Effects of Agile Methods on Website Quality for Electronic Commerce

David F. Rico

Abstract

The purpose of this paper is to summarize a study of the relationships between the use of agile methods to develop Internet websites and their quality. An agile method is a new product development process that is often associated with the Internet software industry. Agile methods are characterized by factors of iterative development, customer feedback, well-structured teams, and flexibility. Use of agile methods may improve software quality by injecting customer feedback into a stream of working software versions to converge on a solution. Surveys of 250 software professionals were used to help determine whether the use of agile methods improves the quality of Internet websites. The central contributions of this study include conceptual models, survey instruments, and original measurement data for agile methods and website quality. Our findings indicated iterative development and customer feedback were related to website quality, but well-structured teams and flexibility were not.

1. Introduction

Does use of agile methods improve the quality of Internet websites used for the $2 trillion electronic commerce industry by U.S. firms? Agile methods are a new product development process consisting of iterative development, customer feedback, well-structured teams, and flexibility. Use of agile methods may improve website quality by injecting customer feedback into a stream of working software versions to converge on a solution.

There is a well-documented debate among management scholars about the best approaches for managing the development of Internet websites. Some scholars believe traditional methods rooted in well-established scientific management principles lead to high quality Internet websites. However, other management scholars believe agile methods have the characteristics of a job-shop or craft industry predating the scientific management era and are the best approach for managing the development of high-quality Internet websites.

Our challenge is to identify, survey, select, or develop scholarly instruments for measuring the use of agile methods and the quality of Internet websites, and then collect data to determine whether their use is linked to higher quality websites for electronic commerce by U.S. firms. Though agile methods are a general purpose approach for managing the development of new products, they are often associated with Internet software, which is the focus of this study.

There seems to be no middle ground on this issue. Some management scholars firmly believe agile methods lead to lower quality websites and others believe agile methods lead to higher quality websites, with neither side offering much empirical evidence to support their claims. We hope to create the first in a long line of scholarly studies that test the relationships between the use of agile methods as a management approach for developing Internet websites and improved website quality for electronic commerce among U.S. firms.

2. Research Problem

The challenge is to investigate, examine, and determine whether the use of agile methods for managing the development of Internet software is linked to website quality. There is scant literature that investigates the linkages between the use of agile methods and project outcomes, such as organizational performance [4]. There is even some literature that links the use of agile methods to higher website quality for the field of electronic commerce [13].

However, the major tenets, principles, and factors of agile methods had yet to fully evolve and emerge at the time of some of these writings. Few, if any, studies examine the effects of all four of the factors associated with agile methods: (a) iterative development, (b) customer feedback, (c) well-structured teams, and (d) flexibility. Furthermore, few studies link any of these factors associated with agile methods to outcomes, such as website quality for electronic commerce, in a scholarly manner. Therefore, this study proposes to analyze the effects of all four factors of agile methods and then empirically link these factors to scholarly models of website quality for electronic commerce.

The basic research area to be explored is whether the use of agile methods to develop Internet websites is linked to their quality among U.S. firms. Is the use of iterative development linked to website quality? Is the use of customer feedback linked to website quality? Is the use of well-structured teams linked to website quality? Is the use of flexibility linked to website quality? Equally important is whether the answer to these questions is “no.”

3. Literature Review

3.1 New Product Development Game

In 1986, two management scholars from the School of International Corporate Strategy at Hitotsubashi University in Tokyo, Japan, published a approach called the "new product development game" in the Harvard Business Review [20]. In their article, they argued that Japanese "companies are increasingly realizing that the old sequential approach to developing new products simply will not get the job done." They cited the sport of Rugby as the inspiration for the principles of their new product development game—In particular, Rugby's special play called the Scrum, when the players interlock themselves together as a tightly bound group to gain possession of the ball. The new product development game consisted of six major factors: (a) built-in instability, (b) self organizing project teams, (c) overlapping development phases, (d) multi-learning, (e) subtle control, and (f) organizational transfer of learning.

3.2 New Development Rhythm

In 1989, three managers from IBM in Rochester, Minnesota, published an article on how IBM devised a management approach called the "new development rhythm," to bring the AS/400 midrange computer to market in only two years [19]. In their article, they stated that "user involvement programs yielded a product offering that met the user requirements with a significantly reduced development cycle." The new development rhythm consisted of six major factors: (a) modularized software designs, (b) software reuse, (c) rigorous software reviews and software testing, (d) iterative development, (e) overlapped software releases, and (f) early user involvement and feedback.

