall, 2004). Furthermore, there are more than 250,000 software projects in the U.S. of which more than 72% have

failed or are failing (Standish Group, 2000). And, executives and managers of Fortune 1,000 firms representing more than 16 major industries may benefit from the knowledge that agile methods may be linked to better software project performance (Liu, Arnett, & Litecky, 2000).

The key terms and definitions include agile methods, early customer involvement, flexible development processes, and iterative product releases. Agile methods are defined as software development ecosystems or methodologies that describe a holistic environment, which is simultaneously chaotic and orderly, fosters collaborative values and principles, and consists of informal software development processes (Highsmith, 2002, p. xxiii). Early customer involvement is defined as the release of early prototypes and beta versions to customers for evaluation (MacCormack, Verganti, & Iansiti, 2001) or a customer who is a part of the project team (Beck, 1999a). Flexible development processes are defined as the ability to respond to changes in order to be profitable in highly dynamic and uncertain market conditions (Highsmith, p. xxiii) and the use of informal customer requirements, project plans, and computer programming practices (Beck). And, iterative product releases are defined as making "software development economically valuable by spending our money more slowly, earning revenue more quickly, and increasing the probable productive lifespan of our project" (Beck, 1999b, p. 11) or the use of frequent product release cycles (Beck, 1999a).

This proposal is organized into eight sections: an introduction, research problem, research theory, literature review, research methodology, project plan, conclusion, and references. The introduction defines the purpose, scope, relevance, key terms, and organization of the proposal. The research problem identifies the problem statement, background, research questions, goals, and assumptions. The research theory identifies the research hypotheses, variables, model, and approach. The literature review identifies concepts in software project performance (e.g., resources, time-to-market, and customer satisfaction) and agile methods (e.g., early customer involvement, flexible development processes, and iterative product releases). The research methodology outlines the proposed research method, sample size, measurement instrument, data collection procedures, and data analysis procedures. The project plan describes the timeline for the proposed study, the conclusion summarizes this document, and references contain the bibliographic information used in this proposal.

Research Problem

The research problem is simply to investigate possible connections between the use of software methods and some measure of better software project performance. The literature survey contains an in-depth review of key concepts in software project performance (e.g., resources, time-to-market, and customer satisfaction) and agile methods (e.g., early customer involvement, flexible development processes, and iterative product releases). However, few studies empirically link the use of any software method to better software project performance (MacCormack, Verganti, & Iansiti, 2001). Therefore, the general research problem is to verify that a link may exist between the use of agile methods by Internet firms and better software project performance.

The research problem is to collect empirical evidence that might establish a link between the use of agile methods by Internet firms and some measure of better software project performance. Examples of Internet firms are Microsoft, Yahoo, Google, America Online, Amazon, E-Bay, or any number of other lesser known firms. The firms do not have to be major names such as these and do not even have to be commercial Internet service providers, but could certainly be U.S. government agencies and contractors that build Internet products and services. Examples of agile methods are the use of early customer involvement, flexible development processes, and iterative product releases (Beck, 1999a; MacCormack, Verganti, & Iansiti, 2001). Examples or measures of better software project performance are increased revenues, profits, market share, productivity, quality, reliability, customer satisfaction, and time-to-market reductions (Diaz & Sligo, 1997; King & Diaz, 2002).

The research background for this subject area is characterized by numerous attempts to improve software project performance with software methods. Computer programmers, computer scientists, software engineers, and management scientists specializing in technology have been trying to solve computer programming problems such as productivity, quality, reliability, customer satisfaction, cost effectiveness, and time-to-market for more than five decades (Cusumano, 1991; Dijkstra, 1968; Fagan, 1976; Jones, 1985; Kan, Dull, Amundson, Lindner, & Hedger, 1994; Knuth, 1963; MacCormack, Verganti, & Iansiti, 2001; Mays, Jones, Holloway, & Studinski, 1990; Pine, 1989;

Sulack, Lindner, & Dietz, 1989). Fagan introduced software inspections to increase quality and productivity by an order of magnitude in 1976, Jones introduced defect prevention to increase quality by another order of magnitude in 1985, and Sulack, Lindner, and Dietz used early customer involvement and iterative product releases to produce a seven million line of code computer operating system with 30 billion lines of application code in record time (Pine). Hewlett Packard saved $350 million using Fagan's software inspections over a 10 year period (Grady, 1997, p. 252-253), Motorola successfully produced an error free paging system at 25 times the normal productivity levels (Ferguson, Humphrey, Khajenoori, Macke, & Matvya, 1997), and Electronic Brokering Services designed a 65,000 line of code Java system using team processes that conducted $1 billion worth of online trades per day without error in record time (Goth, 2000). Motorola and General Dynamics have even noted order of magnitude improvements in productivity, quality, cycle time, and cost reductions (Diaz & Sligo, 1997; King & Diaz, 2002). Yet, all of these isolated breakthroughs linking software methods to better software project performance have left executives, managers, computer programmers, and management scientists demanding more empirical evidence (Sassenburg, 2002). This is the background, which establishes the context for seeking empirical evidence that may link the use of agile methods by Internet firms to better software project performance.

The research questions relate to the fundamental issue of whether agile methods may be linked to better software project performance. The 1990s and the new millennium have introduced an onslaught of Internet technologies that have transformed industries, the world economy, and computer programming, resulting in highly dynamic and uncertain market conditions (Cao & Gruca, 2003; Carey, 1998; Christensen, Schmidt, & Larsen, 2003; Gandal, 2001; Kauffman & Wang, 2001). Therefore, the basic research question is whether the use of agile methods is linked to better software project performance (e.g., fewer resources, faster time-to-market, and higher customer satisfaction). Agile methods are characterized by three fundamental features: early customer involvement, flexible development processes, and iterative product releases (Beck, 1999a). Therefore, three research questions have been posed as shown in Table 1.

Table 1

Research Questions for Examining Link between Agile Methods and Software Project Performance

Dependent Variables	Research Questions
Early customer involvement	Q_1: In dynamic and uncertain market conditions, is early customer involvement linked to better software project performance?
Flexible development processes	Q_2: In dynamic and uncertain market conditions, are flexible development processes linked to better software project performance?
Iterative product releases	Q_3: In dynamic and uncertain market conditions, are iterative product releases linked to better software project performance?

The research goals and objectives are to objectively gather information that might determine if a link exists between early customer involvement, flexible development processes, iterative product releases, and software projects that consume fewer resources, exhibit faster time-to-market, and result in higher customer satisfaction. The goals of this study include analyzing the software project performance and software methods of organizations that produce Internet products and services. Then, the goals of this study include performing an investigation of whether there may be a correlation between the use of agile methods and better software project performance among them.

The assumptions and constraints of this research proposal include accepting the hypotheses that the current market conditions are dynamic and uncertain (Cao & Gruca, 2003; Carey, 1998; Christensen, Schmidt, & Larsen, 2003; Gandal, 2001; Kauffman & Wang, 2001). Further assumptions and constraints include accepting the hypotheses that fewer resources (McGibbon, 1996), faster time-to-market (Reifer, 2000, 2002), and higher customer satisfaction (Kan, 1995, p. 273-293; Kan, Dull, Amundson, Lindner, & Hedger, 1994; Mackey, 2000) are the principal characteristics of better software project performance. This research is also limited to the study of agile methods, which may be characterized by the use of fewer resources, faster time-to-market, and higher customer satisfaction (Beck, 1999a; Highsmith, 2002, p. 29-34; MacCormack, Verganti, & Iansiti, 2001). Software methods characterized by inflexible project management and development processes will not be analyzed (Humphrey, 1989, 1996, 2000), since they may be incongruent with agile methods (Highsmith, p. xv).

