

January 20, 2010

Agile Development: Mainstream Adoption Has Changed Agility

by Dave West and Tom Grant, Ph.D.

for Application Development & Program Management Professional

January 20, 2010

Agile Development: Mainstream Adoption Has Changed Agility

Trends In Real-World Adoption Of Agile Methods

by **Dave West and Tom Grant, Ph.D.**

with Mary Gerush and David D'Silva

EXECUTIVE SUMMARY

In the past few years, Agile processes have not only gained increasing adoption levels; they have also rapidly joined the mainstream of development approaches. And while more organizations are adapting to Agile conventions, Agile is also adapting to the workplace. Perhaps the clearest sign of the mainstreaming of Agile is the abandonment of orthodoxy: Teams are puzzling out the mix of methodologies and combining them to fit within their organizational realities, blending Agile and non-Agile techniques and practices to create a hybrid methodology that fits larger organizations. Other changes, such as new team dynamics and the redefinition of roles such as the business analyst, show the genuine force behind Agile adoption. It's time for software development professionals to stop sitting on the fence where Agile is concerned. According to those who have successfully adopted Agile, the benefits are well worth the effort, and with the recent dramatic increase in Agile adoption, the probability of working in or with an Agile team has increased for everyone.

TABLE OF CONTENTS

2 Agile Adoption Goes Mainstream

10 Teams Are Changing To Support Agility

WHAT IT MEANS

17 Agile Becomes Mainstream — But Not Without Some Changes

RECOMMENDATIONS

18 App Dev Professionals Should Blend Agile Methods To Meet Their Own Needs

19 Supplemental Material

NOTES & RESOURCES

Forrester conducted three Agile-related surveys in 2009, all of which are referenced in this document. The results in this document are also based on 15 interviews with vendors, more than 30 briefings with vendors, 60 research-related interviews in the past year, and more than 60 inquiries related to Agile methods.

Related Research Documents

["From Agile Development To Agile Engagement"](#)
May 6, 2009

["Ensure Success For Agile Using Four Simple Steps"](#)

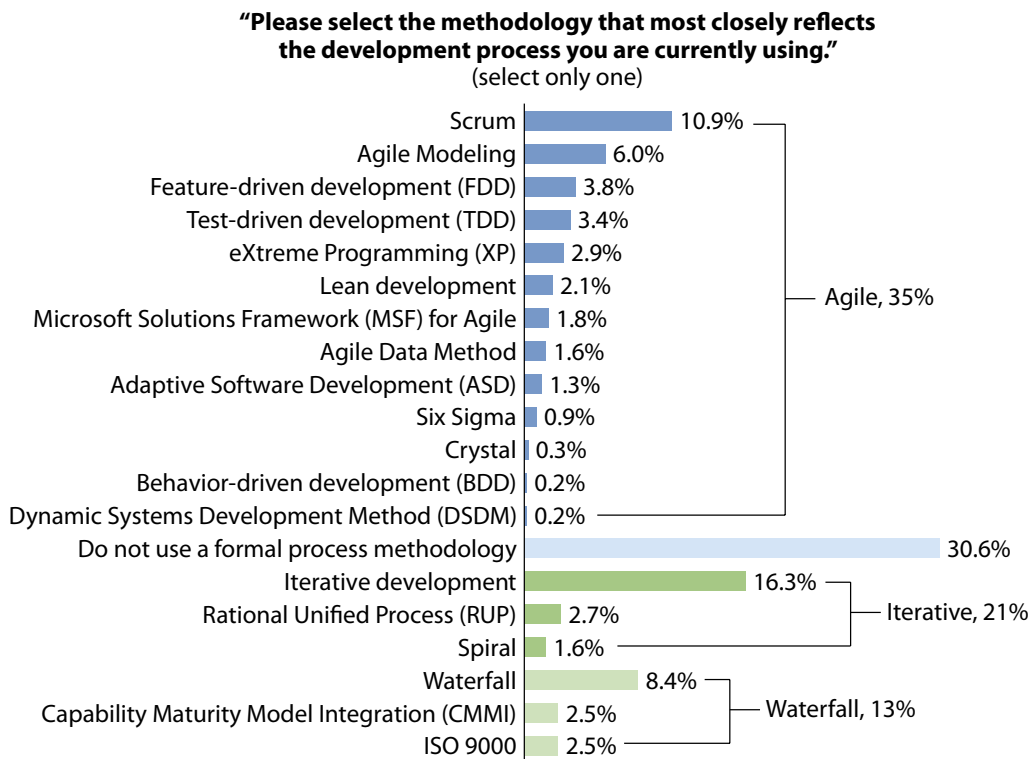
March 24, 2009

["Make Agile Lean To Boost Business Impact"](#)
December 17, 2008

AGILE ADOPTION GOES MAINSTREAM

Agile development methods have become very popular: In our recent Forrester/Dr. Dobbs Global Developer Technographics® Survey, Q3 2009, 35% of respondents stated that Agile most closely reflects their development process, with the number increasing to 45% if you expand what you include in Agile’s definition (see Figure 1). Both waterfall and iterative approaches are giving ground to much lighter, delivery-focused methods based on the principles the Agile Manifesto describes.¹ The older methods are not disappearing, however: 34% of the survey respondents stated that they continue to use either an iterative or waterfall development process as their primary method of software delivery.