3.3 Crystal Methods

In 1991, a software manager with IBM was asked to create an approach for managing the development of object oriented systems called "crystal methods" [3]. Crystal methods were piloted on a "$15 million firm, fixed-price project consisting of 45 people." Crystal methods are a "family of methods with a common genetic code, one that emphasizes frequent delivery, close communication, and reflective improvement." The seven properties of crystal methods are: (a) frequent delivery; (b) reflective improvement; (c) close communication; (d) personal safety; (e) focus; (f) easy access to expert users; and (g) a technical environment with testing, configuration management, and frequent integration.

3.4 Scrum

In 1993, Jeff Sutherland of the Easel Corporation adapted the principles from the "new product development game" [20] to the field of computer programming management, explicitly calling it "scrum" [18]. In particular, scrum assumes that the "systems development process is an unpredictable and complicated process that can only be roughly described as an overall progression." Furthermore, scrum's creators believed "the stated philosophy that systems development is a well understood approach that can be planned, estimated, and successfully completed has proven incorrect in practice." Therefore, scrum's creators set out to define a process as a "loose set of activities that combines known, workable tools and techniques with the best that a development team can devise to build systems." Today, scrum is composed of three broad phases: (a) pre-sprint planning, (b) sprint, and (c) post-sprint meeting.

3.5 Dynamic Systems Development

In 1993, 16 academic and industry organizations in the United Kingdom banded together to create a management approach for commercial software called the "dynamic systems development method" or simply DSDM [14]. Their goal was to "develop and continuously evolve a public domain method for rapid application development" in an era dominated by proprietary methods. Initially, DSDM emphasized three success factors: (a) "the end user community must have a committed senior staff that allows developers easy access to end users," (b) "the development team must be stable and have well established skills," and (c) "the application area must be commercial with flexible initial requirements and a clearly defined user group." These were expanded to functionality versus quality, product versus process, configuration management, business objectives focus, testing, risk management, and flexible requirements. DSDM consists of five major stages: (a) feasibility study, (b) business study, (c) functional model iteration, (d) design and build iteration, and (e) implementation.

3.6 Synch-n-Stabilize

In 1995, two management scholars from MIT's Sloan School of Management published a textbook on how Microsoft managed the development of

software for personal computers, dubbed the "synch-n-stabilize" approach [4]. Experts on software management approaches for the mainframe market, their two year case study from 1993 to 1995 was more of a grounded theory or emergent research design. At one point in their textbook, they stated that "during this initial research, it became clear why Microsoft was able to remain on top in its industry while most of its contemporaries from the 1970s had disappeared." Synch-n-stabilize consisted of six major factors: (a) parallel programming and testing, (b) flexible software requirements, (c) daily operational builds, (d) iterative development, (e) early customer feedback, and (f) use of small programming teams. This influential study was based on principles from [19].

3.7 Feature Driven Development

In 1997, three software managers and five software developers created a software development approach called "feature driven development" to help save a failed project for an international bank in Singapore [15]. In their textbook, they stated that "the bank had already made one attempt at the project and failed, and the project had inherited a skeptical user community, wary upper management, and a demoralized development team." Feature driven development consists of five phases: (a) develop an overall model, (b) build a features list, (c) plan by feature, (d) design by feature, and (e) build by feature. Feature driven development also consists of other best practices in software management and development such as domain object modeling, developing by feature, individual class ownership, feature teams, inspections, regular builds, configuration management, and reporting and visibility of results.

3.8 Open Source Software Development

The term "open source software development" or OSS was coined in 1997, though the practice of open source software started in 1970 [2]. Simply put, open source software is a "set of computer instructions that may be used, copied, modified, and distributed by anyone, anywhere, and for any purpose whatsoever" [7]. Another definition stated "open source software is labeled with free source, fast evolution, and extensive user collaboration" [22]. One study identified eight factors of open source software: (a) is parallel rather than linear; (b) involves large communities of globally distributed developers; (c) utilizes truly independent peer review; (d) provides prompt feedback to user and developer contributions; (e) includes the participation of highly talented developers; (f) includes increased user involvement; (g) makes use of extremely rapid release schedules; and (h) produces evolutionary designs [6]. One author wryly mused, "Internet time refers to something much faster, revolutionary, and more basic—It describes the process of developing open source software" [16].

