

Is there data to suggest that agile has had a positive impact on performance; e.g. productivity, quality, time to market?

December 2013

Scope of this Report

This report seeks evidence that agile software development methods have had a positive impact on traditional performance metrics such as productivity, quality and time-to-market. We consider the assertion that this is not a fair question because, for agile, customer satisfaction is the primary metric. We also consider the possibility that the reported data is skewed by not considering the failure rates of agile projects and/or self-serving optimism from the large and vociferous agile coaching/consulting community (in which we must include ourselves). Finally, we provide the data.

Whenever a DCG Trusted Advisor Report question starts with, “Is there data ...”, a long list of sources can be expected and this report is no exception. In an effort to be concise, we have not explicitly referenced all of the sources that we have listed but we have read them all and have included them to allow our readers to explore their own questions a little more deeply.

Is this even a useful question?

Anu Kendry, an Agile Coach at Freelance, presenting in February 2013 argues that there are some metrics that should simply be avoided in agile projects (see also blog post by “Paul” in Sources). She lists the following:

- Velocity comparisons
- Productivity e.g. Story Points per day
- Effort Estimation accuracy
- Number of stories completed
- LOC/Function Points
- Code quality

Seriously? Kendry presents a pretty robust set of alternative metrics. However, there is no avoiding her core argument that our traditional metrics are not appropriate for agile projects. How then should we compare traditional methodologies with agile? For many agile advocates, the answer is that we should not. Primarily this is because of the philosophy behind agile which prioritizes customer satisfaction over just about every other metric. In other words, productivity is not important if the customer gets what they want. This is embodied in the Agile Manifesto as “Customer collaboration over contract negotiation” and “Responding to change over following a plan.”

Whether this is justifiable depends on your viewpoint. At DCG, we believe that “what they want” implicitly includes “when they want it”. In turn, this implies that the customers satisfaction depends in

part on delivery date and, hence, productivity and quality. This more nuanced perspective, which includes other stakeholders in addition to the developers, is acknowledged in the less well known (and later) extension to the Agile Manifesto, the Declaration of Interdependence (DoI) signed, among others, by many of the Agile Manifesto authors. The DoI includes assertions such as “We boost performance through group accountability for results and shared responsibility for team effectiveness.”

We agree that if the agile definition of productivity is “Story Points per Day” then this is not a useful metric because story points are not standardized. The same is true for productivity metrics based on lines of code (see Capers Jones’ paper referenced in Sources). For example, we suggest that the productivity improvement due to refactoring reported by Moser et al, by using LOC/effort as the productivity metric, might be confusing a reduction in overall LOCs through refactoring with an increase in productivity generated by the agile process. The best (though itself not perfect) size metric for measuring software development productivity is function points. Unfortunately, Kendry rejects these too.

The Data

Failure

Before we consider reported improvements in productivity and the other metrics, we must acknowledge an assumption: Agile project reporting productivity and other data have been completed successfully. Unfortunately, such projects are only a subset of all agile projects because, like all software projects, many fail.

For context, looking at all IT projects, in 2012 McKinsey & Company in conjunction with the University of Oxford reported a study of 5,400 large-scale IT projects (projects with initial budgets greater than \$15M). Among the key findings quoted from the report:

1. 17 percent of large IT projects go so badly that they can threaten the very existence of the company
2. On average, large IT projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted

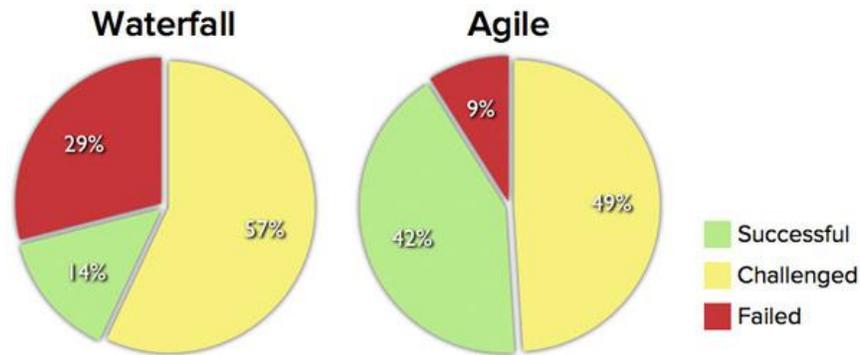
Looking specifically at agile projects, in 2007 and 2011 (update), Dr Dobbs Journal surveyed 586 respondents and found:

