Agile Software Development And The Factors That Drive Success

Successful Software Development Requires Much More Than “A Fool With A Tool”

September 2012
Table Of Contents

Executive Summary................................................................. 2
The Agile Revolution Is Playing At A Theater Near You! ................................................................. 3
The Successful Group Executed Agile Projects More Successfully ........................................................................ 4
The Successful Group Performed Traditional Projects Well ............................................................................ 6
The Successful Group Believes That Agile Improves These Results .............................................................. 8
Key Recommendations................................................................................................................................. 9
Appendix A: Methodology.................................................................................................................................... 10
Appendix B: Supplemental Material .............................................................................................................. 10
Appendix C: Demographics .......................................................................................................................... 11
Appendix D: Endnotes......................................................................................................................................... 12

© 2012, Forrester Research, Inc. All rights reserved. Unauthorized reproduction is strictly prohibited. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change. Forrester®, Technographics®, Forrester Wave, RoleView, TechRadar, and Total Economic Impact are trademarks of Forrester Research, Inc. All other trademarks are the property of their respective companies. For additional information, go to www.forrester.com. [1-K4ZZ9J]

About Forrester Consulting
Forrester Consulting provides independent and objective research-based consulting to help leaders succeed in their organizations. Ranging in scope from a short strategy session to custom projects, Forrester’s Consulting services connect you directly with research analysts who apply expert insight to your specific business challenges. For more information, visit www.forrester.com/consulting.
Agile software development is one tool in a vast toolbox. But a fool with a tool is still a fool.

Executive Summary

If Agile is the answer, what exactly is the question? If you believe that Agile software development is raining success on all of its acolytes, you are mistaken. Agile software development is more akin to a hammer and chisel: The quality and desirability of the products it yields depend on the experience and talent of the craftspeople wielding the tools.

The digital revolution continues to destroy firms that don’t anticipate market shifts quickly enough to leverage new opportunities and avoid new pitfalls. The need for business agility is driving the Agile movement in software development — and continuous improvement is at the heart of the movement.

Every software development team, Agile or not, wants to continually improve — but what does it mean to be better? Ask a thousand people what it means, and you’ll get several hundred different answers. We surveyed 112 IT professionals in traditional and emerging industries about their Agile and traditional application development habits and practices, how they measure success, and what their success rates are. We identified the 24 most successful firms and compared them with the remaining 88 firms, which functioned as the control group. Next, we analyzed the data to determine the common characteristics of the successful group: What Agile or traditional software development and project management practices and habits do they share that consistently yield better results? Our findings challenge some of the myths and mystique that have developed around the Agile movement.

Key Findings

Forrester’s study yielded a number of intuitive and counterintuitive data points that confirm or conflict with commonly held beliefs about the how Agile development tenets should guide the day-to-day behavior of development team members:

- **Agile tenets and techniques help the best firms to be better.** All of our survey respondents applied the tenets of the Agile manifesto in their approach to application development — for example, limiting the amount of work in progress to reduce the impact of sudden business change. The net benefit to the successful group is that their “requirements grow stale awaiting coding” 81% less often than the control group.

- **But there is more to “success” than implementing a few Agile tenets and techniques.** Simply choosing and implementing some of the Agile tenets is no guarantee of success — many members of the control group claim to use story points, develop software using sprints, and deliver software continuously.

- **Most firms still have a long way to go to become more agile.** If Agile software development is a steppingstone on the path to increasing business agility, then most firms in our survey still have a long way to go. For example,

---

1 The term ‘control group’ does not meet the conventional definition for the purposes of this paper.
Agile espouses smaller, faster development cycles that deliver functioning software sooner, which in turn limits the total body of work in progress. This has the effect of testing hypotheses about required functionality much earlier than waterfall methods so that developers and business sponsors can quickly prove or disprove them. Yet 80% of all firms surveyed still cling to the iron-triangle measures of on-time, within budget, and full-scope delivery as their primary measures of success. Even the most successful firms are not really Agile; rather, they are practicing “water-Scrum-fall.”