3.9 Judo Strategy

In 1998, two management scholars from both the Harvard Business School and MIT's Sloan School of Management published a textbook on how Netscape managed the development of software for the Internet, dubbed the "judo strategy" [5]. The more notable characteristics of Netscape's judo strategy included: (a) design products with modularized architectures; (b) use parallel development; (c) rapidly adapt to changing market priorities; (d) apply as much rigorous testing as possible; and (e) use beta testing and open source strategies to solicit early market feedback on features, capabilities, quality, and architecture.

3.10 Internet Time

In 1998, a management scholar from the Harvard Business School conducted a study on how U.S. firms manage the development of websites, referring to his approach as "Internet time" [13]. His study states that "constructs that support a more flexible development process are associated with better performing projects." He surveyed 29 projects from 15 Internet firms such as Microsoft, Netscape, Yahoo, Intuit, and Altavista. He set out to test the theory that website quality was associated with three major factors: (a) greater investments in architectural design, (b) early market feedback, and (c) greater amounts of generational experience.

3.11 Extreme Programming

In 1998, 20 software managers working for the Chrysler Corporation published an article on how they devised a management approached called "extreme programming" or XP to turn around a failing software project that would provide payroll services for 86,000 Chrysler employees [1]. Extreme programming consisted of 13 factors: (a) planning game, (b) small releases, (c) metaphor, (d) simple design, (e) tests, (f) refactoring, (g) pair programming, (h) continuous integration, (i) collective ownership, (j) onsite customer, (k) 40 hour workweek, (l) open workspace, and (m) just rules.

4. Conceptual Framework

The conceptual framework presented in Figure 1 is based on four major factors of agile methods identified from an analysis of agile methods: (a) iterative development, (b) customer feedback, (c) well-structured teams, and (d) flexibility. In total, nine major agile methods were analyzed: new development rhythm, scrum, dynamic systems development, synch-n-stabilize, feature driven development, open source software, judo strategy, Internet time, and extreme programming.

Furthermore, four hypotheses were formulated linking the factors of agile methods to website quality: (a) iterative development is linked to higher website quality, (b) customer feedback is linked to higher website quality, (c) well-structured teams are linked to higher website quality, and (d) flexibility is linked to higher website quality.

Since the 1980s, the factors of new product development have been adapted to software methods to produce innovatively new computer software. The new development rhythm emphasized early user involvement, iterative processes, cross-functional teams, and modularity. Scrum emphasized stakeholder feedback, iterative development, self-managed teams, and early architectural design. The dynamic systems development method emphasized user involvement and stakeholder cooperation, iterative development, empowered teams, and simple flexible designs. Synch-n-stabilize emphasized continuous customer feedback, iterations, small teams, and evolving specifications. Internet time emphasized early market feedback, prototypes and beta versions, experienced teams, and architectural design. The judo strategy emphasized early market feedback, beta testing, small teams, and cross-platform designs. All of the known agile methods have four major factors in common: (a) iterative development, (b) customer feedback, (c) well-structured teams, and (d) flexibility.

There have been hundreds of studies of software methods since the 1950s, there have been hundreds of studies on what constitutes system success, and there have been hundreds of studies linking subfactors of software methods to the subfactors of system success. However, there are several limitations associated with some of these more recent studies of agile methods among Internet firms: (a) none of these studies were based on a conceptual framework, (b) few of them were based on all four major factors of agile methods, and (c) none of them attempted to link the factors of agile methods to scholarly models of website quality. Thus, the conceptual framework identified here may be one of the first holistic theories of agile methods.