1. 70% of respondents had been involved in a project they knew would fail right from the start
2. Success rates for Agile projects 72%, success rate for traditional approaches 63%

The 2011 CHAOS report from the Standish Group included the comparison of software project success and failure rates shown in Figure 1. As ever with such reports, the reader can take from it what they choose. Mike Cohn reported that agile projects are successful three times more often than non-agile projects – this is true: 42% versus 14%. However, while agreeing with Cohn on the benefits of agile, Headspring Labs have a different view of the same data,

“The 2012 report is still fairly damning of the industry as a whole. It cites 12% success rate for waterfall projects and 42% success rates for agile projects. In 2012, 9% of agile projects still out-right fail, and 49% are “challenged”. i.e. they unacceptably ran over budget and/or over schedule – either can have significant impacts on the business.”

Figure 1 – Comparison of Software Project Success Rates - 2011 CHAOS report from the Standish Group



Source: The CHAOS Manifesto, The Standish Group, 2012.

Anecdotally, we are starting to hear similar concerns from agile “guru’s”. It is hard to find published material but in private conversations and at conferences, there is talk of too many agile initiatives failing.

Productivity

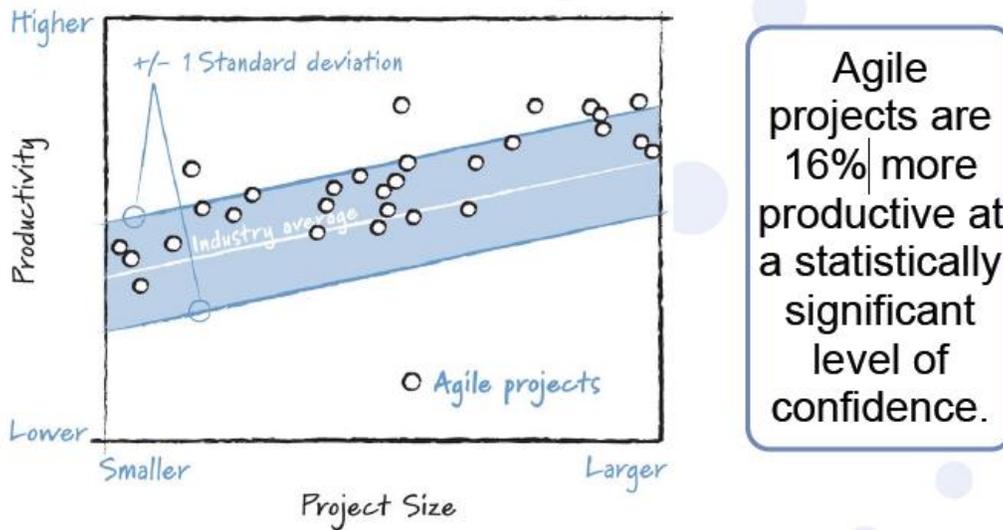
In 2006, David Garmus of DCG reviewed the projects in DCG’s Industry project Data based and produced a comparison of projects using agile methodologies to projects using traditional methodologies for similar technology environments (see Figure 2). This was a useful early snapshot but its limited scope (small projects) might be expected to favor agile methods over traditional methods.

Figure 2 - Average Hours/Function Point of Recent Enhancement Projects Across Different Platforms from DCG Database for Small Projects, 2006

Technology	Traditional Average Hrs/FP	Agile Average Hrs/FP	Apparent Improvement
Client Server	6.5	4.2	35%
Mainframe	8.1	7.0	14%
Web	4.8	3.2	33%
E-business web	6.6	5.8	12%

It turned out that DCG’s findings were broadly consistent with the findings reported by Michael Mah in 2008 (Figure 3).

Figure 3 – Michael Mah’s research on Agile Project Productivity (2008)



Source: Mah 2008.

Don Reifer takes a more nuanced view:

“Agile productivity, as measured in terms of output/unit of input, is higher than the norms experienced on plan-driven projects for delivered products. The gains experienced range from nominal to 50 per cent over the course of ten years, averaging at best between a 10 to 20 per cent improvement in a single year after adoption. However, these gains vary greatly by application domain and are a function of many factors (workforce composition, product complexity, size of project, etc).”

Productivity and Quality

Experienced software engineers and managers will not be satisfied by such findings because they will argue, correctly, that any project team can achieve higher productivity if they abandon quality (although both the DCG and Mah numbers above were for completed projects which implies at least a basic level of quality was achieved). In 2008, VersionOne and David Rico independently surveyed both productivity and quality (see Figure 4). Arguably, such surveys must be treated skeptically because respondents tend to overstate success and not report failure. However, the similar findings of the independent surveys somewhat counter-balance this reasonable skepticism.

