What is the Return on Investment (ROI) of Agile Methods?

by David F. Rico

Abstract

The purpose of this paper is to investigate the return-on-investment (ROI) of agile methods. Agile methods are new product development processes for creating software-based goods and services. Agile methods are a lightweight alternative to traditional methods based on sequential product development processes created over the last four or five decades. The use of traditional methods is theorized to result in higher quality software products because of well-documented customer requirements and products that exhibit fewer problems over their life cycle. Agile methods on the other hand are used to achieve higher customer satisfaction and product quality through rapid implementation and early market testing. The ROI of agile methods is yet to be fully explored because of their newness, while the ROI of traditional methods is well-understood. Therefore, the purpose of this paper is to investigate and summarize the literature on the ROI of agile methods. These results show that the use of agile methods results in increased cost-effectiveness, productivity, quality, cycle-time reduction, and customer satisfaction ranging from 10% to 100%.

1. Introduction

An agile method is a contemporary new product development process for creating computer software such as operating systems, middleware, applications, and web-based technologies. A new product development process is a streamlined management and development methodology for quickly and efficiently creating innovative goods and services. New product development processes span the life cycle of a novel technology from its inception, concept, or idea-stage right on through its operations, market, or end-user stage. Agile methods are generally characterized by lightweight, informal, and highly-adaptable new product development processes. Agile methods are rooted in concept testing, rapid prototyping, and early market feedback, and are lightweight forms of overlapping, cross-functional, simultaneous, integrated-product, time-based, and concurrent development. Agile methods are also rooted in chaos theory, systems theory, systems thinking, systems dynamics, double-loop learning, learning organizations, organizational learning, and adaptable systems. Agile methods evolved directly from their earlier traditional cousins.

2. Agile Methods

Agile methods are different from traditional software development methods. Traditional methods are also product development processes based on the theory of sequentially building software goods and services. Traditional methods are a sequential product development process of systems planning, analysis, architecture, design, development, and testing. Traditional methods are based on rigidly defined policies, processes, procedures, documentation, and tools with rigid interfaces between them. Traditional methods are based on the theory that a customer's or end-user's requirements can be defined at the beginning of the process, sequentially transformed into a software product, and then delivered to the customer when it is complete. A traditional method may take years or even decades to cycle through its process. Traditional methods generally fail to deliver a product that satisfies its customer's requirements. Agile methods, on the other hand, may be better than traditional methods because they may be used to achieve customer satisfaction and software quality by soliciting customer feedback on a series of rapid product releases. Agile methods are based on four broad processes of using iterative development, customer feedback, small software development teams, and flexible software technologies. Here are some of the major forms of agile methods [14].

2.1 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 [28]. 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.

2.2 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 [27]. 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.

2.3 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" [5]. 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.

2.4 Scrum

In 1993, Jeff Sutherland of the Easel Corporation adapted the principles from the "new product development game" [28] to the field of computer programming management, explicitly calling it "scrum" [26]. 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.

2.5 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 [16]. 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.

2.6 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 [6]. 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 [27].

2.7 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 [17]. 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.

2.8 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 [4]. 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" [12]. Another definition stated "open source software is labeled with free source, fast evolution, and extensive user collaboration" [31]. 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 [9]. One author wryly mused, "Internet time refers to something much faster, revolutionary, and more basic—It describes the process of developing open source software" [18].

2.9 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" [7]. 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.

2.10 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" [15]. 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.

2.11 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 [3]. 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.

3. ROI of Agile Methods

The purpose of ROI studies is to illustrate the business-value of using agile methods. There is the notion of soft-side ROI and hard-side ROI. Soft-side ROI refers to qualitative benefits such as improved morale or attitudes towards agile methods. While this is a legitimate form of ROI, this paper examines hard-side ROI. Hard-side ROI refers to the quantitative benefits of agile methods, often expressed in economic terms. For instance, if the use of agile methods takes half the time of traditional methods, then there is a direct economic benefit to increased productivity. That is, using agile methods may cost half as much as traditional methods. Below are only 11 major studies on the ROI of agile methods. There are many studies of agile methods ROI not mentioned here. However, these studies only looked at one or two techniques of agile methods such as pair programming. It is the intent of this paper to investigate the ROI of using agile methods in their entirety. In other words, what is the ROI of iterative development, early customer feedback, small teams, and flexible software technology?