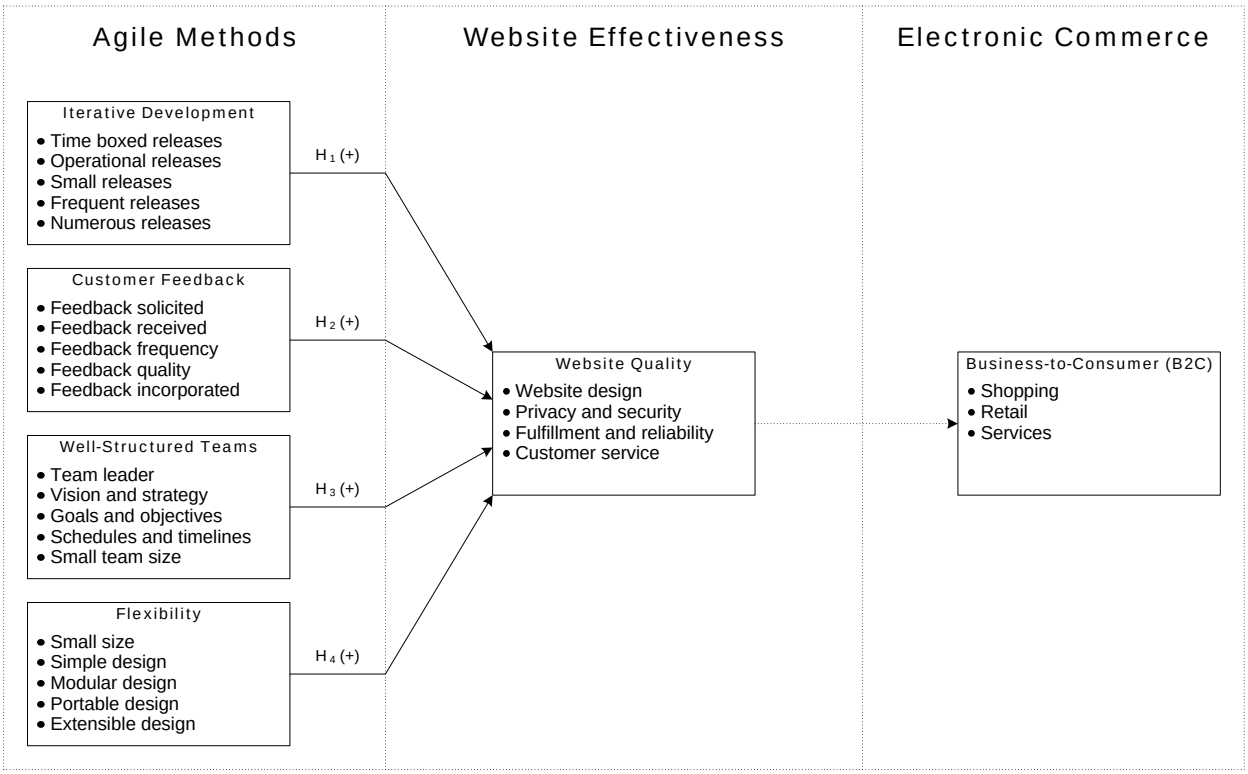

Figure 1. Conceptual Model of Agile Methods, Website Quality, and E-Commerce.

4.1 Iterative Development

Iterative development was defined as "an approach to building software (or anything) in which the overall lifecycle is composed of several iterations in sequence" [12]. Sometimes iterative development is a scaled-back traditional software life cycle completed within three to six months (e.g., new development rhythm, dynamic systems development method, synch-n-stabilize, extreme programming, scrum, and feature driven development). In simpler, leaner approaches, iterative development refers to a much more dynamic daily release of beta versions to the market using the Internet (e.g., Internet time, judo strategy, and open source software development). Therefore, the following five subfactors of iterative development have been selected from the literature on agile methods: (a) time-boxed releases, (b) operational releases, (c) small releases, (d) frequent releases, and (e) numerous releases.

4.2 Customer Feedback

Customer feedback is "a general term describing direct contact with users and covering many approaches" and lays on the "continuum from informative, through consultative to participative" [11]. Sometimes customer feedback refers to user participation in all life cycle activities (e.g., new development rhythm, dynamic systems development method, extreme programming, feature driven development, and open source software development). In simpler and leaner approaches, customer feedback refers to solicitation, receipt, and incorporation of market feedback on beta releases (e.g., synch-n-stabilize, Internet time, judo strategy, and scrum). Therefore, the following five subfactors of customer feedback have been selected from the literature on agile methods: (a) feedback solicited, (b) feedback received, (c) feedback frequency, (d) feedback quality, and (e) feedback incorporated.