The Agile Revolution Is Playing At A Theater Near You!

In response to demands for much greater business agility and dissatisfaction with the current pace of software delivery — often with the threat of outsourcing looming in the air — application development professionals are turning to Agile software development to break free of the chains of traditional big-bang and waterfall projects. Agile purists contend that this is a unique event in the history of software development, while the silver-haired software developers argue that it’s simply another step forward in its evolution — from spaghetti code to structured code to object-oriented development to “smaller, faster, and sooner is better.”

This paper avoids the polar extremes of those theatrical arguments — it neither ignores the historical perspective nor remains mired in it — and instead focuses on success measures that are meaningful to business leaders and app-dev pros alike. To that end, we surveyed 112 IT professionals about their Agile and traditional application development habits and practices, how they measure success, and what their success rates are. We split these 112 respondents into two groups — a “successful” group of 24 and a “control” group comprised of the remaining 88. How did we separate the two groups? To be “successful,” 70% or more of the projects at a respondent’s organization in the previous 12 months had to meet all three of the iron-triangle measures; on-time, full-scope, and within-budget. In contrast, just 21% of the control group’s projects met all three success criteria (see Figure 1).

Figure 1
The Successful Group Delivered 70% Or More Of Its Projects On Time, Within Budget, And With Full Scope

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012
Once we identified the challenge and problem statement with the successful group, we analyzed its data to understand which characteristics successful teams had in common — what practices and habits do they all use that consistently yield better results? Surely Agile techniques would figure prominently in our results? Yes — and no.

“Iron-Triangle Agile” And Other Counterintuitive Findings

Agile purists may be tempted to discount this survey because the measures of success it uses — on-time, on-budget, and on-scope — is the “iron triangle” measure used for traditional or waterfall projects, instead of the measures Agile development uses, like velocity and burndown rates. This criticism is valid to a degree — Agile efforts do not seek to precisely fix project cost, scope, and deadlines at the onset of the project, as not enough is known to have that level of precision and certainty.

However, the survey data clearly shows that project management practices and management scrutiny have yet to catch up to Agile practices: 80% of all firms use iron-triangle measures as their primary measure of success. Ninety-two percent of firms in the successful group use iron-triangle measures as the primary measure of success for both Agile and traditional projects. That’s 15% more often on Agile projects and 19% more often on traditional projects than the control group. We may be developing software leveraging Agile tenets, but the overall process looks more like a variation on water-Scrum-fall.

Many comparisons between the successful and control groups warrant our attention because they are unremarkable — the places where conventional wisdom suggests that Agile would vary greatly in fact varied by only a few to several percentage points. Within a few percentage points, members of both groups use visual planning boards such as Kanban and project boards; measure velocity and the cost of deliverables consistently; employ self-directing teams; employ specialized team members on Agile teams; assign team members to multiple projects; and claim that Agile affords the ability to deliver usable functionality more consistently and predictably.

The Successful Group Executed Agile Projects More Successfully

Counterintuitive measures notwithstanding, what else did our survey show about the habits of the successful group — what did they do differently than the control group?

The Successful Group Planned And Managed Agile Projects Differently

Whether they employed T-shirt sizes or other estimation methods, respondents in the successful group based their project sizing on their own history and employed some form of application life-cycle management (ALM) to automate the software development life cycle (SDLC), using each technique 14% more often than the control group. They also used tools to mitigate the issues of distributed teams more often, consistent measures of quality nearly one-third more often, and significantly involve business sponsors 46% more often than the control group (see Figure 2).
The Successful Group Executed Agile Projects Differently

The real meat of the differences, and where they really pay off, comes in some of the things the successful group did less often than the control group. The successful group clearly embraces the theory that limiting work in progress is good — its requirements sit awaiting coding 81% less often than the control group, its coded modules await testing 70% less often, and it holds off system testing until all coding is complete 47% less than the control group (see Figure 3).