Figure 1 Agile Adoption Has Reached Mainstream Proportions



Base: 1,298 IT professionals

Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

56100

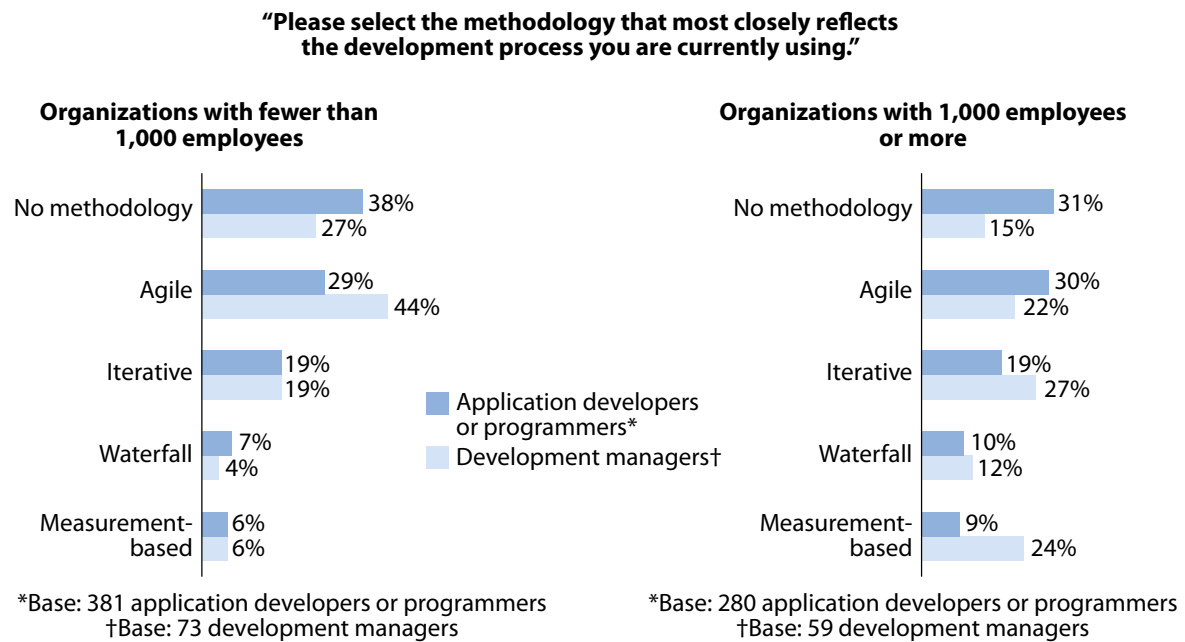
Source: Forrester Research, Inc.

Developers Still Prefer No Method, But Agile Comes In A Close Second

Agile methods encourage more-collaborative development than do traditional approaches, and many developers who have shied away from formal development methods in the past — believing them to be the province of “management” — have embraced Agile as a “formal” development process. A developer at a large financial company described the change as fundamental, explaining, “I was never interested in process before Agile, but now I read books on how to organize my work better and deliver better software.”

The best way to see how developers embrace Agile is to look at the number of developers in both large and small organizations who say they use Agile. In big and small organizations, developer adoption of Agile is at 30% and 29%, respectively (see Figure 2). While development managers in large organizations are using measurement-based and iterative approaches more often than Agile approaches, in smaller organizations Agile approaches are most popular with managers, followed by no methodology and then iterative.

Figure 2 Developers Couldn't Care Less About Methodology Unless It's Agile



(percentages may not total 100 because of rounding)

Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009

Scrum Leads As The Most Adopted Agile Methodology

When it comes to selecting an Agile methodology, Scrum is the overwhelming favorite.² Scrum focuses on *how* people work instead of on the work that they do, and it relies on the principles laid out in the Manifesto for Agile Software Development. Many Agile practitioners have adopted Scrum because:

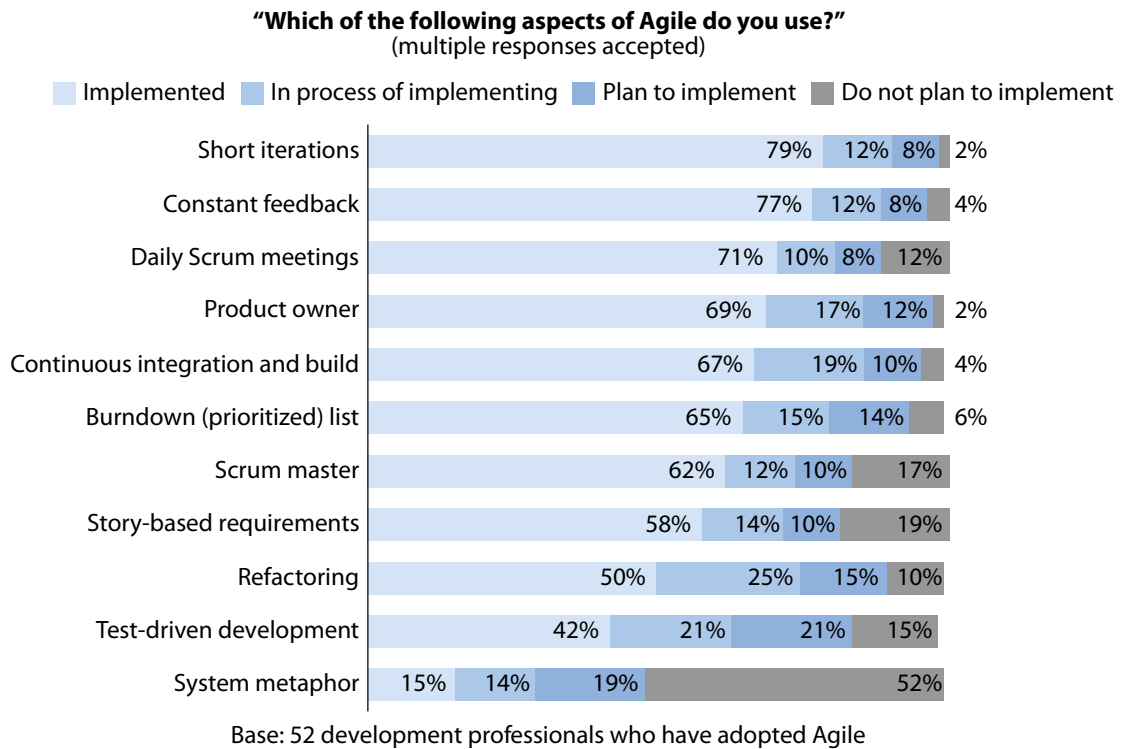
- **Scrum is simple.** At the heart of Scrum is a simple set of common-sense, easy-to-understand guidelines that describe the key roles, interactions, and artifacts required to deliver high-quality, high-value software. Scrum focuses on delivery and practice rather than activities and templates and allows the team to determine how it will undertake the tasks required to build the software.
- **Scrum is practical.** Scrum addresses the practicalities of organizing the team, allocating work, reviewing status, and correcting course. The heart of Scrum is the daily stand-up meeting, which the team uses to report progress and make decisions.
- **Scrum is popular.** When choosing a methodology, teams prefer an approach that others have adopted. As soon as Scrum began to emerge as the leader among Agile methods, teams began to see it as a “safer bet” than other Agile methodologies. At this stage in the evolution of Agile methodologies, success drives adoption.