4.3 Well-Structured Teams

Well-structured teams are defined as "work groups who are made up of individuals who see themselves as a social entity, who are interdependent because of the tasks they perform, who are embedded in one or more larger social systems, and who perform tasks that affect others" [9]. Sometimes well-structured teams have historically been regarded as small groups of workers who are responsible for accomplishing their tasks with little or no supervision. In simpler, leaner approaches, well-structured teams call for strategic vision to be mixed together with small teams who were responsible for producing releases of working software as rapidly as possible, even on a daily basis (e.g., synch-n-stabilize, judo strategy, Internet time, and open source software development). Therefore, the following five subfactors of well-structured teams have been selected from the literature on agile methods: (a) team leader, (b) vision and strategy, (c) goals and objectives, (d) schedules and timelines, and (e) small team size.

4.4 Flexibility

Flexibility is defined as "the ease with which a system or component can be modified for use in applications or environments other than those for which it was specifically designed" [10]. Sometimes flexibility refers to "responding to change," which embodies the theory of "adaptable organizations" and is represented by an entire genre of popular literature equating organizations to living organisms, evolutionary biology, and survival of the fittest. In simpler and leaner approaches, flexibility refers to a product architecture, which allows for a product to gradually grow, expand, and evolve as engineers solicit, acquire, and incorporate more and more end user, customer, and market needs (e.g., synch-n-stabilize, judo strategy, Internet time, and open source software development). Therefore, the last major factor of agile methods has been named "flexibility" and the following five subfactors have been selected from the literature on agile methods: (a) small size, (b) simple design, (c) modular design, (d) portable design, and (e) extensible design.

4.5 Website Quality

Within the context of electronic commerce, website quality refers to "the extent to which a website facilitates efficient and effective shopping, purchasing, and delivery of products and services" [8]. Sometimes website quality models have numerous factors and subfactors, as well as unusually large measurement instruments, which are economically prohibitive to apply. Since the purpose of this study is to determine the effects of agile methods on website quality, we propose to use the eTailQ model to measure website quality. The eTailQ model is unusually concise, was extensively tested on over one thousand respondents, and exhibits high levels of reliability and validity [21]. The eTailQ model of website quality consists of four major subfactors: (a) website design, (b) privacy and security, (c) fulfillment and reliability, and (d) customer service.

5. Research Method

5.1 Survey method

The research design was a small-scale survey in order to measure the relationships between the four factors of agile methods and scholarly models of website quality. Survey research is one of the premier approaches for collecting both small and large quantities of data to support scholarly research in the fields of both administrative and management science. Survey research has been successfully used to study the four factors of agile methods: (a) iterative development, (b) customer feedback, (c) well-structured teams, and (d) flexibility.

5.2 Population and sample

The sample consisted of up to 30,000 subscribers to a major U.S. computer programming journal. The subscribers largely represent managers and programmers within the U.S. computer programming industry. Recent response rates to similar surveys using this journal have ranged from 700 (2.3%) to 2,500 (8.3%) respondents. So, we expected around 300 (1%) of the magazine's subscribers to respond to our survey of agile methods and website quality.

5.3 Research Variables

The research variables for agile methods consist of time boxed releases, operational releases, small releases, frequent releases, numerous releases, feedback solicited, feedback received, feedback frequency, feedback quality, feedback incorporated, team leader, vision and strategy, goals and objectives, schedules and timelines, small team size, small size, simple design, modular design, portable design, and extensible design.

The research variables for website quality consist of in-depth information, processing efficiency, processing speed, personalization, product selection, protection of privacy, feelings of safety, adequate security, order received, on time delivery, order accurate, willingness to respond, desire to fix issues, and promptness of service.

5.4 Research Instruments

The research instrument for agile methods consisted of the following items: we develop software using time-based iterations; we develop software using operational iterations; we develop software using small iterations; we develop software using daily, weekly, bi-weekly, or monthly iterations; we develop software using multiple (several) iterations; we seek customer feedback on our software iterations; we receive customer feedback on our software iterations; we receive timely customer feedback on our software iterations; we receive a lot of (detailed) customer feedback on our software iterations; we incorporate customer feedback into our software iterations; our software teams have clear administrative or technical leaders; our software teams have clear visions, missions, or strategies; our software teams have clear goals or objectives; our software teams have clear schedules or timelines; our software teams have a small size with no more than 10 people; our software is designed to be as small as possible; our software is designed to be as simple as possible; our software is designed to be modular or object-oriented; our software is designed to work on multiple operating systems; and our software is designed to be changed, modified, or maintained.