Figure 3
Successful Companies Maximized The Flow Of Their Agile Projects

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012
The Successful Group Deployed Its Agile Projects Differently
The delivery end of the spectrum exposes a few more differences between the two groups: Operations groups were more apt to be in control of the final deployment and development staff was less likely to be manually involved in the final production deployment. The successful group also tended to deliver more incrementally and avoid big-bang deployments — another nod to Agile and continuous incremental improvement (see Figure 4).

The Successful Group Performed Traditional Projects Well
So was the successful group merely a conglomeration of Agile whiz kids who would choke on a large waterfall project? Hardly — these folks use some techniques for both Agile and traditional development — but, for the most part, they pulled different levers on traditional projects.

The Successful Group Did Some Things More Often On Traditional Projects
As with Agile projects, members of the successful group based project sizing on their own history — but in this case to a much larger degree (23%), no doubt owing to the fact that they have executed more traditional projects and have a larger base from which to draw. This group avoided change requests — conflicting with Agile tenets — 32% more often than the control group. This suggests more process rigor and better requirements development. Better requirements jibes with the next point — they use requirements to drive the estimates of successive SDLC phases and, in a nod to waterfall, they document all requirements up front 21% more than the control group (see Figure 5).
**The Successful Group Did Some Things Less Often On Traditional Projects**

As was the case with Agile projects, the real meat of the differences between the successful group and the control group becomes clear as we examine work in progress and the flow of work through the SDLC. The successful group clearly embraces the theory that limiting work-in-progress is good: its requirements sit awaiting coding 59% less often, coded modules await testing 43% less often, and tested modules sit on the shelf awaiting larger-bang implementation a whopping 63% less often than with the control group (see Figure 6).

**Figure 5**

Success With Waterfall Suggests Tighter Processes, Controls, And Estimates

![Figure 5](image_url)

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012

**Figure 6**

Successful Firms Paid Attention To Flow Even On Traditional/Waterfall Projects

![Figure 6](image_url)

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012
Even on traditional and waterfall projects, respondents in the successful group maximized project flow and manually deployed 40% less often than the control group.

**The Successful Group Believes That Agile Improves These Results**

We’ve established that the successful group performs markedly better on both Agile and traditional projects. But beyond their high success rates, what do they believe is better because of Agile? The successful group reported that:

- **Agile teams are 36% more productive higher than the control group.** This is no doubt owing to the tendency to dedicate team members to the task at hand and allowing the team to self-direct its activity.

- **Agile improves team morale 20% more than the control group.** This is a tribute to the “respect for people” tenet of the Agile manifesto, and likely a reflection that happy teams are productive teams.

- **Agile business sponsors are involved 47% more than the control group.** The concept of “product owners” is another key component of Agile.

- **Agile results in simple, flexible applications 62% more often than the control group.** Isn’t this where we started — the need to be more agile with respect to business needs?
KEY RECOMMENDATIONS

Agile is here, Agile is real, and — in direct refutation of headline-grabbing-pundits — Agile is far from dead. Agile is part of something bigger that can trace its historical roots much further back than the birth of the Agile manifesto to the works of Deming, Drucker, Taiichi Ohno, and other business and business-process luminaries. The trick for all of us is to apply it in context: according to the needs of our business leaders and at the fastest possible pace that our organizations can successfully absorb. While that sounds simple enough — these survey results show that “there be demons” in the detail of Agile execution and scale! What should you do?

• **Bask in Agile’s bright light, but don’t be the moth to Agile’s flame.** Every software movement since the dinosaur days of punchcards (or pre-Internet/pre-smartphone for younger people) has had its share of acolytes — many of whom over-rotated on a good idea, upset existing processes, and flirted with business disaster in the name of wholesale, seemingly magical technical improvement. Some of these characteristics fit the Agile movement’s most fervent supporters, who beseech us to abandon all in favor of sudden, complete change. This camp is well-intentioned but naïve — the existing application estate that now runs the backbone of the business cannot simply be swept away like 40 years of accumulated detritus.