Research Theory

The research theory consists of a proposed design for research hypotheses, variables, a model, and an approach to examine the relationship of using agile methods by Internet firms to software project performance. The proposed research theory is designed to test the relationships between early customer involvement, flexible development processes, iterative product releases, and the characteristics of software projects (e.g., resources, time-to-market, and customer satisfaction). The proposed research theory has been adapted from an empirical study of agile methods (MacCormack, Verganti, & Iansiti, 2001) and other empirical studies of Internet software project performance, which have exhibited some level of historical reliability and validity (Kwon, Kim, & Lee, 2002; Liu, Arnett, & Litecky, 2000; McKinney, Yoon, & Zahedi, 2002; Torkzadeh & Dhillon, 2002).

The research hypotheses are derived from the research questions in Table 1 and are designed to test the research model that the use of agile methods by Internet firms may be linked to better software project performance as shown in Table 2. The first hypothesis tests to see if Internet firms with better software project performance use early customer involvement (e.g., whether customers are present, form informal requirements, and provide early feedback on releases). The second hypothesis tests to see if Internet firms with better software project performance use flexible development processes (e.g., informal planning, flexible scheduling, and non-rigid processes). The third hypothesis tests to see if Internet firms with better software project performance use iterative product releases (e.g., continuous testing, weekly iterations, and monthly releases). Numerous mixed responses to these tests will no doubt result, such as a mix of software project performance and use of agile methods by Internet firms. These results may be a significant finding in order to help executives and managers understand how the use of agile methods by Internet firms may affect software project performance. For instance, the proposed study may reveal that better software project performance is not linked to all three characteristics of agile methods, or vice versa, use of agile methods is not linked to all three characteristics of software project performance. Again, these findings may be significant to executives and managers, so that they may understand the performance implications of using agile methods.

Table 2

Research Hypotheses for Examining Link between Agile Methods and Software Project Performance

Dependent Variables	Research Hypotheses
Early customer involvement	H_1: In dynamic and uncertain market conditions, early customer involvement is linked to better software project performance.
Flexible development processes	H_2: In dynamic and uncertain market conditions, flexible development processes are linked to better software project performance.
Iterative product releases	H_3: In dynamic and uncertain market conditions, iterative product releases are linked to better software project performance.

There are two major classes of research variables, namely software project performance and the principal characteristics of agile methods (see Table 3). The first major class (e.g., software project performance) refers to the resource levels, time-to-market, and customer satisfaction achieved by Internet firms. The characteristics (e.g., independent variables) of resources are development effort, testing effort, and maintenance effort. (Effort is a measure of the staff hours consumed by a software activity.) The characteristics of time-to-market are development schedule, testing schedule, and maintenance schedule. The characteristics of customer satisfaction are software quality, software reliability, and software performance. The second major class (e.g., agile methods) refers to the early customer involvement, flexible development processes, and iterative product releases used by Internet firms. The characteristics (e.g., independent variables) of early customer involvement are customer presence, informal requirements, and customer feedback. The characteristics of flexible development processes are informal planning, flexible scheduling, and non-rigid processes. The characteristics of iterative product releases are continuous testing, weekly iterations, and monthly releases. The major classes of research variables and the independent variables were derived from literature describing the requirements for software methods in general (Kan, 1991, 1995; Kan, Dull, Amundson, Lindner, & Hedger, 1994) and agile methods (Beck, 1999a; Beck 1999b; Highsmith, 2002; MacCormack, Verganti, & Iansiti, 2001).

Table 3

Research Variables for Examining Link between Agile Methods and Software Project Performance

Dependent Variables	Intervening Variables	Independent Variables
Software project performance	Resources	Development effort
		Testing effort
		Maintenance effort
	Time-to-market	Development schedule
		Testing schedule
		Maintenance schedule
	Customer satisfaction	Software quality
		Software reliability
		Software performance
Agile methods	Early customer involvement	Customer presence
		Informal requirements
		Customer feedback
	Flexible development processes	Informal planning
		Flexible scheduling
		Non-rigid processes
	Iterative product releases	Continuous testing
		Weekly iterations
		Monthly releases

Effort refers to the number of staff hours required to perform development, testing, and maintenance. Schedule refers to the length of time required to perform development, testing, and maintenance. Software quality refers to defect density (e.g., number of defects), software reliability refers to mean time between failures (e.g., number of software crashes), and software performance refers to system response times. Alternative models of customer satisfaction may be adapted to accurately measure customer experiences (Kan, 1995, p. 273-293; Kwon, Kim, & Lee, 2002; Liu, Arnett, & Litecky, 2000; McKinney, Yoon, & Zahedi, 2002; Torkzadeh & Dhillon, 2002).

The research model exhibited in Figure 1 shows the possible links between market conditions, software project performance, and the use of agile methods by Internet firms. The research model was principally adapted from an empirical study of the use of agile methods by Internet firms (MacCormack, Verganti, & Iansiti, 2001). However, the actual selection of the dependent variables (e.g., software project performance and agile methods) came from other sources. The selection of the intervening variables for software project performance was heavily influenced by in-depth studies of software process and quality improvement (e.g., Kan, 1991, 1995; Kan, Dull, Amundson, Lindner, & Hedger, 1994; McGibbon, 1996). The selection of the intervening variables for agile methods was principally derived from seminal works describing the characteristics of agile methods (e.g., Beck 1999a, 1999b; Highsmith, 2002). Beck (1999a) principally recommended customer presence, while MacCormack, Verganti, and Iansiti recommended early market feedback via prototypes and beta releases. Beck recommended flexible development processes, while MacCormack, Verganti, and Iansiti recommended flexible Internet product architectures. Beck recommended iterative product releases, while MacCormack, Verganti, and Iansiti recommended overlapping releases.