The research instrument for website quality consisted of the following items: the website provides in-depth information; the site doesn't waste my time; it is quick and easy to complete a transaction at this website; the level of personalization at site is about right, not too much or too little; this website has good selection; I feel like my privacy is protected at this site; I feel safe in my transactions with this website; the website has adequate security features; you get what you ordered from this site; the product is delivered by the time promised by the company; the product that came was represented accurately by the website; the company is willing and ready to respond to customer needs; when you have a problem, the website shows a sincere interest in solving it; and inquiries are answered promptly.

5.5 Research Scales

All of the instrument items for the survey instruments consisted of a seven point Likert-type scale: (a) strongly disagree, (b) disagree, (c) somewhat disagree, (d) neutral, (e) somewhat agree, (f) agree, and (g) strongly agree.

5.6 Data Collection Process

We conducted cognitive interviews to test the survey instruments. Then we surveyed 250 developers to determine the extent to which they were using agile methods. We also asked them to provide the addresses of websites they have produced. We then conducted an independent assessment of the quality of the websites. We also collected self-report data on the benefits of agile methods, in order to gauge the progress of the data collection process, such as improvements in cost efficiency, productivity, quality, cycle time, and customer satisfaction.

6. Data Analysis

6.1 Descriptive Data

Almost 250 respondents provided data on agile methods, 150 respondents provided data on project outcomes, and 10 respondents provided data on website quality. The first two groups were self-reported data and the last group was based on an independent assessment of website quality. There were an adequate number of data points on agile methods and project outcomes to analyze the relationships between these variables. A total of 27 Internet addresses were provided, but only 10 of them were for e-commerce websites. The means were quite high for agile methods, which may mean that respondents agreed with statements about these variables. The means for project outcomes were lower, which means respondents were a little more conservative about their statements of benefits. The means for website quality were also high.

6.2 Demographic Data

Software engineers represented the largest job function at 33%. However, 38% of the job functions ranged from executives to project managers. The largest group of respondents had 11 to 15 years of experience, came from firms with under 20 employees, and came from firms with under $1 million in annual revenues. The largest industry sectors were manufacturing, information, finance, and professional. The ratio of respondents for demographic data to agile methods data was quite high. Early pilot testing indicated respondents would be hesitant to provide demographic data. However, this was not the case after all.

6.3 Agile Methods Data

For iterative development, the majority of the respondents answered with strongly agree, with the exception of operational releases. For customer feedback, the majority of respondents strongly agreed with feedback solicited, agreed with feedback received, somewhat agreed with feedback frequency, agreed with feedback quality, and strongly agreed with feedback incorporated. This revealed a minor trend in which feedback is solicited from customers, but not always received, not always received frequently, and is not always of the best quality. For well-structured teams, the majority of the respondents agreed with team leader, vision and strategy, goals and objectives, and schedules and timelines, while strongly agreeing with small team size. For flexibility, small size should have been better phrased as small change, and portable design should have been better qualified to strengthen these responses. These data show the ability of the survey instrument to measure compliance with agile methods.

6.4 Website Quality Data

For website design, the majority of the respondents answered with agree, and strongly agree for processing speed. For privacy and security, the majority of respondents strongly agreed with protection and privacy, feelings of safety, and adequate security. For fulfillment and reliability, the majority of the responses were equally assigned to agree and strongly agree for order received, on-time delivery, and order accurate. For customer service, the majority of respondents answered strongly agree for willingness to respond, desire to fix issues, and promptness of service. These data generally show the ability of the measurement instrument to gauge perceptions of website quality. The data were arranged from left to right to show the degree to which the evaluator strongly disagreed, disagreed, somewhat disagreed, was neutral, somewhat agreed, agreed, or strongly agreed with perceptions of website quality. As shown, the largest issues seem to be slow processing speed, inadequate product selection, and in some cases inadequate security and failure to deliver.