• **Embrace the fact that inertia is a sure path to business extinction.** Equally naïve and dangerous are the technical Luddites who espouse that Agile is merely a passing fad that exposes the status quo to untenable risk. We need only look at firms in industries where digital disruption virtually killed businesses like Kodak, Blockbuster, Borders, and the Big Four music companies to see the real effect of untenable risk: extinction.

• **Avoid the “I have a hammer” syndrome — use Agile as a tool, not a religion.** Technical people love tools; as the saying goes, “If you’re a hammer, everything looks like a nail.” One underlying conclusion we can draw from this survey’s myriad data points is that the successful group wasn’t simply good at Agile — it was good at software development, whether that was run using traditional or Agile methods. They wield Agile like a tool in a toolbox, not as religious fervor — and they still have a lot of room for improvement.

• **Get the help you need to step up to the next level and handle second- and third-order impacts.** Changing any component in an ecosystem has a ripple effect throughout the pond. If you change how developers work, you have to change their incentives, compensation, governance, and interfaces to other groups such as operations and the PMO. Firms fail when they pull a lever in one area without regard to its impact elsewhere. You can use a trial-and-error discovery method or you can hire some experienced help to help guide you. Spend the money, avoid the trouble, and give yourself a fighting chance at success.
Appendix A: Methodology

In this study, Forrester surveyed 112 IT professionals about their Agile and traditional application development habits and practices, how they measure success, and what their success rates are. Survey participants included decision-makers in application development and IT executive roles. We split these 112 respondents into two groups — a “successful” group of 24 and a “control” group comprised of the remaining 88. To be “successful,” 70% or more of the projects at a respondent’s organization in the previous 12 months had to meet all three of the iron-triangle measures; on-time, full-scope, and within-budget. In contrast, just 21% of the control group’s projects met all three success criteria. We then analyzed the data to determine the common characteristics of the successful group: What Agile or traditional software development and project management practices and habits do they share that consistently yield better results? The study began in July 2012 and was completed in August 2012.

Appendix B: Supplemental Material

Related Forrester Research


“Determine If You’re Agile And Lean Enough,” Phil Murphy, Forrester Research, Inc., April 25, 2012


“Revamp Your Organization For Agile And Lean,” Phil Murphy, Forrester Research, Inc., January 30, 2012

“Increase Agile Efficacy To Improve Customer Value,” Tom Grant, Ph.D., Forrester Research, Inc., January 10, 2012
Appendix C: Demographics

Figure 7
Survey Respondent Job Functions Ran The Gamut Of Application Decision-Making Roles

Base: 112 IT professionals

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012

Figure 8
Survey Respondents Held Senior-Level Positions At Their Firms

Base: 112 IT professionals

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012
Figure 9
Survey Respondents Represented Several Key Industries, Including Commercial Software Development

<table>
<thead>
<tr>
<th>Industry</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturing</td>
<td>21%</td>
</tr>
<tr>
<td>Financial services, banking, and insurance</td>
<td>21%</td>
</tr>
<tr>
<td>Healthcare</td>
<td>13%</td>
</tr>
<tr>
<td>Commercial software development</td>
<td>11%</td>
</tr>
<tr>
<td>Other</td>
<td>11%</td>
</tr>
<tr>
<td>Government, education, and non-profit</td>
<td>8%</td>
</tr>
<tr>
<td>Transportation services and logistics</td>
<td>6%</td>
</tr>
<tr>
<td>Telecommunications services</td>
<td>5%</td>
</tr>
<tr>
<td>Retail</td>
<td>4%</td>
</tr>
<tr>
<td>Energy and utilities</td>
<td>3%</td>
</tr>
</tbody>
</table>

Base: 112 IT professionals

Source: A commissioned study conducted by Forrester Consulting on behalf of HP, August 2012

Appendix D: Endnotes

1 "Water-Scrum-fall" is a term that recognizes how many firms blend an approach of writing most requirements up front, using time-boxed sprints to code, and then falling back on traditional big-bang deployments. Source: "Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today," Forrester Research, Inc., July 26, 2011.