What Does It Mean To Be Agile?

Research has shown that Agile adoption is not black and white, with teams being non-Agile one day and Agile the next. Scrum and other Agile methodologies prescribe a set of techniques that, when used collectively, are intended to lead to successful software development and delivery. However, teams do not usually implement all of these techniques simultaneously; even in the most mature Agile adoptions, teams pick the techniques that work best for them (see Figure 3). Agile techniques do not strictly depend on each other; teams can adopt daily Scrums before implementing continuous integration, and vice-versa.

Likewise, teams operating in traditional software delivery environments may adopt certain Agile techniques — for example, daily stand-up meetings or shorter iterations — to improve their outcomes. This variation in the adoption of Agile components indicates that teams are more concerned with making sure they are working well together and producing high-quality software than with changing their software engineering process.

Figure 3 Teams Embrace Different Components Of Agile Methodologies



Source: Q3 2009 Global Agile Adoption Online Survey

56100

Source: Forrester Research, Inc.

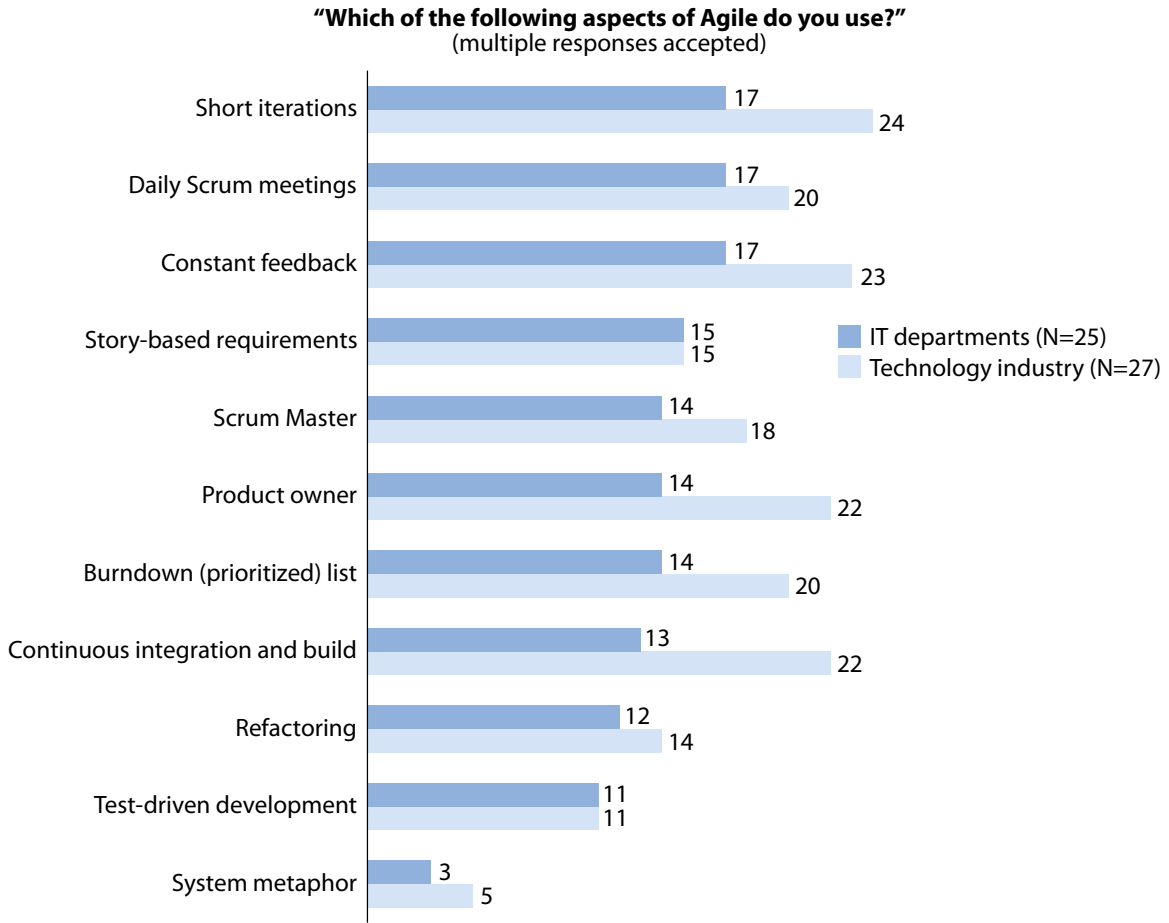
When Adopting Agile Techniques, Organizational Priorities Come First . . .