For Reifer, there is not a positive impact of agile on quality,

“Agile quality, as measured in terms of defect densities post-release, is between 2 to 8 percent lower than that being experienced on traditional plan-driven projects, with quality at delivery off by as much as 20 to 30 percent. Again, quality varies greatly by application domain and is a function of many factors including those revolving around quality control and assurance practices.”

Figure 4 – Improvements through Agile reported in surveys by VersionOne (Interviews) and David Rico (Literature) in 2008

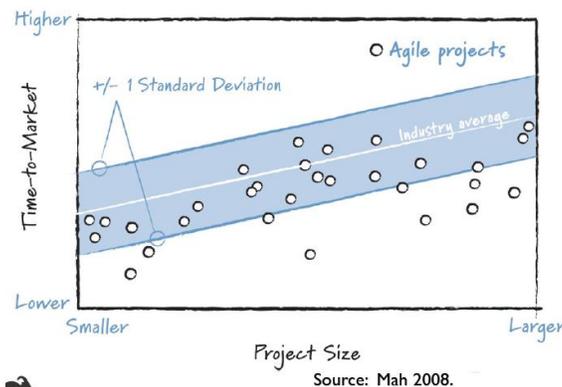
	VersionOne	David Rico
Productivity	<ul style="list-style-type: none"> • 23% said productivity was significantly improved • 50% said productivity was improved 	<ul style="list-style-type: none"> • Lowest Reported Improvement of 14% • Median Reported Improvement of 88% • Highest Reported Improvement of 384%
Quality	<ul style="list-style-type: none"> • 44% said quality had improved • Another 24% said quality improved significantly • 84% said defects had gone down by 10% or more • 30% said defects were down by 25% or more 	<ul style="list-style-type: none"> • Minimum quality improvement of 10% • Median quality improvement of 63%

Time to market

Of the three traditional metrics that this report focuses on, we might expect that Time to Market would be the one where Agile champions would expect most improvement. Philosophically, one might expect that through an agile approach like SCRUM, customers would be more inclined to settle for “good enough” over “perfect” in the desire to get to market quickly. That said, while direct comparisons are difficult, our experience has been that for a given fixed (if this ever happens in waterfall) set of initial requirements, customer input after each sprint in agile mean that we don’t generally expect improvement in Time to Market through agile. There will always be tension in an agile deployment between the customer’s desire to get to market and their desire to add “just one more feature.” The reported data confirms the expectation:

- VersionOne study found:
 - 64% said time to market improved
 - Another 23% said it was significantly improved
- Michael Mah found(see Figure 5):
 - Agile projects have a 37% faster time to market at a statistically significant level of confidence

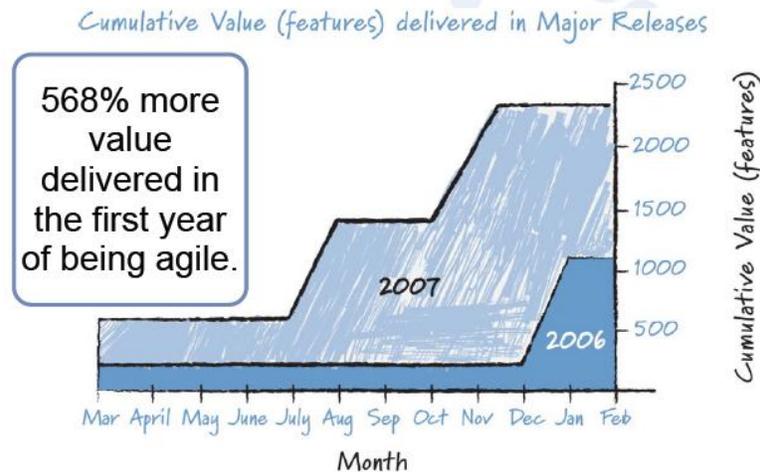
Figure 5 – Michael Mah’s research on Agile Project Time-to-Market (2008)



Value Delivered

A much quoted successful implementation of agile was that implemented by Salesforce.com in 2007. The key here is to note the focus on “Value delivered.” While there is a laudable increase in value delivered, the success at Salesforce.com does not necessarily reflect an increase in productivity of similar magnitude although some increase in productivity can safely be assumed.