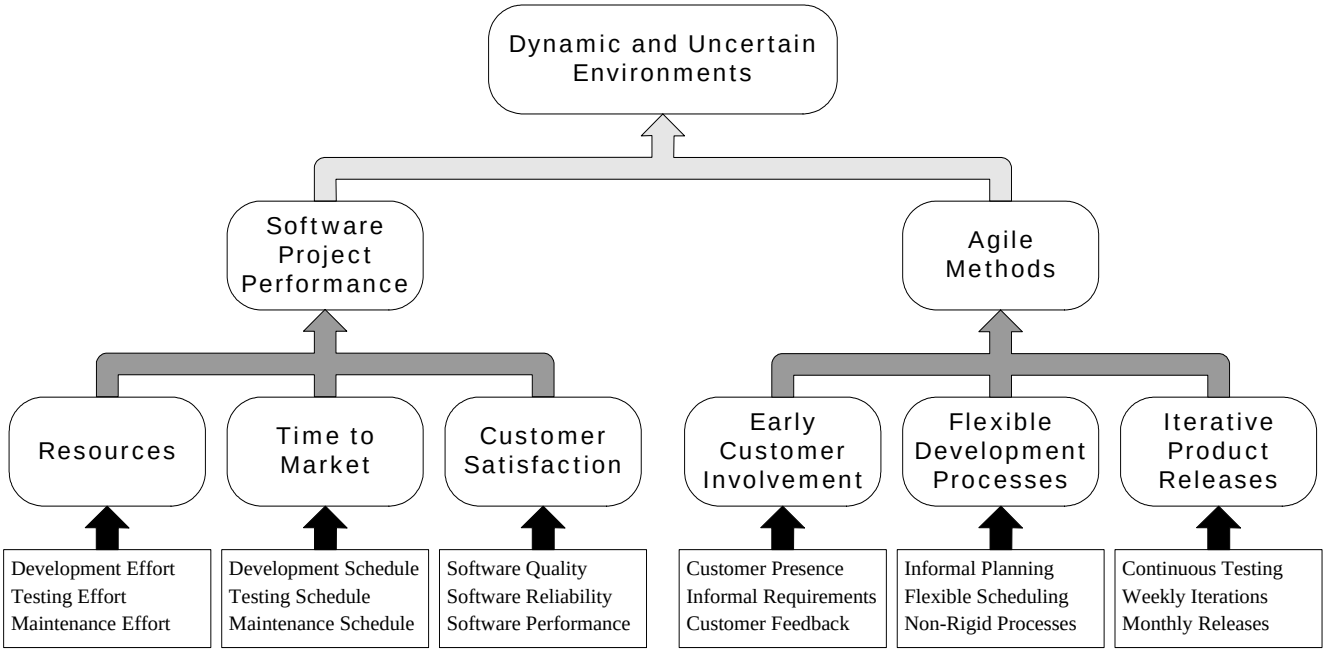

Figure 1. Research Model for Examining Link between Agile Methods and Software Project Performance

The research approach for the proposed study to examine the effects of using agile methods by Internet firms on software project performance is a survey. A survey of 25 projects from 15 Internet firms is suggested in order to test the hypotheses that the use of agile methods may be linked to better software project performance. Questionnaires will be designed based on the constructs depicted in the research model, namely software project performance and agile methods (see Figure 1). The questionnaires will also have categories of questions based on the intervening variables of resources, time-to-market, customer satisfaction, early customer involvement, flexible development processes, and iterative product releases. Individual questions will be based on closed-ended responses to measure the magnitude of development, testing, and maintenance effort and schedules, as well as software quality, reliability, and performance. Questions with closed-ended responses will also be designed to measure customer presence, the use of informal requirements, and the degree of customer feedback; the use of informal planning, flexible scheduling, and non-rigid processes; and the use of continuous testing, weekly iterations, and monthly releases. The purpose of the proposed survey is to collect data from Internet firms and first measure the level of software project performance. Then, the use of early customer involvement, flexible development processes, and iterative product releases will be measured from these same Internet firms. Finally, simple linear regression will be performed to test for a possible correlation between better software project performance and the use of agile methods by these Internet firms. If the proposed study discovers a high incidence rate of significant correlation between better software project performance and the use of agile methods, this may be a significant finding. The ultimate goal and objective of the proposed study is to examine the possible link between better software project performance and agile methods. There are only a few empirical studies that examine the possible links between agile methods and software project performance (Erdogmus & Williams, 2003; MacCormack, Verganti, & Iansiti, 2001). By some estimates, two-thirds of all software projects are now using agile methods, which amount to nearly 165,000 software projects (Sliwa, 2002; Standish Group, 2000). More empirical research may be needed on the potential links between software project performance and the use of agile methods by Internet firms to justify the current adoption rate.

Literature Review

The literature review contains a synopsis of seminal, empirical, and scholarly works, which are related to software project performance and central to the concepts underlying the use of agile methods by Internet firms. This study proposes to examine the effects of using agile methods on software project performance. The constructs associated with software project performance in the research model shown in Figure 1 are resources (e.g., costs), time-to-market, and customer satisfaction. The constructs associated with agile methods in the research model shown in Figure 1 are early customer involvement, flexible development processes, and iterative product releases. Therefore, the literature review examines the history of key management concepts underlying software project performance and the use of agile methods by Internet firms as shown in Figure 2.

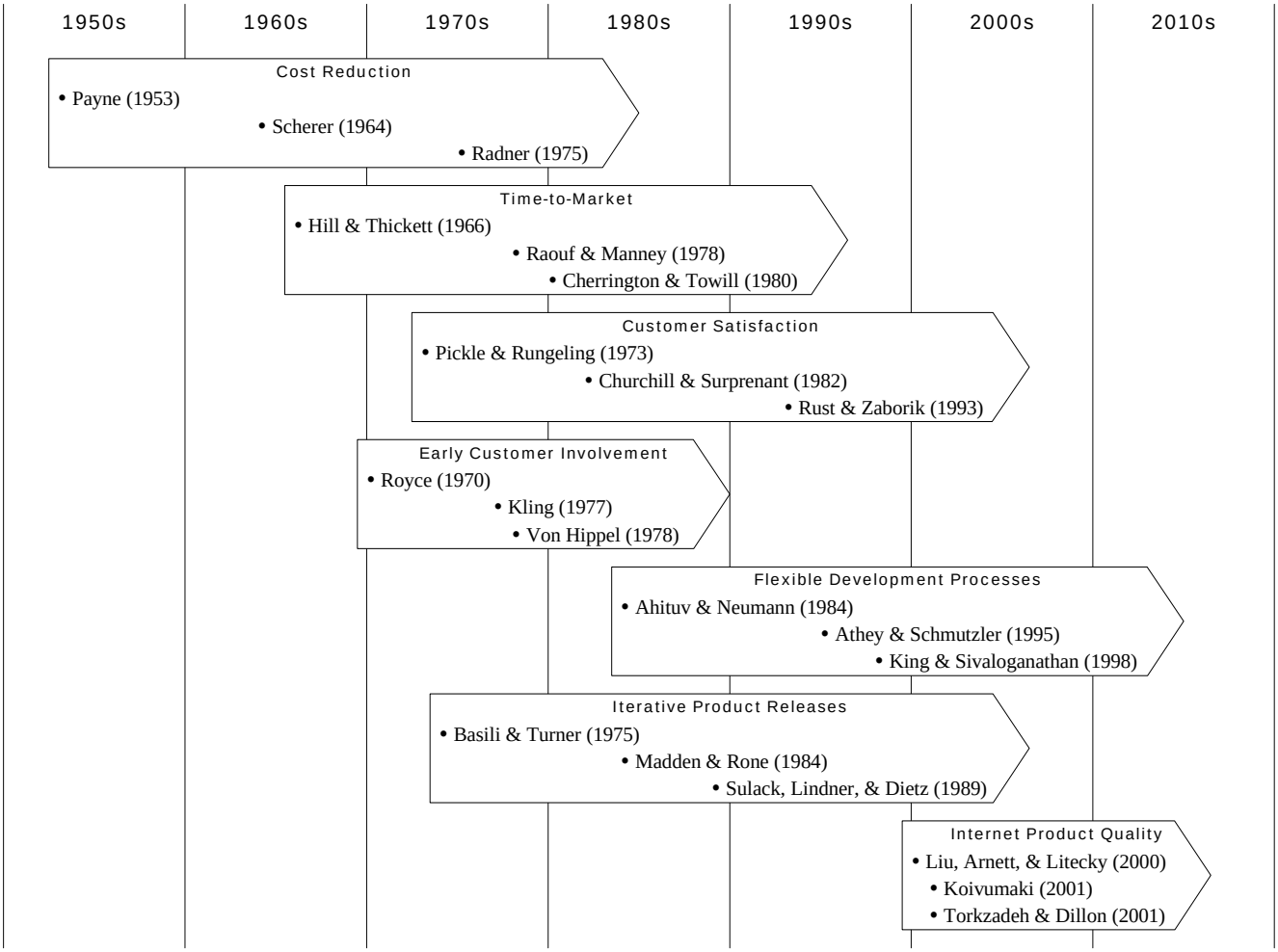

Figure 2. Literature for Examining Link between Agile Methods and Software Project Performance