3.1 Harvard Business School

In 1998, two management scholars from the Harvard Business School conducted a survey of 391 respondents to test the effects of flexible versus inflexible product technologies [29]. What they found was that projects using inflexible product technologies required over two times as much engineering effort as flexible product technologies (e.g., 17.94 vs. 8.15 months).

3.2 Harvard Business School

In 1998, a management scholar from the Harvard Business School conducted a survey of 29 projects from 15 U.S. Internet firms to test the effects of flexible software development management approaches on website quality [15]. What he found was that flexible product architectures and customer feedback on early beta releases were correlated to higher levels of website quality.

3.3 Boston College Carroll School of Management

In 1999, two management scholars from Boston College's Carroll School of Management conducted a case study of 28 software projects to determine the effects of iterative development on project success [10]. What they found was that software projects that use iterative development deliver working software 38% sooner, complete their projects twice as fast, and satisfy over twice as many software requirements.

3.4 Reifer Consultants

In 2003, Reifer Consultants conducted a survey of 78 projects from 18 firms to determine the effects of using agile methods to manage the development of software [20]. What they found was that 14% to 25% of respondents experienced productivity gains, 7% to 12% reported cost reductions, and 25% to 80% reported time-to-market improvements.

3.5 Shine Technologies

In 2003, Shine Technologies conducted an international survey of 131 respondents to determine the effects of using agile methods to manage the development of software [13]. What they found was that 49% of the respondents experienced cost reductions, 93% of the respondents experienced productivity increases, 88% of the respondents experienced quality increases, and 83% experienced customer satisfaction improvements.

3.6 CIO Magazine

In 2004, CIO Magazine conducted a survey of 100 information technology executives with an average annual budget of $270 million to determine the effects of agile management on organizational effectiveness [19]. What they found was that 28% of respondents had been using agile management methods since 2001, 85% of the respondents were undergoing enterprise wide agile management initiatives, 43% of the respondents were using agile management to improve organizational growth and market share, and 85% said agile management was a core part of their organizational strategy.

3.7 Digital Focus

In 2006, Digital Focus conducted a survey of 136 respondents to determine the effects of using agile methods to manage the development of software [8]. What they found was that 27% of the respondents were adopting agile methods for a project, 23% of the respondents were adopting agile methods company wide, 51% of the respondents wanted to use agile methods to speed up the development process, 51% of the respondents said they lacked the skills necessary to implement agile methods at the project level, 62% of the respondents said they lacked the skills necessary to implement agile methods at the organization level, and 60% planned on teaching themselves how to use agile methods.

3.8 Version One

In 2006, Version One conducted an international survey of 722 respondents to determine the effects of using agile methods to manage the development of software [30]. What they found was that 86% of the respondents reported time-to-market improvements, 87% of the respondents reported productivity improvements, 86% of the respondents reported quality improvements, 63% of the respondents reported cost reductions, 92% of the respondents reported the ability to manage changing priorities, 74% of the respondents reported improved morale, 72% of the respondents reported risk reductions, 66% of the respondents reported satisfaction of business goals, and 40% were using the scrum method.

3.9 AmbySoft 2006

In 2006, Ambysoft conducted an international survey of 4,232 respondents to determine the effects of using agile methods to manage the development of software [1]. What they found was that 41% of

organizations were using agile methods; 65% used more than one type of agile method; 44% reported improvements in productivity, quality, and cost reductions; and 38% reported improvements in customer satisfaction.

3.10 AmbySoft 2007

In 2007, Ambysoft conducted another international survey of 781 respondents to further determine the effects of using agile methods to manage the development of software [2]. What they found was that 69% of organizations had adopted agile methods, 89% of agile projects had a success rate of 50% or greater, and 99% of organizations are now using iterative development.

3.11 UMUC

In 2007, a student at the University of Maryland University College (UMUC) conducted a survey of 250 respondents to determine the effects of using agile methods on website quality [21], [22], [24], [25]. What he found was that: (a) 70% of all developers are using many if not all aspects of agile methods; (b) 79% of all developers using agile methods have more than 10 years of experience; (c) 83% of all developers using agile methods are from small to medium-sized firms; (d) 26% of all developers using agile methods have had improvements of 50% or greater; (e) developers using all aspects of agile methods produced better e-commerce websites.