Teams do not work in isolation from the larger organization, so the choice of which Agile components an organization will adopt depends, in part, on organizational characteristics. For example, interviews and survey data show important differences in Agile adoption between technology industry firms — where the development team’s output is a product that the company sells — and IT departments in other types of organizations — where the software delivery output affects how the organization’s employees, partners, or customers work (see Figure 4). In general, we see that:

- **IT departments adopt techniques that improve project deliverables.** IT organizations struggle to get requirements right and test effectively, so they emphasize techniques — such as story-based requirements, daily Scrum meetings, and rapid feedback mechanisms — that help keep project deliverables on track to meet business needs.

- **Technology industry companies stress pace.** Tech industry development teams are under constant pressure to deliver new products and services, so Agile techniques related to speed, such as daily Scrum meetings and short iterations, naturally get priority.

Figure 4 Agile Adoption Varies Based On Organizational Priorities



Base: development professionals who have adopted Agile
Source: Q3 2009 Global Agile Adoption Online Survey

... And Engineering Priorities Quickly Follow

Aspects of Agile relevant to software engineering processes are as important as those related to organizational and team dynamics. In the adoption of Agile engineering techniques:

- **Continuous integration and build enhancements pay off quickly.**³ The Agile discipline of always checking in working code to ensure that nightly automated builds don't break is not easy to adopt. While traditional build processes with their manual steps and scripts are laborious and time-consuming, implementing nightly automated builds takes a dedicated effort. However, the continuous integration and build technique's relatively high adoption rate clearly indicates confidence that this investment pays off. This method catches defects early and often, as should be the case in an Agile environment.
- **Test-driven development remains an objective for many teams.**⁴ A large proportion of Agile teams aspire to build software by writing test cases first and then writing code to pass those tests, but they do not adopt this discipline as frequently as they adopt continuous integration and build. As an Agile team member in a large financial services company told us: "Test-driven development is easy for a function, or even a service, but for a story, it is hard. We don't have the environment, the data, or the right skills for a business analyst to hand me a test to develop from." In place of test-driven development, teams try other means — such as using code-coverage tools or pair-programming techniques — to catch issues early.
- **Teams often address architectural questions later, not sooner, through refactoring.**⁵ The age-old debate between the practitioners of architecture and Agile delivery continues to create friction and pain for Agile adopters.⁶ However, the very low rate at which Agile teams adopt a system metaphor and the informality with which they describe software do not mean that they are indifferent to architectural issues. The practice of refactoring gives Agile teams the opportunity to return to address architectural questions after they have developed software. Integrated development environment (IDE) features that encourage or simplify refactoring, along with impact reporting to gauge refactoring results, support practitioners in undertaking post-delivery architectural changes.

Mixed Processes Are A Reality

Perhaps the most important aspect of Agile's entrance into the mainstream is the way that teams pragmatically mix methodologies. Instead of sticking strictly to a particular Agile orthodoxy, teams cherry-pick Agile methods, often including non-Agile techniques in the mix as well (see Figure 5).

Interestingly, this heterodox attitude persists in even the most-mature Agile implementations. A recent survey on Agile adoption in the technology industry shows that as teams advance in their Agile maturity, their average amount of non-Agile techniques may decrease; however, their use of non-Agile techniques will not completely disappear (see Figure 6). Among the teams that cleave to a

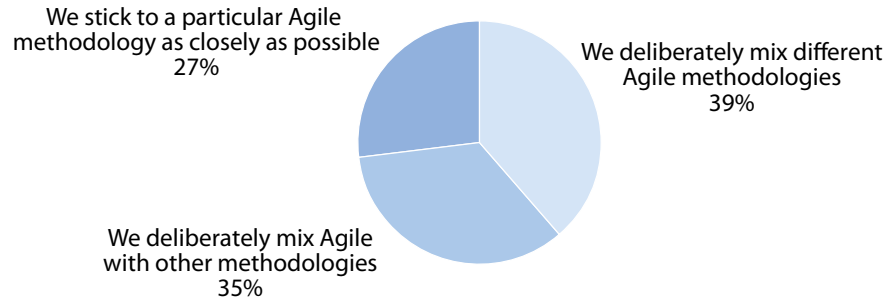
particular Agile methodology during the early and middle stages of adoption, a small percentage of them abandon this orthodoxy as their adoption reaches full maturity.

We see this trend in both technology industry organizations and IT departments in other industries: Most teams are not adopting Scrum, eXtreme Programming (XP), or another specific Agile approach but are embracing Agile as an ethos or philosophy and cherry-picking the best bits from many different process models to develop a formula unique to their own situation. This mix-and-match approach to Agile adoption works well in organizations that need to:

- **Fit Agile into a broader release cycle.** In real-world Agile adoption, Agile teams don't necessarily deliver a product with each iteration. In many cases, they contribute components to a larger release that is delivered on its own, lengthier timeline. Work flows from the individual team to a larger release team in the form of deliverables that the integration, quality assurance (QA), and other release-level team members handle on their own schedule. At the same time, work flows in the other direction in the form of requirements that end up in the component-level team's backlog. Microsoft takes this release-oriented approach to deliver the architecture functionality within Visual Studio. The feature crews, working for product managers, deliver functionality to the overall Visual Studio release. The release imposes requirements on those teams, including a timeline for functionality cutoff.⁷
- **Address governance and compliance requirements.** The rules that govern an organization may not support full Agile adoption. For example, in heavily regulated industries, such as the pharmaceutical and financial service industries, compliance requirements add work in the form of additional testing, documentation, and reviews. In these cases, internal and external authorities work at their own pace, often outside of an Agile team's ideal schedule.
- **Prioritize more than just the delivery of working software.** Even without compliance requirements, teams frequently create more documentation than is strictly required for their next deliverable, often because they have historically created such documentation to record progress and describe software. Teams look beyond the current sprint to create documentation for other needs, such as bringing new team members up to speed or educating outside parties such as support teams, trainers, and service partners.