6.5 Agile Methods and Website Quality

Five statistical models were constructed between the four major factors of website quality (including a composite model called eTailQ) and the four major factors of agile methods. Two of the models, privacy and security and fulfillment and reliability as a function of iterative development, customer feedback, well-structured teams, and flexibility had high adjusted R^2 values (and were statistically significant). The composite model, eTailQ was significant at the 0.10 level, which was far above the minimum threshold for significance used in this analysis. About half of the Beta values associated with the factors of iterative development, customer feedback, well-structured teams, and flexibility were statistically significant. Only one of the models, fulfillment and reliability, had a high adjusted R^2 value, good F-value, high significance, and statistically significant Beta values. The weakest model was the customer service model, though few of the models were very strong. This analysis indicates the aggregated factors of agile methods are strongly correlated to two of the factors of website quality (67% and 84%) and aggregated factors of website quality (e.g., 54%).

7. Conclusions

Does use of iterative development, customer feedback, well-structured teams, and flexibility improve e-commerce website quality? Based on our analysis, use of iterative development and customer feedback improved website quality, but well-structured teams and flexibility did not (as shown in Figure 2). There was some evidence that iterative development was correlated to website quality, website design, privacy and security, and fulfillment and reliability at the 0.05 level. Customer feedback was correlated to website quality and fulfillment and reliability at the 0.10 level. Well structured teams were negatively correlated to website quality and fulfillment and reliability at the 0.10 level. Flexibility was negatively correlated to website quality, privacy and security, fulfillment and reliability, and customer service at the 0.05 level. However, our hypotheses were stated as positive correlations, so negative ones are viewed as failed hypotheses. The final analysis indicates iterative development and customer feedback are correlated to factors of website quality.

Does use of agile methods improve the quality of Internet websites used for the $2 trillion electronic commerce industry by U.S. firms? Does the use of iterative development, customer feedback, well-structured teams, and flexibility lead to higher quality Internet websites? A survey of 250 respondents was conducted to help determine whether the use of agile methods is linked to website quality. The results of this study showed that the use of iterative development and customer feedback is linked to higher website quality, but not well-structured teams and flexibility. However, agile methods were linked to other benefits such as improvements in cost efficiency, productivity, quality, cycle time, and customer satisfaction. Other contributions of this study include a history of agile methods, a conceptual framework, survey instruments, a repository of original data, and a roadmap for conducting future studies of agile methods.

Recommendations for future work include studying a broader industry sector, refining the conceptual model, scaling the research scope, and carefully selecting proper outcome measures. More recommendations include reexamining our research method, carefully selecting our source of data, and ensuring the success of our data collection. Some of these recommendations are short-term improvements and some are long-term improvements. Future studies may seek to refine our conceptual model of agile methods or devise a conceptual model for a new product development process scaled-up to the program, product-line, or organizational level. The detailed results of this study may be found at [17].

