

Agile Savings versus Traditional Methods

By Brian Lucas

With excerpts taken from, “Is Agile a Fad or an Evolution”

By Brian Lucas

Time-to-market savings realized by agile methods comes from many different aspects of agile's natural mechanisms and flows. Most of these have a combined effect that contributes even greater economies. To begin with let us look at two aspects of agile. The first is the work unit basis of small self-contained product backlog items that are developed, tested and released in discrete time boxes. The second is the serious application of paring down the Pareto Principle to prioritize this development.

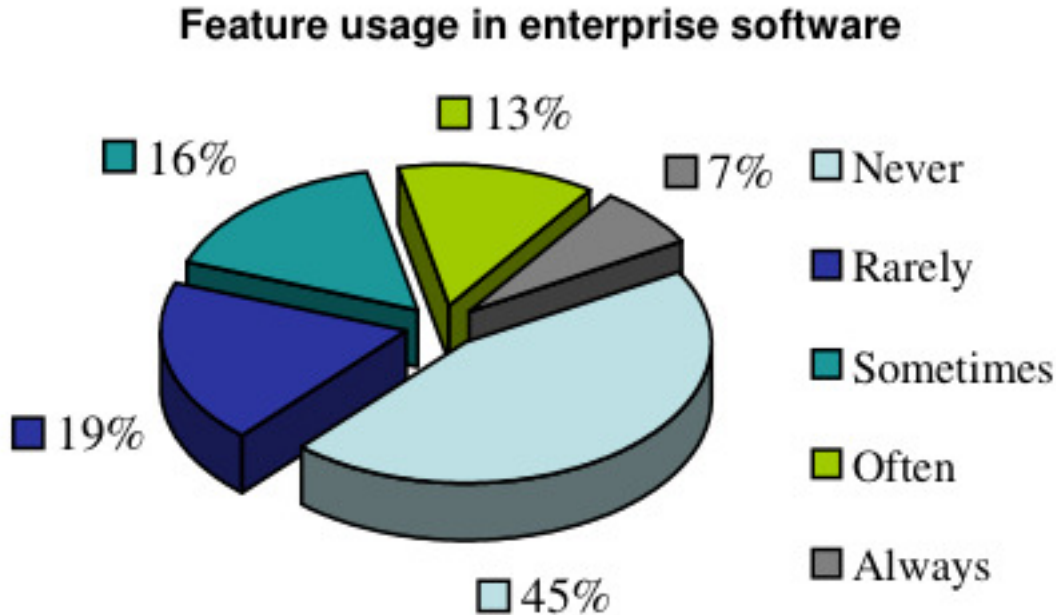
With the much improved flow over traditional processes, agile teams increase their productivity working as a dedicated team on a single project for short term iteration cycles. There are vast reductions in waiting for someone upstream in the process to feed you work, switching between tasks because of a lack of dedication, time lost creating, explaining and interpreting written specifications. Development teams utilizing Agile practices were on average 37% faster delivering their software to market and increased their teams' productivity by at least 16% According to a QSM Associates study.

The biggest cost driver for software development is employee salaries. The best way to control this cost is to only have associates develop features and functions that are actually needed. The focus of agile of continuous develop-release-feedback in short iterations forces the development concentration to remain on the most important business needs. This is a great benefit of agile; the emphasis on simple solutions. All this tight team work and utilization of previous results to guide new work and direction directly results in a continually refined focus on what is most important and avoidance on trivia and features that lack serious consequence.

It's a maxim that 80% of the effect is derived from 20% of the cause. This is the Pareto Principle. It is named for the Italian economist Vilfredo Pareto who noted in 1906 that 80% of the land in Italy was owned by 20% of its inhabitants. The Pareto Principle has wide application in business, software development, and product design. For example, most companies realize 80% of their revenue from 20% of their customers. You will get 80% of your web traffic from only 20% of your keywords. In software development as well, 80% of the features that customers request, they never use.

The noted information technology consulting firm, The Standish Group, has documented over the past decade the breakdown of features requested in enterprise software versus the ones customers actually

use. The Pareto Principle is alive and well once again as you can see in the pie chart below.



Agile helps you focus on the key 20% saving your organization time and money, by enforcing simple, standalone user stories supported by a common understanding throughout the whole team.

What kind of savings numbers does this actually generate? Well, Agile's benefits and saving can be quite profound, capable of showing a cumulative 10 to 1 return on investment and reduction in the risk of scope creep ranging from 80% to 10%. Specifically, agile has shown in two recent large scale surveys to:

- ✓ **Improved time to benefit by 69%**
- ✓ **Reduced cost by 57%**
- ✓ **Reduced effort by 62%**
- ✓ **Reduced critical defects by nearly 80%**
- ✓ **Reduced overall defects by more than 60%**

- Forrester Research

- ✓ **91% better schedule**
- ✓ **97% better productivity**
- ✓ **400% better satisfaction**
- ✓ **470% better ROI**

- Dr. David Rico PMP, CSM

But this is the results of only two pro-agile studies; what has the long haul of historical research shown over the last decade? The chart below shows a long history of agile measurements from diverse sources:

Year	Source	Findings
1998	Harvard (Thomke et al., 1998)	50% reduction in engineering effort 55% improvement in time to market 925% improvement in number of changes allowed
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
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
2003	Reifer Consultants (Reifer, 2003)	20% reported productivity gains 10% reported cost reductions 53% reported time-to-market improvements
2003	Shine Technologies (Johnson, 2003)	49% experienced cost reductions 93% experienced productivity increases 88% experienced customer satisfaction improvements
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
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
2006	Version One (Version One, 2006)	86% reported time-to-market improvements 87% reported productivity improvements 92% reported ability to dynamically change priorities
2006	AmbySoft (Ambler, 2006)	41% of organizations used agile methods 44% reported improved productivity, quality, and costs 38% reported improvements in customer satisfaction levels
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
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

These show that the productivity and savings numbers of practioners working in agile are very definitive. These historical studies span many different organizations from 1998 through 2007 show the tremendous success of Agile in business and industry as well as government.

Finally, I would point out some additional advantages of agile:

- 1) Change which is embraced in agile and resisted in waterfall has very significant impact. The cost of change in agile does not grow as in waterfall. It is dealt with constantly and made upon a smaller and more flexible code base. In fact most major code rewrites are avoided before they ever occur. It is a fact that it is much cheaper to introduce changes relatively late in an agile project. The constant refactoring, test automation safety net, continuous integration and splitting the work into stories allocated into iterations makes this savings a natural outcome.
- 2) Feedback from previous iterations keeps things predictable and based in reality. Actual data on the project's performance, not plans and wishful thinking is used to understand and estimate effort.
- 3) Design and codes defects are discovered at the earliest possible moment because of frequent builds and early releases.

With agile being used successfully from everything from Thomas Edison's Menlo Park to the Apollo Space Program to Lockheed's SR-71 Blackbird Program and with companies like Google and 66% of the world's projects using Agile Methods; Agile's is an unqualified success. Can you afford not to bring agility to your organization?