Figure 5 Agile Orthodoxy Is The Exception, Not The Rule

“How would you characterize your adoption of Agile?”



Base: 52 development professionals who have adopted Agile (percentages do not total 100 because of rounding)

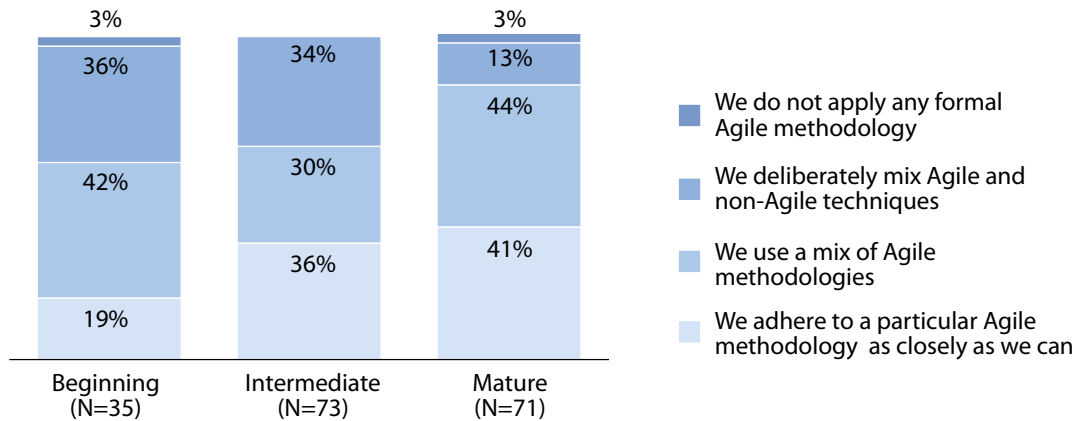
Source: Q3 2009 Global Agile Adoption Online Survey

56100

Source: Forrester Research, Inc.

Figure 6 Agile Heterodoxy Starts Early

“How would you describe your adherence to a particular Agile methodology?”



Base: technology industry professionals

Source: December 2008 Global Agile Company Online Survey

56100

Source: Forrester Research, Inc.

TEAMS ARE CHANGING TO SUPPORT AGILITY

While teams have adapted Agile to fit the particular circumstances of their organization, teams have also adapted themselves to fit the essentials of Agile. New attitudes have transformed individual contributors into team players. Work environments have changed, with some teams moving out of separate offices into a more open working environment. The prevalence of these changes is another sign of how Agile adoption has gone mainstream.

These changes, however, can be difficult for an organization to implement, and they often expose problems within an organization, putting new strains on weak points. Teams that have been sloppy about prioritization can face serious problems assembling a backlog or sticking to the functionality planned for a single sprint. Employees used to being responsible for their own deliverables can struggle to place the team's needs above their own. But teams are adjusting their dynamics to make Agile possible, focusing on evolved roles, updated practices, and practical application of process.

Successful Agile Teams Define Clear Lines Of Authority

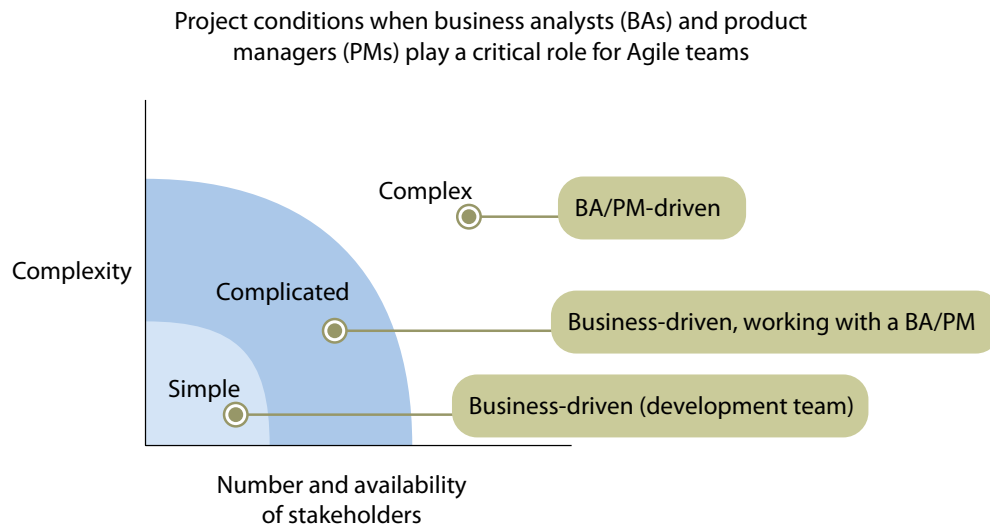
Agile depends on rapid decisions, so teams must streamline the decision-making process as much as possible. Delays scheduling meetings with key stakeholders or the “power vacuum” that occurs when no one wants to take responsibility for making a critical decision can be fatal to an Agile team.