Cost Reduction

Cost reduction has been a major goal and objective of using software methods (Hager, 1989). Software maintenance had been reported to account for as much as 60% of the total life cycle cost of producing software products and services (Hager). That is, if a software product or service costs $100,000 to design and develop, the same software system required an additional $200,000 to maintain over its entire life cycle (Hager). For years, a major goal of the design of software methods had been to reduce software maintenance expenses, also known as total life cycle costs (Hager). Some recent comparative studies have shown that not all software methods are created equal, and some reduced total life cycle costs by as much as 30 times over their nearest competitors (McGibbon, 1996). Cost reduction had been a major concern of U.S. corporations for decades and some studies showed that costs could be reduced by as much as 15%, 25%, or even 40% (Payne, 1953). Early approaches to cost reduction involved setting up industrial engineering teams, measuring all work activities, analyzing methods to reduce operating expenses, and training employees in the use of these new cost saving techniques (Payne). One early empirical study showed that the U.S. government could reduce costs by as much as 50% by competitively bidding out defense work instead of relying on sole source contracts (Scherer, 1964). Another management scientist designed mathematical models of satisficing for Simon's theory of Bounded Rationality in order to show executives that significant cost reductions could be achieved through the application of sound decision-making processes (Radner, 1975). An early empirical study of product quality, prices, market position, costs, and return on investment of 1,200 firms revealed that higher quality products and services were less expensive to produce, which ran contrary to conventional wisdom during that time period (Phillips, Chang, & Buzzell, 1983). A more recent case study of overlapped product development, which is very similar to the characteristics of agile methods (Beck, 1999a), showed that the use of iterative development releases could increase costs rather than decrease them (Roemer, Ahmadi, & Wang, 2000). These findings were consistent with a survey of Internet firms showing that some aspects of agile methods were more expensive to use than traditional methods (MacCormack, Verganti, & Iansiti, 2001).

Time-to-Market

Time-to-market, also known as cycle time, has been a major goal and objective of using software methods (Diaz & Sligo, 1997). Time-to-market is defined as the length of time a software firm requires to design, develop, test, and deliver a software product or service (Diaz & Sligo). The use of software methods by information technology firms has shown to improve time-to-market by as much as eight times (Diaz & Sligo). The study of time-to-market has been a subject of inquiry for many decades, as exhibited by early empirical studies of highly repetitive work activities versus custom work orders, productivity of well-rested versus fatigued workers, and productivity improvements associated with process reengineering (Cherrington & Towill, 1980; Hill & Thickett, 1966; Raouf & Manney, 1978). A recent empirical study of 21 firms showed that the use of cross functional teams and formal development processes resulted in time-to-market improvements of 50% to 60% (Griffin, 1997). Agile methods may be linked to improved time-to-market because of cross functional teams, but instead advocate informal or flexible development processes (Beck, 1999a). In another recent empirical study of 184 firms, time-to-market improvements were not associated with higher revenues, profits, or organizational performance, and did not exhibit a correlation between time-to-market and the application of cross functional teams (Ittner & Larcker, 1997). This may pose an interesting challenge to the findings of MacCormack, Verganti, and Iansiti (2001), which linked the use of agile methods to better performing projects, since the sample by Ittner and Larcker was three times as large and included nearly 90 information technology firms. An empirical study of 30 software projects exhibited increases in development time with overall decreases in maintenance time due to the use of software methods (Harter, Krishnan, & Slaughter, 2000), which was very similar to the findings by Hager (1989) and McGibbon (1996). There was some evidence indicating that the use of software methods improves time-to-market by reducing total life cycle costs (Diaz & Sligo; Hager; Harter, Krishnan, & Slaughter; McGibbon). Only MacCormack, Verganti, and Iansiti showed evidence that the use of agile methods improved time-to-market through their limited empirical study of 29 projects from 17 Internet firms. Time-to-market was a rarely studied phenomenon (Carmel, 1995) and may be an important finding associated with a study of agile methods by Internet firms for this reason.

Customer Satisfaction

Customer satisfaction has been an important goal and objective of using software methods for companies like IBM that successfully developed a large scale midrange computer operating system in record time (Kan, Dull, Amundson, Lindner, & Hedger, 1994) and Microsoft for developing its entire suite of software products (Cusumano & Selby, 1995, p. 377). The use of customer satisfaction measurement dates back nearly five decades, as indicated by an empirical study of 97 firms linking customer satisfaction to profitability (Pickle & Rungeling, 1973). Another empirical study of two products showed that disconfirmation (e.g., disappointment after purchase) was a significant factor in determining customer satisfaction (Churchill & Surprenant, 1982), and its bibliography showed no earlier works of customer satisfaction than the study by Pickle and Rungeling. A small scale empirical study of 100 respondents showed that customer satisfaction was positively correlated to customer retention, market share, and profitability (Rust & Zaborik, 1993). An empirical study of 40 unique scales for measuring customer satisfaction warned that disconfirmation constructs exhibit high levels of predictive validity (Danaher & Haddrell, 1996). And, numerous other empirical studies linked customer satisfaction to factors such as productivity, profitability, financial performance, quality, and shareholder value (Agus & Abdullah, 2000; Anderson, Fornell, & Mazvancheryl, 2004; Anderson, Fornell, & Rust, 1997; Eklof, Hackl, & Westlund, 1999). In an empirical study of 2,500 users of 100 mainframe software products, IBM found that the most important factors of customer satisfaction were capability (e.g., functionality) and usability (e.g., user friendliness), rather than reliability, installability, maintainability, performance, and documentation, which have been standard measures for the computer industry (Kekre, Krishnan, & Srinivasan,1995). Other empirical studies have found correlations between the use of software methods such as reusability and customer satisfaction (Succi, Benedicenti, & Vernazza, 2001). An empirical study of 180 customers of software products from 18 firms also indicated a correlation between customer satisfaction, product functionality, and customer service (Leem & Yoon, 2004). As far as software methods are concerned, Beck (1999a) showed no correlation between the use of agile methods and customer satisfaction, and MacCormack, Verganti, and Iansiti (2001) only correlated the use of agile methods to quality, reliability, and performance.

Early Customer Involvement

Early customer involvement has been recognized as an important goal and objective of using software methods for nearly four decades, as characterized by the inclusion of soliciting requirements from customers prior to computer programming (Royce, 1970). An early empirical study of 473 organizations demonstrated that customers were involved early in the software development process nearly 56% of the time, and ranged as high as 75% to 80% for individual activities (Kling, 1977). Customers were recognized as the source of design innovations in a unique paradigm called Customer Active Participation (Foxall, Johnson, & Murphy, 1987; Von Hippel, 1978). Even more case studies also recognized the central role of early customer involvement in the new product development process (Saren, 1994; Song & Adams, 1993). An empirical study of 110 respondents demonstrated that early customer involvement was an important tool for designing innovative user interfaces for software products and services (Gietzmann & Selby, 1994). The most popular methods of early customer involvement in software development during the 1990s included quality function deployment, user-oriented product development, concept testing, beta testing, consumer idealized design, the lead user method, and participatory ergonomics (Kaulio, 1998). A detailed case study of 11 end users indicated little or no impact to the design and development of software products and services based on early customer involvement (Sugar, 2001). In contrast, an empirical study of 207 firms indicated that the use of quality function deployment, a form of early customer involvement, results in achievement of quality and cost targets for product development (Hoque, Akter, & Monden, 2000). New and holistic methods are now starting to emerge, which advocate early customer involvement in new product development to a wide range of products and services over the Internet (Burnett, Cook, & Rothermel, 2004; Huang & Mack, 2000; Leonard, 2004). A recent empirical study indicated that a link may exist between the use of agile methods by Internet firms and increases in quality, reliability, and performance (MacCormack, Verganti, & Iansiti, 2001). However, few empirical studies seem to conclusively link early customer involvement in software development to goals such as cost reduction, faster time-to-market, or customer satisfaction, leaving these questions open to further investigation.