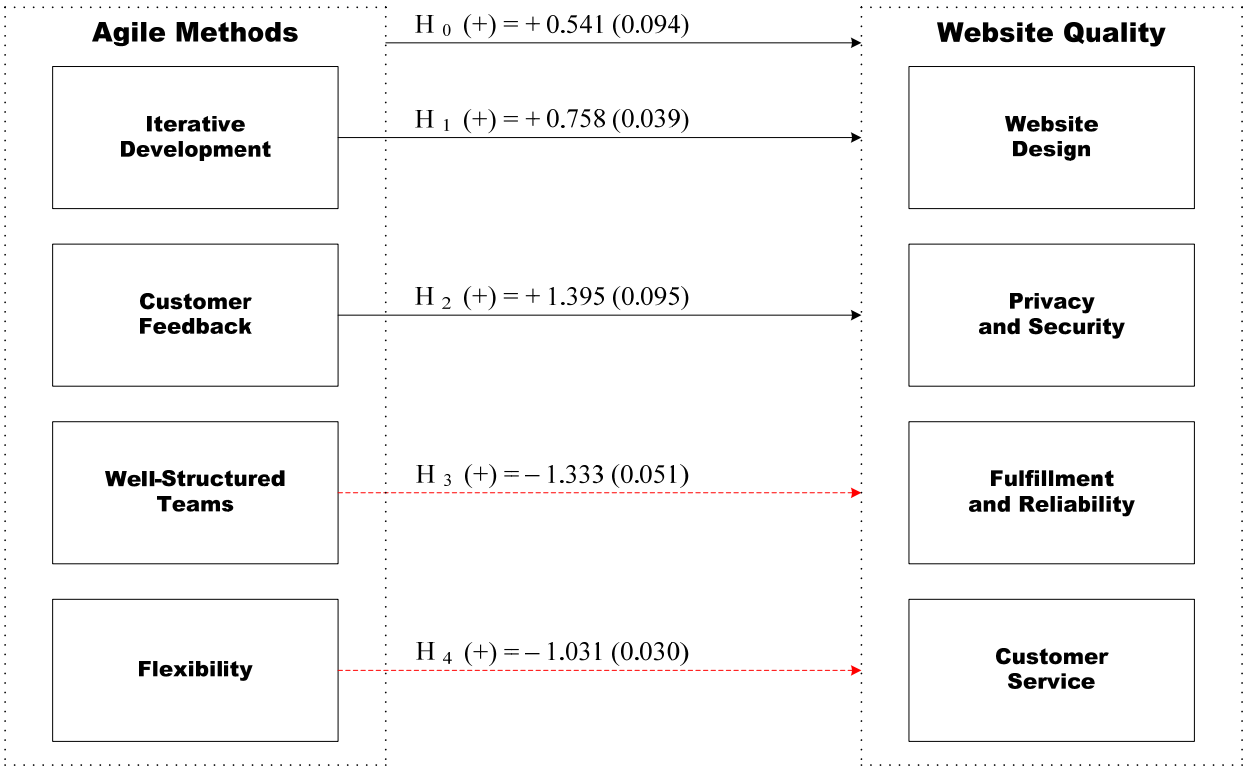

Figure 2. Final Conceptual Model of Agile Methods and Website Quality.

8. References

[1] Anderson, A., Beattie, R., Beck, K., Bryant, D., DeArment, M., Fowler, M., et al. (1998). Chrysler goes to extremes. *Distributed Computing Magazine, 1*(10), 24-28.

[2] Bretthauer, D. (2002). Open source software: A history. *Information Technology and Libraries, 21*(1), 3-10.

[3] Cockburn, A. (2002). *Agile software development*. Boston, MA: Addison Wesley.

[4] 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.

[5] Cusumano, M. A., & Yoffie, D. B. (1998). *Competing on internet time: Lessons from netscape and its battle with microsoft*. New York, NY: The Free Press.

[6] Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*. London, England: Pearson Education.

[7] Fink, M. (2003). *The business and economics of linux and open source*. Upper Saddle River, NJ: Prentice Hall.

[8] Gounaris, S., Dimitriadis, S., & Stathakopoulos, V. (2005). Antecedents of perceived quality in the context of internet retail stores. *Journal of Marketing Management, 21*(7/8), 669-700.

[9] Guzzo, R. A., & Dickson, M. W. (1996). Teams in organizations: Recent research on performance and effectiveness. *Annual Review of Psychology, 47*(1), 307-338.

[10] Institute of Electrical and Electronics Engineers. (1990). *IEEE standard glossary of software engineering terminology* (IEEE Std 610.12-1990). New York, NY: Author.

[11] Kujala, S. (2003). User involvement: A review of the benefits and challenges. *Behaviour and Information Technology, 22*(1), 1-16.

[12] Larman, C. (2004). *Agile and iterative development: A manager's guide*. Boston, MA: Pearson Education.

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

[14] Millington, D., & Stapleton, J. (1995). Developing a RAD standard. *IEEE Software, 12*(5), 54-56.

[15] Palmer, S. R., & Felsing, J. M. (2002). *A practical guide to feature driven development*. Upper Saddle River, NJ: Prentice Hall.

[16] Pavlicek, R. C. (2000). *Embracing insanity: Open source software development*. Indianapolis, IN: Sams Publishing.

[17] Rico, D. F. (2007). *Effects of agile methods on website quality for electronic commerce*. Unpublished doctoral dissertation, University of Maryland University College, Adelphi, MD, United States. (http://davidfrico.com/rico07k.pdf)

[18] Schwaber, K. (1995). Scrum development process. *Proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages, and Applications (OOPSLA 1995), Austin, Texas, USA*, 117-134.

[19] 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.

[20] Takeuchi, H., & Nonaka, I. (1986). The new product development game. *Harvard Business Review, 64*(1), 137-146.

[21] Wolfinbarger, M., & Gilly, M. C. (2003). Etailq: Dimensionalizing, measuring, and predicting etail quality. *Journal of Retailing, 79*(3), 183-198.

[22] Zhao, L., & Deek, F. P. (2004). User collaboration in open source software development. *Electronic Markets, 14*(2), 89-103.