Figure 6 - Salesforce.com Agile Implementation (Source: Green & Fry 2008)



Conclusion

There is data to suggest that agile has had a positive impact on performance e.g. productivity, quality, time to market. However, there is also data to suggest that agile implementations can and do fail. The assertion that customer satisfaction trumps all other metrics can be valid in some scenarios but is not sufficient to justify the widespread adoption of agile that we have seen. Our conclusion is that performance gains are there to be had from agile implementations but those implementations must be set up with discipline and care for the benefits to be realized. Of course, this begs the question of whether the same is true of all projects and all methodologies – is the documented evidence of agile’s positive impact on performance simply a Hawthorne effect? On balance, we think not – the positive improvement from agile is there to be had if you go about it the right way.

Sources

- Presentation on Agile Metrics by Anu Kendry, an Agile Coach (amongst other things) at Freelance, presenting in February 2013
- Lessons learned by a software tester (Paul), <http://swtester.blogspot.com/2012/01/quality-agile-metrics.html>
- [... of refactoring on quality and productivity in an agile ...](#) - Moser et al
- “A short history of Lines of Code”, Capers Jones, <http://namcookanalytics.com/a-short-history-of-the-lines-of-code-loc-metric/>

- Manifesto for Agile Software Development, ©2001, Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas (<http://agilemanifesto.org/>)
- Declaration of Interdependence (DOI), ©2005 David Anderson, Sanjiv Augustine, Christopher Avery, Alistair Cockburn, Mike Cohn, Doug DeCarlo, Donna Fitzgerald, Jim Highsmith, Ole Jepsen, Lowell Lindstrom, Todd Little, Kent McDonald, Pollyanna Pixton, Preston Smith and Robert Wysocki (www.pmdoi.org)
- McKinsey & Company in conjunction with the University of Oxford reported a study of 5,400 large-scale IT projects, [Delivering large-scale IT projects on time, on budget, and on value](#)
- Dr Dobbs Journal, [IT Project Success Rates Survey](#) (2007 version). Updated version 2011 – [2011 Project Success Rates](#)
- Scrum Alliance, Agile Project Reporting and Metrics, 29 July 2013, Saji Pillai, <http://www.scrumalliance.org/community/articles/2013/july/agile-project-reporting-and-metrics>
- Various Agile productivity reports by Don Reifer, available for purchase on the ISBSG website: <http://www.isbsg.com/collections/analysis-reports>
- <http://www.succeedingwithagile.com/wp-content/uploads/2009/09/Reported-Benefits-of-Agile.pdf>
- Mah, Michael. 2008. How agile projects measure up, and what this means to you. *Cutter Consortium Agile Product & Project Management Executive Report* 9 (9).
- Mann, Chris, and Frank Maurer. 2005. A case study on the impact of Scrum on overtime and customer satisfaction. In *Proceedings of the Agile Development Conference*, 70–79. IEEE Computer Society.
- Rico, David F. 2008. What is the ROI of agile vs. traditional methods? An analysis of extreme programming, test-driven development, pair programming, and Scrum (using real options). A downloadable spreadsheet from David Rico’s personal website. <http://davidfrico.com/agile-benefits.xls>.
- VersionOne. 2008. The state of agile development: Third annual survey. Posted as a downloadable PDF in the Library of White Papers on the VersionOne website. http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.
- [Empirical studies of agile software development: A ...](#) - Dybå - Cited by 702
- [On the Productivity of Agile Software Practices: An Industrial Case ...](#)
www.itu.dk/.../On%20the%20Productivity%20of%20Agile%20Software.by F Maurer
- Ambler, Scott. 2008a. Agile adoption rate survey, February. <http://www.ambyssoft.com/surveys/agileFebruary2008.html>.
- Cohn, Mike. 2009. *Succeeding with Agile: Software development using Scrum*. Addison-Wesley.
- Greene, Steve, and Chris Fry. 2008. Year of living dangerously: How Salesforce.com delivered extraordinary results through a “big bang” enterprise agile revolution. Session presented at Scrum Gathering, Stockholm. <http://www.slideshare.net/sgreene/scrumbgathering-2008-stockholm-salesforcecom-presentation>.
- “Why Projects Fail” – Links to reports on IT Project Failure at http://calleam.com/WTPF/?page_id=1445

- Mike Cohn blog post on 2011 CHAOS report from the Standish Group, <http://www.mountangoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>
- Headspring Labs blog post, <http://www.headspringlabs.com/blog/the-failed-record-of-the-software-industrywhy-software-people-dont-get-it/>
- Hawthorne Effect, http://en.wikipedia.org/wiki/Hawthorne_effect