Flexible Development Processes

Flexible development processes for software engineering have only recently emerged as an important goal and objective, consisting of iterative or even overlapping product releases, particularly for Internet firms (Aoyama, 1998). An early case study suggested that software methods should vary in rigidity and flexibility based on context specific needs and exhibited a framework for such a purpose (Ahituv, Hadass, & Neumann, 1984). However, flexibility of new product development processes for software as well as other industries has only recently become noticeable, as indicated by a case study of an organization that designed a cost and benefit analysis framework for determining the value of flexibility (Athey & Schmutzler, 1995). In another case study, the use of functional analysis, which is a method of identifying product requirements, prioritizing them, and assigning a monetary value to them, was suggested as means of achieving flexibility in new product development (King & Sivaloganathan, 1998, 1999). In an empirical study of 391 projects, the use of technologies that were adaptable to changing customer requirements was attributed to 55% cost reductions in development effort (Thomke & Reinertsen, 1998). In a case study of a $10 million project at Dell, the use of older inflexible technologies did not prove to be any more economically disadvantageous, because of the risks associated with using newer, innovative, and flexible technologies (Krishnan & Bhattacharya, 2002). A recent empirical study of 40 firms, showed that the use of agile methods, which included early customer involvement, design flexibility, decentralized decision making, and cross functional teams, was correlated to increased competitiveness for Internet products (Kassim & Zain, 2004). Another empirical study of 162 projects from 92 firms, exhibited a correlation between flexible development processes (e.g., product, process, and managerial flexibility) and project success (e.g., increased quality, improved time-to-market, and reductions in cost) in high technology settings (Singh & Sushil, 2004). In an empirical study of 29 projects from 17 Internet firms, MacCormack, Verganti, and Iansiti (2001) found a correlation between both product and process flexibility and better performing software projects (e.g., quality, reliability, and performance). It is important to note that agile software development methods are generally linked to flexible development processes rather than flexible product technologies (Beck, 1999a; 1999b; Highsmith, 2002).

Iterative Product Releases

Iterative product releases have been recognized as an important goal and objective for software development for nearly four decades (Basili & Turner, 1975; Royce, 1970). The use of iterative product releases by IBM to design and develop spacecraft software was viewed as a key principle to the success of NASA's space shuttle program (Madden & Rone, 1984). IBM once again exploited the use of iterative product releases to develop a seven million line of code midrange computer operating system, which helped garner $14 billion in revenues (Sulack, Lindner, & Dietz, 1989). In one case study, the use of iterative product releases for a two year software development project resulted in the delivery of 30% more product functionality with an overall cost reduction of 20% (Woodward, 1999). In another case study of a single firm, 10 of 28 projects that used iterative product releases experienced a time-to-market improvement of 28%, cost reductions of 50%, and implemented 125% more product features (Fichman & Moses, 1999). In a study of 161 software projects, the use of iterative product releases for small projects increased costs and time-to-market by 15%, while their use on medium and large software projects reduced costs and time-to-market by 15% to 35%, respectively (Benediktsson & Dalcher, 2003, 2004). IBM used iterative product releases on four software projects (e.g., a kiosk for the world's fair, an employment management system for the state of Illinois, an automobile loan management system, and a kiosk for New York City's Museum of Modern Art), and attributed improvements in product quality, reductions in training costs, reductions in maintenance costs, and increases in customer satisfaction to their use (Greene, Jones, Matcheri, & Thomas, 2003). MacCormack, Verganti, and Iansiti (2001) attributed the use of iterative product releases combined with overlapping stages to improvements in product quality, reliability, and performance, but did not specifically mention cost reductions, time-to-market improvements, or increases in customer satisfaction, though implied. However, Erdogmus and Favaro (2003) showed that the use of iterative product releases for software development increased estimations of net present value for projects by as much as 300% using formulas for real options. The use of iterative product releases for software development has exhibited benefits for large projects, but negligible benefits for small projects, which may have implications when examining the effects of agile methods on software project performance.

Internet Product Quality

Internet product quality is perhaps the most important goal and objective of using any software method. The question is, “how should organizations measure Internet product quality and which measures should they use?” MacCormack, Verganti, and Iansiti (2001) chose to measure the quality, reliability, and performance of Internet products. There was certainly some empirical basis, which showed that quality, reliability, and performance were adequate measures for determining the success of software methods (Kan, 1991, 1995). IBM computed customer satisfaction of mainframe software products as a function of capability, usability, performance, reliability, documentation, and overall system software (Kan, Dull, Amundson, Lindner, & Hedger, 1994). In another instance, IBM determined that customer satisfaction with mainframe software products was a function of capability, usability, performance, reliability, installability, maintainability, and documentation (Kekre, Krishnan, & Srinivasan, 1995). Kan (1995, p. 273-293) of IBM performed several empirical studies of customer satisfaction to determine the optimal reliability and validity of these mainframe software measures. However, Torkzadeh and Dillon (2001) determined that product choice, online payment method, vendor trust, shopping travel, shopping errors, shopping convenience, ecology, customer relations, and product value were important measures of Internet product quality from a survey of 620 online shoppers. McDevitt (2004) determined that customer satisfaction was more important to online shoppers than it was to traditional shoppers based on a survey of 341 respondents. Based on a survey of over 1,000 respondents, Kwon, Kim, and Lee (2002) found that the most important measures of customer satisfaction for Internet products and services were price level, complexity of product usage, layout, information organization, reference information, convenience functions, auction information, product information, and bidding information. The number of empirical studies measuring customer satisfaction with Internet products and services is rather large (Koivumaki, 2001; Lightner, 2003; Liu, Arnett, & Litecky, 2000; McKinney, Yoon, & Zahedi, 2002; Schubert, 2002). A study of potential links between the use of agile methods by Internet firms and better software project performance may hinge on the selection of contemporary measures of product quality and customer satisfaction.

Research Methodology

The purpose of the proposed research methodology is to gather empirical data about software project performance and use of software methods from Internet firms. A survey was chosen for this purpose in order to generalize from a sample to a population so that inferences may be made about the potential linkages between the use of agile methods and software project performance. A survey is an economical and expedient method of collecting data about the use of agile methods by Internet firms.

A sample of 25 projects from 15 Internet firms is recommended, in order to gather empirical data about the potential relationships between software project performance and use of agile methods. A list of 100 Internet firms with revenues exceeding $20 million will be generated in order to improve the probability that the organizations will have one or more projects that may be using agile methods. All 100 Internet firms will be contacted in advance to determine their level of interest in the proposed survey in order to improve the likelihood that a sample of 15 organizations may be identified.

A survey instrument will be designed based on pre-existing questionnaires from prior studies of software project performance (Torkzadeh & Dhillon, 2001) and the use of agile methods by Internet firms (MacCormack, 1998). However, a new survey instrument may need to be designed in order to reflect the constructs of the research model shown in Figure 1, namely resources, time-to-market, customer satisfaction, early customer involvement, flexible development processes, and iterative product releases. In any case, comparison of the new instrument to prior questionnaires will improve the reliability and validity of gathering empirical data for this proposed study of agile methods.