To resolve these problems, teams adopting Agile clarify who has authority over which domains of decision-making. The “who” could be a group or an individual; the critical factors are that the power is vested and its definition is clear. Some common strategies for making the decision-making process more efficient include:

- **Having a product owner.** While teams might adopt Scrum without having a product owner, many teams embrace the product owner role to ensure that someone is clearly responsible for answering questions, defining general product direction, and prioritizing work.
- **Letting business analysts and product managers make business decisions.** Both business analysts (BAs) in IT departments and their counterparts in technology industry companies, product managers (PMs), bear de jure responsibility for making important business decisions, such as setting the priority for backlog items. Developers do not have the time — or often the desire — to meet with customers or users, analyze markets and use cases, and perform the other work needed to understand the business context and make sound product decisions. Therefore, teams devolve decision-making responsibility to BAs and PMs, who normally are not product owners but who focus more on the development team's daily activities.⁸
- **Defining project roles based on complexity and stakeholder characteristics.** As the complexity of a project increases, teams need to look to outside resources, such as business analysts or product managers, to handle research tasks and to mediate with stakeholders, particularly when there are many stakeholders with limited availability (see Figure 7).

- **Empowering the team.** Teams that have to wait for stakeholders to give them their time and attention must renegotiate their relationship with those stakeholders. Executives must have the power to set general direction but in all other matters step aside, giving the development team the ability to define the problem space, craft the solution, and deliver the results — for which the team will be held accountable.
- **Leaving room for mistakes.** Software development in any form entails an ongoing process of experimenting, learning, and revising. Allowing teams to fail gives them the opportunity to make mistakes early and correct them. This is especially powerful when risks arise late in the development cycle — for example, in system testing following unit testing — because it ensures that teams have the flexibility and the attitude to address errors head-on and ensure quality deliverables.

Figure 7 Project Characteristics Define When You Need A Business Analyst Or Product Manager



56100

Source: Forrester Research, Inc.

Coaching, Reporting, And A Desire For Improvement Replace Nitpicking And Arm-Twisting

For many of the same reasons that decision-making processes change in Agile environments, Agile environments also change how decisions are implemented. Traditional command-and-control models, with their lengthy processes of decision-making, approval, execution, review, and adjustment, take too much time. They also invite stakeholders and executives to become overly involved in how the team executes instead of empowering the team to figure out how best to achieve positive results. Additionally, in Agile environments, the relationship between the project manager and the team is different; in successful Agile environments:

- **Project managers become team advisors.** In the place of project management, coaching and support gain importance as techniques for keeping teams on track. Strong Agile project managers take responsibility for creating the best working environment possible for the team, removing impediments and distractions. The project manager increasingly takes on the guise of an outside expert, providing advice on how to get past obstacles and recognize potential risks.
- **Teams assume responsibility for planning and estimates.** Traditionally, project managers have assumed responsibility — officially or unofficially — for the project plan. Team members provide estimates, but as they are not the actual “owners” of the plan, they have weaker incentives to ensure that their estimates are accurate. Predictably, this commonly results in cost overruns and project delays — both of which are toxic to Agile adoption. Alternatively, teams that successfully adopt Agile methods share responsibility for planning, with the project manager coaching and facilitating that process. The team owns the estimates, the plan, and the outcomes.
- **Project managers build and maintain project “radiators.”** Instead of being the enforcement arm of the project plan, Agile project managers provide enormous value to the Agile team by tracking and communicating project status. Investment in new tools and procedures that provide continuous feedback enhance this value. Freed from being enforcers, project managers can experiment with better ways to monitor, analyze, and communicate status, with the ultimate goal of providing continuous and meaningful information.
- **Teams move from retrospection to modification.** Agile adoption involves not just new procedures but also new values. A desire to improve team structure and processes is critical. The project manager must ensure that, at the end of sprints, retrospectives happen and the team translates its conclusions into actual improvements. Once again, the project-manager-as-coach model depends on the project manager’s ability to collect good information, analyze it, and make suggestions to the team.

Teams Balance Architecture And Delivery

Many in the architecture community question the value of Agile methods, believing their lack of focus on artifacts and gaining a complete understanding of the problem leads to poorly architected

systems.⁹ However, we do not see signs that Agile teams set themselves up for architectural headaches. If these postponed architectural crises were happening, teams would eventually lose confidence in Agile, and we would expect to see Agile adoption decrease. Survey data and interviews show that this is not the case.

Instead of unleashing an architectural apocalypse, teams are figuring out how to balance long-term architectural considerations with each iteration's short-term deliverables by:

- **Balancing technical needs with business time frames.** While teams do risk overlooking architectural considerations in the short time span of a sprint, the reverse is also true: Without the discipline of Agile time-boxing, many teams succumb to the temptation to overengineer the system architecture. Adopting Agile forces teams to rethink how they balance business and technical needs, a question that easily gets lost in the complexities of longer release cycles. By creating a better formula to balance these requirements, teams not only improve the quality of the end product but also generate other benefits. The road map can become a more meaningful and accurate depiction of the business and technical trajectory of the team's work, ensuring more-balanced delivery of results.
- **Delivering architecture incrementally.** Architecture is not an all-or-nothing deliverable. In many cases, teams should be delivering architecture more incrementally to avoid making large-scale mistakes that are not identified until the end of the project. For example, engineers often try to anticipate performance bottlenecks as they build the underlying architecture. However, without a clear notion of the use cases and real-world tests of these use cases in practice, teams cannot have a realistic sense of how users will put stress on the system.
- **Delivering code, not PowerPoint slides.** Most architecture overviews come in the form of large PowerPoint decks. Effective adoption of balanced architecture along with Agile practices encourages the creation of code-based architectural models that developers use as the basis for delivering business value. To create these models, enterprise architects must not operate on a rarified plane but must align themselves closely with the team's regular work.