Year	Source	Findings	Responses
1998	Harvard (Thomke et al., 1998)	50% reduction in engineering effort 55% improvement in time to market 925% improvement in number of changes allowed	391
1998	Harvard (MacCormack, 1998)	48% productivity increase over traditional methods 38% higher quality associated with more design effort 50% higher quality associated with iterative development	29
1999	Boston College (Fichman et al., 1999)	38% reduction in time to produce working software 50% time to market improvement 50% more capabilities delivered to customers	28
2003	Reifer Consultants (Reifer, 2003)	20% reported productivity gains 10% reported cost reductions 53% reported time-to-market improvements	78
2003	Shine Technologies (Johnson, 2003)	49% experienced cost reductions 93% experienced productivity increases 88% experienced customer satisfaction improvements	131
2004	CIO Magazine (Prewitt, 2004)	28% had been using agile methods since 2001 85% initiated enterprise-wide agile methods initiatives 43% used agile methods to improve growth and marketshare	100
2006	Digital Focus (Digital Focus, 2006)	27% of software projects used agile methods 23% had enterprise-wide agile methods initiatives 51% used agile methods to speed-up development	136
2006	Version One (Version One, 2006)	86% reported time-to-market improvements 87% reported productivity improvements 92% reported ability to dynamically change priorities	722
2006	AmbySoft (Ambler, 2006)	41% of organizations used agile methods 44% reported improved productivity, quality, and costs 38% reported improvements in customer satisfaction levels	4,232
2007	AmbySoft (Ambler, 2007)	69% of organizations had adopted agile methods 89% of agile projects had a success rate of 50% or greater 99% of organizations are now using iterative development	781
2007	UMUC (Rico, 2007)	70% of developers using most aspects of agile methods 26% of developers had improvements of 50% or greater Agile methods are linked to improved website quality	250

4. Conclusion

The purpose of this article was to examine and identify the ROI of agile methods. More specifically, its purpose was to identify the ROI of using agile methods in their entirety, not just some of the tools and techniques of agile methods like pair programming. This paper looks at the ROI of using all of the major factors of agile methods including iterative development, early customer feedback, small software development teams, and flexible software technologies that enhance productivity. The studies examined here identify many hard-side ROI benefits for using agile methods. The benefits of using agile methods range from 10% to 100% for increased cost-effectiveness, productivity, quality, cycle-time reduction, and customer satisfaction. The use of agile methods as a new product development approach does result in increased ROI. This begins to dispel the notion that agile methods result in lower ROI than traditional methods. However, it is important to note that these are only early studies and further study of the ROI of agile methods is necessary to make better conclusions. Promising new studies are starting to emerge based on more sophisticated approaches to measuring the ROI and more extensive historical data [11], [23].

5. References

[1] Ambler, S. W. (2006). Agile adoption rate survey: March 2006. Retrieved September 17, 2006, from http://www.ambysoft.com/downloads/surveys/AgileAdoptionRates.ppt

[2] Ambler, S. W. (2007). Agile adoption survey: March 2007. Retrieved July 23, 2007, from http://www.ambysoft.com/downloads/surveys/AgileAdoption2007.ppt

[3] 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.

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

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

[6] 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.

[7] 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.

[8] Digital Focus. (2006). Agile 2006 survey: Results and analysis. Herndon, VA: Author.

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

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

[11] Fichman, R. G., Keil, M., & Tiwana, A. (2005). Beyond valuation: Options thinking in IT project management. California Management Review, 40(2), 74-96.

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

[13] Johnson, M. (2003). Agile methodologies: Survey results. Victoria, Australia: Shine Technologies.

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

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

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

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

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

[19] Prewitt, E. (2004). The agile 100. CIO Magazine, 17(21), 4-7.

[20] Reifer, D. J. (2003). The business case for agile methods/extreme programming (XP). Proceedings of the Seventh Annual PSM Users Group Conference, Keystone, Colorado, USA, 1-30.

[21] Rico, D. F. (2007). Effects of agile methods on electronic commerce: Do they improve website quality? Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), Waikaloa, Big Island, Hawaii.

[22] 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/rico07.pdf).

[23] Rico, D. F. (2007). ROI of technology readiness assessments using real options: An analysis of gao data from 62 u.s. dod programs. Retrieved November 28, 2007, from http://davidfrico.com/rico07o.pdf

[24] Rico, D. F. (2008). Effects of agile methods website quality for electronic commerce. Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), Waikaloa, Big Island, Hawaii.

[25] Rico, D. F., Sayani, H. H., Stewart, J. J., & Field, R. F. (2007). A model for measuring agile methods and website quality. TickIT International, 9(3), 3-15.

[26] 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.

[27] 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.

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

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

[30] Version One. (2006). The state of agile development. Apharetta, GA: Author.

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