The data collection procedures may consist of using an automated online computer-assisted survey system in order to gather information about the use of agile methods by 15 Internet firms and the characteristics of software project performance. The questionnaires will be sent out to the sample of 15 Internet firms using the U.S. postal system, if an automated survey system is not available. Data for a minimum of 15 Internet projects, and a maximum of 25, will be collected using this method.

The data analysis procedures will include linear regression to test for a correlation between the use of agile methods by these 15 Internet firms and better software project performance. The use of ordinary least squares is recommended among other techniques for ensuring reliability and validity of the statistical correlations. Testing of the hypotheses will be subsequently performed.

Project Plan

The project plan consists of a timeline of activities for the proposed study to examine potential links between the use of agile methods by Internet firms and better software project performance. Four major groups of activities over a 23 week period are proposed: candidate prepares proposal (e.g., 5 weeks), committee reviews proposal (e.g., 4 weeks), candidate conducts research (e.g., 11 weeks), and candidate defends dissertation (e.g., 3 weeks). The first period (e.g., candidate prepares proposal) will be used to design a survey instrument and organize the draft proposal. The second period (e.g., committee reviews proposal) will be used to solicit committee approval and prepare the final proposal. The third period (e.g., candidate conducts research) will be used to administer the proposed survey to the sample of 15 Internet firms and organize the draft dissertation. The fourth period (e.g., candidate defends dissertation) will be used to defend the dissertation and organize the final dissertation.

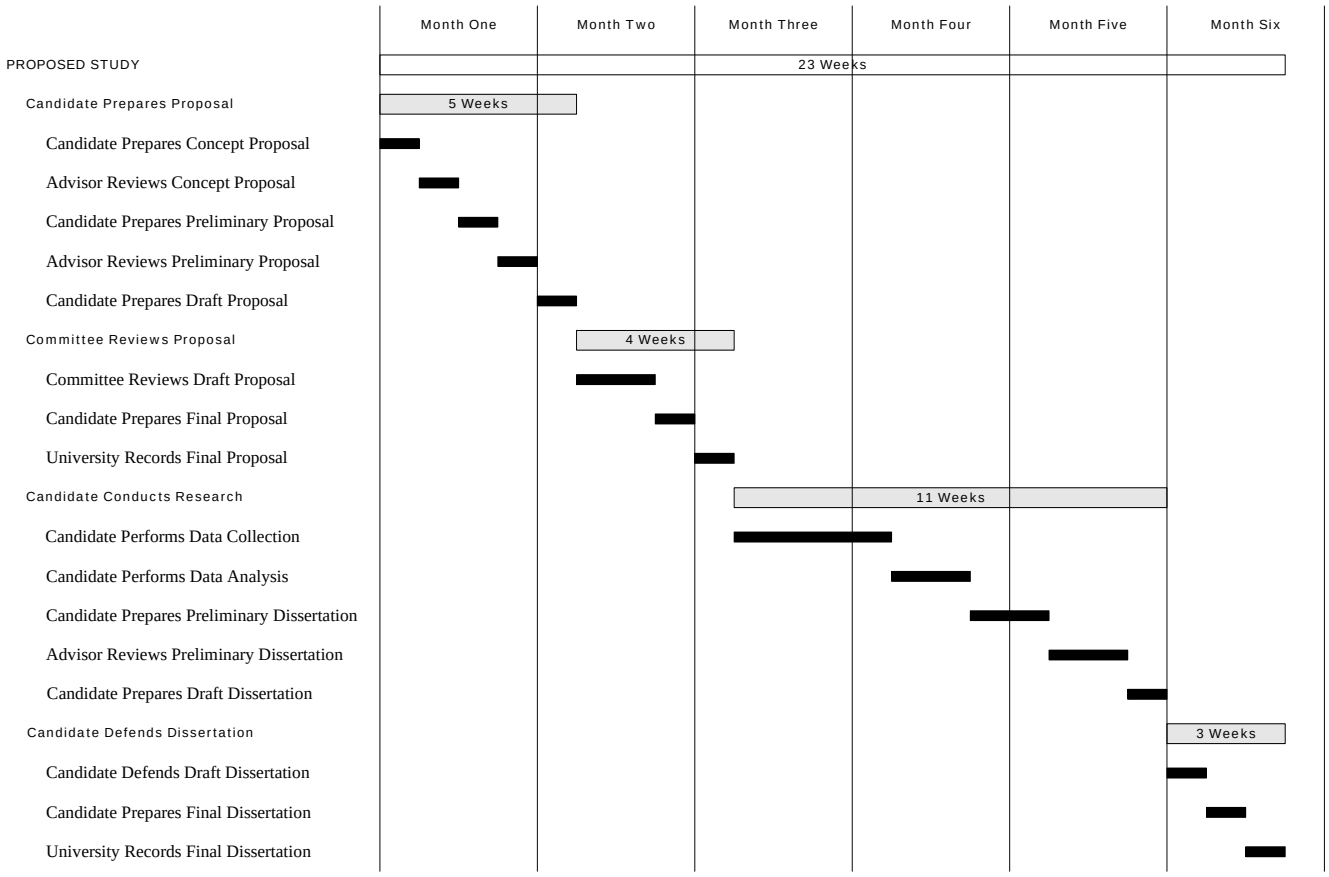

Figure 3. Project Plan for Examining Link between Agile Methods and Software Project Performance

Conclusion

The intention of this research proposal is to recommend a study to examine the potential links between the use of agile methods by Internet firms and better software project performance. Agile methods were defined as an approach to software engineering characterized by early customer involvement, flexible development processes, and iterative product releases. Better software project performance was defined in terms of the use of fewer resources, faster time-to-market, and higher customer satisfaction by Internet firms. Through a survey of 25 projects from 15 Internet firms, this study proposes to generalize from a sample to a population in order to make inferences about potential linkages between the use of agile methods by Internet firms and better software project performance. Similar, but not identical, studies have shown linkages between agile methods and better performing projects by Internet firms, thus demonstrating some empirical evidence of the reliability and validity of these hypothesized inferences and knowledge claims, which may reduce the risks associated with this proposed 23 week study. The uniqueness of the proposed study is a focus on early customer involvement and flexible development processes, which are the tenets of agile methods, versus the manufacturing concepts of rapid prototyping, beta releases, and flexible product architectures. Empirical studies testing the major tenets of agile methods have yet to emerge, thus justifying the relevance, significance, and importance of this proposed study to examine the potential linkages between the use of agile methods by Internet firms and better software project performance. Potential findings along these lines may assist executives and managers with improved decision-making outcomes regarding the use of agile methods by Internet firms.

References

Agus, A., & Abdullah, M. (2000). The mediating effect of customer satisfaction on TQM practices and financial performance. *Singapore Management Review, 22*(2), 55-73.

Ahituv, N., Hadass, M., & Neumann, S. (1984). A flexible approach to information system development. *MIS Quarterly, 8*(2), 69-78.

Anderson, E. W., Fornell, C., & Mazvancheryl, S. K. (2004). Customer satisfaction and shareholder value. *Journal of Marketing, 68*(4), 172-185.

Anderson, E. W., Fornell, C., & Rust, R. T. (1997). Customer satisfaction, productivity, and profitability: Differences between goods and services. *Marketing Science, 16*(2), 129-146.