Team Size Doesn't Matter, Nor Does Distribution

When assessing how adopting Agile changes teams, it's worth noting what does *not* change and what organizational characteristics have not been an obstacle to Agile adoption. Prior to adopting Agile, many teams worry that Agile processes can't support geographically distributed or outsourced workers. An equally frequent concern is that Agile won't work for very large teams.

The stereotype of an Agile team as one that works in the same office, attends daily Scrum meetings, and does daily check-ins and builds of working code is the obvious source of these concerns. However, this image is not only a stereotype but also a highly inaccurate one. In practice, teams figure out how to work around potential obstacles such as:

- **Team size.** The stereotypical perception of an Agile team is that it consists of fewer than a dozen members. However, only about 50% of the teams we surveyed for our recent global Agile adoption survey reported having 50 members or fewer. The remaining teams were larger, with a third consisting of more than 100 members (see Figure 8-1).
- **Distributed geography.** Only a small minority of teams (17%) we surveyed are completely colocated. For the majority, *most* of the team members are colocated: 78% reported that up to 50% of their teams are colocated. The remaining 22% reported that more than 50% of the team is distributed in different locations (see Figure 8-2).
- **Use of outsourcing.** More than 50% of the Agile adopters we surveyed are willing to outsource components of software development, even though it requires both geographic and organizational distribution of effort (see Figure 8-3).

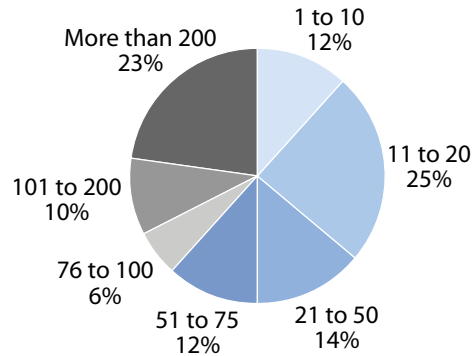
How have teams overcome these obstacles and managed to successfully adopt a methodology originally envisioned for small, colocated teams? Successful real-world Agile adopters succeed by:

- **Dividing labor within projects.** The most obvious solution is to break projects into smaller chunks, some or all of which can accommodate stereotypical Agile methods. Release teams then aggregate the work of these smaller subteams, usually organized through some version of a “Scrum of Scrums.”¹⁰ Higher-level team structures also handle integration, testing, and other tasks for the overall project. The project-level life cycle is usually simple, focused primarily on ensuring that each subteam delivers what’s needed and that dependency management remains feasible across deliverables.¹¹
- **Adopting techniques and technologies that support distributed collaboration.** A variety of tools, from Agile-focused project management aids to wikis and Web conferencing platforms, can help teams bridge geographic and organizational boundaries. Content communicated and managed through these tools includes backlogs, progress dashboards, and task lists. Team members also need to beef up their communication and listening skills. This is particularly important for globally distributed teams, where time differences and cultural variations can significantly impede progress.
- **Creating and sustaining well-designed partnerships.** While working with a systems integrator (SI) that also uses Agile methods is important, more-prosaic details about the contract with the SI are also important. Contracts need to reward progress and value rather than task completion and deliverables. Additionally, teams need to treat SIs as genuine partners, demanding transparent reporting and providing easy access to internal resources and systems and even coaching and training. Another best practice when working with outsourced providers is implementing logical groupings of work around teams to ensure clear ownership.

Figure 8 Size, Distribution, And Outsourcing Of Agile Teams

8-1 Team size among Agile adopters

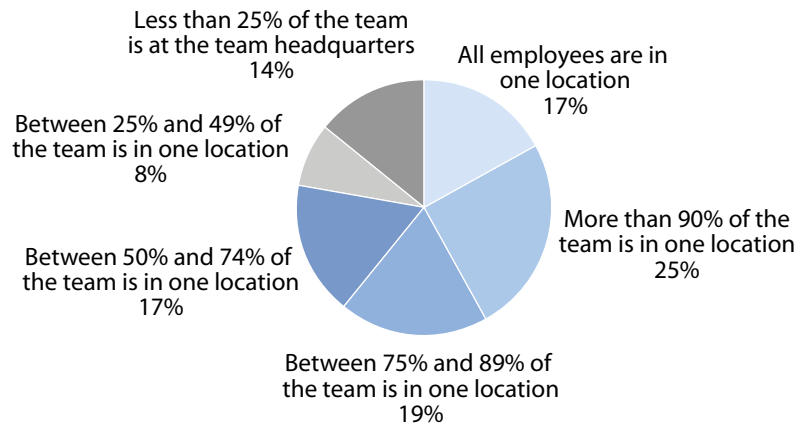
“How many employees belong to your development team?”
(includes all critical functions, such as QA, business analysts, product managers, etc.)



Base: 52 development professionals who have adopted Agile
(percentages do not total 100 because of rounding)

8-2 Team geographic distribution among Agile adopters

“What percentage of your team is at the team headquarters?”