Aoyama, M. (1998). Web based agile software development. *IEEE Software, 15*(6), 56-65.

Athey, S., & Schmutzler, A. (1995). Product and process flexibility in an innovative environment. *RAND Journal of Economics, 26*(4), 557-574.

Basili, V. R., & Turner, J. (1975). Iterative enhancement: A practical technique for software development. *IEEE Transactions on Software Engineering, 1*(4), 390-396.

Beck, K. (1999a). Embracing change with extreme programming. *IEEE Computer, 32*(10), 70-77.

Beck, K. (1999b). *Extreme programming explained: Embrace the change*. Boston, MA: Addison Wesley.

Benediktsson, O., & Dalcher, D. (2003). Effort estimation in incremental software development. *Journal of Engineering Manufacture, 150*(6), 351-357.

Benediktsson, O., & Dalcher, D. (2004). New insights into effort estimation for incremental software development projects. *Project Management Journal, 35*(2), 5-12.

Burnett, M., Cook, C., & Rothermel, G. (2004). End user software engineering. *Communications of the ACM, 47*(9), 53-58.

Cao, Y., & Gruca, T. S. (2003). The effect of stock market dynamics on Internet price competition. *Journal of Service Research, 6*(1), 24-36.

Carey, J. W. (1998). The Internet and the end of the national communication system: Uncertain predictions of an uncertain future. *Mass Communication Quarterly, 75*(1), 28-34.

Carmel, E. (1995). Cycle time in packaged software firms. J*ournal of Product Innovation Management, 12*(2), 110-123.

Cherrington, J. E., & Towill, D. R. (1980). Learning performance of an industrial long cycle time group task. *International Journal of Production Research, 18*(4), 411-425.

Christensen, J. F., Schmidt, M. L., & Larsen, M. R. (2003). Turbulence and competitive dynamics in the Internet services industry. *Industry and Innovation, 10*(2), 117-143.

Churchill, G. A., & Surprenant, C. (1982). An investigation into the determinants of customer satisfaction. *Journal of Marketing Research, 19*(4), 491-504.

Cusumano, M. A. (1991). Factory concepts and practices in software development. *IEEE Annals of the History of Computing, 13*(1), 3-32.

Cusumano, M. A., & Selby, R. W. (1995). *Microsoft secrets: How the world's most powerful software company creates technology, shapes markets, and manages people*. New York, NY: The Free Press.

Danaher, P. J., & Haddrell, V. (1996). A comparison of question scales used for measuring customer satisfaction. *International Journal of Service Industry Management, 7*(4), 4-26.

Diaz, M., & Sligo, J. (1997). How software process improvement helped motorola. *IEEE Software, 14*(5), 75-81.

Dijkstra, E. W. (1968). The structure of THE: Multiprogramming system. *Communications of the ACM, 11*(5), 341-346.

Eklof, J. A., Hackl, P., & Westlund, A. (1999). On measuring interactions between customer satisfaction and financial results. *Total Quality Management, 10*(4/5), 514-522.

Erdogmus, H., & Favaro, J. (2003). Keep your options open: Extreme programming and the economics of flexibility. In M. Marchesi, G. Succi, J. D. Wells, & L. Williams (Eds.), *Extreme programming perspectives* (pp. 503-552). New York, NY: Addison Wesley.

Erdogmus, H., & Williams, L. (2003). The economics of software development by pair programmers. *The Engineering Economist, 48*(4), 283-319.

Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal, 12*(7), 744-751.

Ferguson, P., Humphrey, W. S., Khajenoori, S., Macke, S., & Matvya, A. (1997). Results of applying the personal software process. *IEEE Computer, 30*(5), 24-31.

Fichman, R. G., & Moses, S. A. (1999). An incremental process for software implementation. *Sloan Management Review, 40*(2), 39-52.

Foxall, G. R., Johnson, B., & Murphy, F. S. (1987). The development of control software for flexible manufacturing systems by british aerospace warton: A case history of user initiated product innovation. *Journal of Marketing Management, 2*(3), 259-274.

Fulghum, D. A., & Wall, R. (2004). Is this war? *Aviation Week & Space Technology, 160*(6), 22-24.

Gandal, N. (2001). The dynamics of competition in the Internet search engine market. International *Journal of Industrial Organization, 19*(7), 1103-1117.

Gietzmann, M. B., & Selby, M. J. P. (1994). Assessment of innovative software technology: Developing an end user initiated interface design strategy. *Technology Analysis & Strategic Management, 6*(4), 473-483.

Goth, G. (2000). The team software process: A quiet quality revolution. *IEEE Software, 17*(6), 125-127.

Grady, R. B. (1997). *Successful software process improvement*. Upper Saddle River, NJ: Prentice Hall.

Greene, S. L., Jones, L., Matcheri, P., & Thomas, J. C. (2003). Iterative development in the field. *IBM Systems Journal, 42*(4), 594-612.

Griffin, A. (1997). The effect of project and process characteristics on product development cycle time. *Journal of Marketing Research, 34*(1), 24-35.

Hager, J. A. (1989). Software cost reduction methods in practice. *IEEE Transactions on Software Engineering, 15*(12), 1638-1644.

Harter, D. E., Krishnan, M. S., & Slaughter, S. A. (2000). Effects of process maturity on quality, cycle time, and effort in software product development. *Management Science, 46*(4), 451-466.

Highsmith, J. (2002). *Agile software development ecosystems*. Boston, MA: Addison Wesley.

Hill, A. B., & Thickett, J. M. B. (1966). Batch size, cycle time, and setting time as determinants of productivity in skilled machining work. *Occupational Psychology, 40*(1/2), 83-89.

Hoque, M., Akter, M., & Monden, Y. (2000). Effects of designers' participation and their evaluation measures on simultaneous achievement of quality and cost of product development teams. *International Journal of Innovation Management, 4*(1), 77-96.

Huang, G. Q., & Mak, K. L. (2000). Modelling the customer–supplier interface over the world wide web to facilitate early supplier involvement in the new product development. *Journal of Engineering Manufacture, 214*(9), 759-769.

Humphrey, W. S. (1989). *Managing the software process*. Reading, MA: Addison Wesley.

Humphrey, W. S. (1996). Using a defined and measured personal software process. *IEEE Software, 13*(3), 77-88.

Humphrey, W. S. (2000). *Introduction to the team software process*. Reading, MA: Addison Wesley.

Ittner, C. D., & Larcker, D. F. (1997). Product development cycle time and organizational performance. *Journal of Marketing Research, 34*(1), 13-23.

Jones, C. L. (1985). A process-integrated approach to defect prevention. *IBM Systems Journal, 24*(2), 150-165.

Kan, S. H. (1991). Modeling and software development quality. *IBM Systems Journal. 30*(3), 351-362.

Kan, S. H. (1995). *Metrics and models in software quality engineering*. Reading, MA: Addison Wesley.

Kan, S. H., Dull, S. D., Amundson, D. N., Lindner, R. J., & Hedger, R. J. (1994). AS/400 software quality management. *IBM Systems Journal, 33*(1), 62-88.

Kassim, N. M., & Zain, M. (2004). Assessing the measurement of organizational agility. *Journal of American Academy of Business, 4*(1/2), 174-177.

Kauffman, R. J., & Wang, B. (2001). New buyer's arrival under dynamic pricing market microstructure: The case of the group buying discounts on the Internet. *Journal of Information Systems, 18*(2), 157-188.

Kaulio, M. A. (1998). Customer, consumer, and user involvement in product development: A framework and a review of selected methods. *Total Quality Management, 9*(1), 141-149.

Kekre, S., Krishnan, M. S., & Srinivasan, K. (1995). Drivers of customer satisfaction for software products: Implications for design. *Management Science, 41*(9), 1456-1470.

King, A. M., & Sivaloganathan, S. (1998). Development of a methodology for using function analysis in flexible design strategies. *Journal of Engineering Manufacture, 212*(3), 215-230.

King, A. M., & Sivaloganathan, S. (1999). Development of a methodology for concept selection in flexible design strategies. *Journal of Engineering Design, 10*(4), 329-349.

King, J., & Diaz, M. (2002). How CMM impacts quality, productivity, rework, and the bottom line. *Crosstalk, 15*(3), 9-14.

Kling, R. (1977). The organizational context of user centered software designs. *MIS Quarterly, 1*(4), 41-52.

Knuth, D. E. (1963). Computer-drawn flowcharts. *Communications of the ACM, 6*(9), 555-563.

Koivumaki, T. (2001). Customer satisfaction and purchasing behaviour in a web based shopping environment. *Electronic Markets, 11*(3), 186-192.

Krishnan, V., & Bhattacharya, S. (2002). Technology selection and commitment in new product development: The role of uncertainty and design flexibility. *Management Science, 48*(3), 313-327.

Kwon, O. B., Kim, C. R., & Lee, E. J. (2002). Impact of website information design factors on consumer ratings of web-based auction sites. *Behaviour & Information Technology, 21*(6), 387-402.

Leem, C. S., & Yoon, Y. (2004). A maturity model and an evaluation system of software customer satisfaction: The case of software companies in Korea. *Industrial Management & Data Systems, 104*(4), 347-354.

Leonard, A. (2004). Focusing on maturity levels for the effective management of end users involved in software project teams. *South African Journal of Business Management, 35*(1), 17-25.

Lightner, N. J. (2003). What users want in e-commerce design: Effects of age, education, and income. *Ergonomics, 46*(1-3), 153-168.

Liu, C., Arnett, K. P., & Litecky, C. (2000). Design quality of websites for electronic commerce: Fortune 1000 webmasters' evaluations. *Electronic Markets, 10*(2), 120-129.

MacCormack, A. (1998). *Managing adaptation: An empirical study of product development in rapidly changing environments*. Unpublished doctoral dissertation, Harvard University, Boston, MA, United States.

MacCormack, A. (2001). Product development practices that work: How Internet companies build software. *MIT Sloan Management Review, 42*(2), 15-24.

MacCormack, A., Verganti, R., & Iansiti, M. (2001). Developing products on Internet time: The anatomy of a flexible development process. *Management Science, 47*(1), 133-150.

Mackey, K. (2000). The work culture at cisco systems. *IEEE Software, 17*(6), 16-18.

Madden, W., & Rone, K. (1984). Design, development, integration: Space shuttle flight software system. *Communications of the ACM, 27*(9), 914-925.

Masters of technology. (2004, September 20). *InformationWeek, 1006*, 45-54.

Mays, R. G., Jones, C. L., Holloway, G. J., & Studinski, D. P. (1990). Experiences with defect prevention. *IBM Systems Journal, 29*(1), 4-32.

McDevitt, P. (2004). Customer satisfaction as a basis for targeting online buyers. *Journal of Targeting, Measurement, & Analysis for Marketing, 12*(4), 329-339.

McGibbon, T. (1996). *A business case for software process improvement* (Contract Number F30602-92-C-0158). Rome, NY: Air Force Research Laboratory—Information Directorate (AFRL/IF), Data and Analysis Center for Software (DACS).

McKinney, V., Yoon, K., & Zahedi, F. M. (2002). The measurement of web customer satisfaction: An expectation and disconfirmation approach. *Information Systems Research, 13*(3), 296-315.

Payne, B. (1953). A program for cost reduction. *Harvard Business Review, 31*(5), 71-82.

Phillips, L. W., Chang, D. A., & Buzzell, R. D. (1983). Product quality, cost position, and business performance: A test of some key hypotheses. *Journal of Marketing, 47*(2), 26-43.

Pickle, H. B., & Rungeling, B. S. (1973). Empirical investigation of entrepreneurial goals and customer satisfaction. *Journal of Business, 46*(2), 268-273.

Pine, B. J. (1989). Design, test, and validation of the application system/400 through early user involvement. *IBM System Journal, 28*(3), 376-385.

Radner, R. (1975). A behavioral model of cost reduction. *Bell Journal of Economics, 6*(1), 196-215.

Raouf, A., & Manney, W. (1978). Variations in cycle time and certain physiological measures of workers performing a paced assembly task. *International Journal of Production Research, 16*(5), 409-426.

Reifer, D. J. (2000). Web development: Estimating quick-to-market software. *IEEE Software, 17*(6), 57-64.

Reifer, D. J. (2002). Estimating web development costs: There are differences. *Crosstalk, 15*(6), 13-17.

Roemer, T. A., Ahmadi, R., & Wang, R. H. (2000). Time-cost trade-offs in overlapped product development. *Operations Research, 48*(6), 858-865.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of the Western Electronic Show and Convention (WESCON 1970), Los Angeles, California, USA*, 1-9.

Rust, R. T., & Zaborik, A. J. (1993). Customer satisfaction, customer retention, and market share. *Journal of Retailing, 69*(2), 193-215.

Saren, M. (1994). Reframing the process of new product development: From stages models to a blocks framework. *Journal of Marketing Management, 10*(7), 633-643.

Sassenburg, H. (2002). The nonsense of ROI. *SPIder Koerier, 2*(4), 4-5.

Scherer, F. M. (1964). The theory of contractual incentives for cost reduction. *Quarterly Journal of Economics, 78*(2), 257-280.

Schubert, P. (2002). Extended web assessment method (EWAM): Evaluation of electronic commerce applications from the customer's viewpoint. *International Journal of Electronic Commerce, 7*(2), 51-80.

Singh, N., & Sushil. (2004). Flexibility in product development for success in dynamic market environment. *Global Journal of Flexible Systems Management, 5*(1), 23-34.

Sliwa, C. (2002). Users warm up to agile programming. *Computer World, 36*(12), 8-8.

Song, J. H., & Adams, C. R. (1993). Differentiation through customer involvement in production or delivery. *Journal of Consumer Marketing, 10*(2), 4-12.

Standish Group. (2000). *Extreme chaos*. West Yarmouth, MA: Author.

Succi, G., Benedicenti, L., & Vernazza, T. (2001). Analysis of the effects of software reuse on customer satisfaction in an RPG environment. *IEEE Transactions on Software Engineering, 27*(5), 473-479.

Sugar, W. A. (2001). What is so good about user centered Design: Documenting the effect of usability sessions on novice software designers. *Journal of Research on Computing in Education, 33*(3), 235-250.

Sulack, R. A., Lindner, R. J., & Dietz, D. N. (1989). A new development rhythm for AS/400 software. *IBM Systems Journal, 28*(3), 386-406.

Thomke, S., & Reinertsen, D. (1998). Agile product development: Managing development flexibility in uncertain environments. *California Management Review, 41*(1), 8-30.

Torkzadeh, G., & Dhillon, G. (2002). Measuring factors that influence the success of Internet commerce. *Information Systems Research, 13*(2), 187-204.

Von Hippel, E. (1978). Successful industrial products from customer ideas. *Journal of Marketing, 42*(1), 39-49.

Woodward, S. (1999). Evolutionary project management. *IEEE Computer, 32*(10), 49-57.