Base: 52 development professionals who have adopted Agile

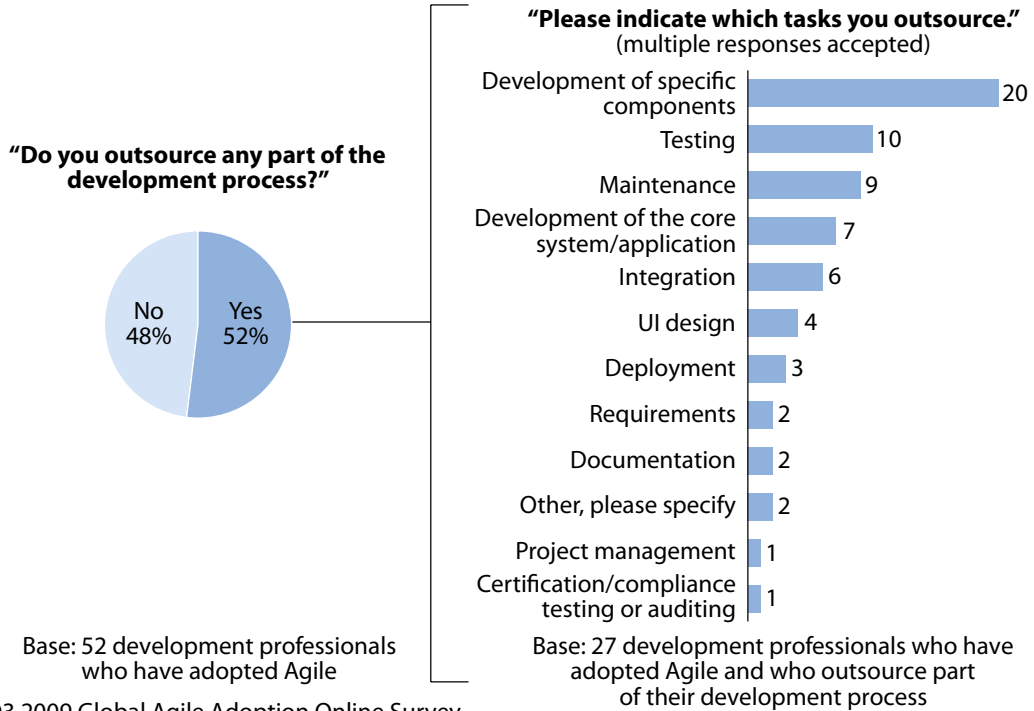
Source: Q3 2009 Global Agile Adoption Online Survey

56100

Source: Forrester Research, Inc.

Figure 8 Size, Distribution, And Outsourcing Of Agile Teams (Cont.)

8-3 Outsourcing among Agile adopters



Source: Q3 2009 Global Agile Adoption Online Survey

Source: Forrester Research, Inc.

56100

WHAT IT MEANS

AGILE BECOMES MAINSTREAM — BUT NOT WITHOUT SOME CHANGES

Agile adoption is a reality. Organizations across all industries are increasingly adopting Agile principles, and software engineers and other project team members are picking up Agile techniques. While historically, management has owned “process,” the adoption of Agile methods has pushed ownership into the hands of team members — many of whom have traditionally been skeptical of process and methodology. Broad Agile adoption requires careful consideration; a strong Agile adoption strategy should include:

- **A support plan.** Adopting Agile practices is not a trivial change; it requires support and time to become effective. The use of external coaches, training materials, and internal support groups can greatly increase the speed and success of adoption. As the use of Agile grows, it is important to build out a community to share ideas and provide support. This community can replace reliance on external coaches over time as it identifies internal mentors.
- **Flexible adoption models.** One size does not fit all. Agile implementations have as much variation as commonality. Specific challenges with people, problems, and environments will require Agile teams to change working practices and define appropriate techniques. They should keep in mind, however, the unifying principles of team ownership and experimentation, regular team collaboration, transparent reporting, and the use of frequent delivery to expose progress.
- **A focus on team empowerment.** It sounds really easy, but this is about more than just *saying* that teams are empowered. Empowered decision-making requires clear objectives and measurement coupled with the ability for teams to make decisions within their defined boundary of influence. Agile adopters who put in place objectives, controls, rewards, and measures are much more likely to succeed, because they rely on the team rather than the coaches for success.
- **A tool strategy.** A single team in one location working alongside a customer may be able to work without any electronic tools, but as organizations scale and teams become more distributed and part of much larger releases, Agile methods benefit greatly from tools. Teams should look to Agile application life-cycle management (ALM) tools that manage backlogs, support planning, and enable reporting to support their Agile approach.

RECOMMENDATIONS

APP DEV PROFESSIONALS SHOULD BLEND AGILE METHODS TO MEET THEIR OWN NEEDS

IT firms are increasingly looking to Agile methods to improve their time to market, provide flexible management approaches, and increase application quality. As Agile adoption has moved into the mainstream, Agile processes have evolved, resulting in a process that accepts and adapts to an organization's context and constraints. "Agile at scale" blends Agile and traditional methods in a hybrid approach. When adopting Agile, application development professionals should:

- **Provide teams with the support to adopt the right Agile mix.** Teams need Agile coaches to help them adopt the right Agile methods for their needs. Different situations warrant different degrees of Agility; therefore, teams that engage an experienced Agile coach are more likely to select the right practices and apply them in the right way. This success can form the basis of organizational implementation patterns over time.
- **Focus on measurement to help provide focus and direction.** Providing a common direction requires a clear set of objectives and a mechanism to report progress against those objectives. Teams adopting Agile practices will be undermined if their project objectives and measurement plan are based on traditional, artifact-centric approaches. Ensuring that the measurement plan focuses on delivering business value, measuring progress and quality, and managing risk provides the framework for achieving success with Agile.
- **Start with team principles and build practices, then move on to adopt other practices.** For IT teams, adopting some of Scrum's team principles combined with improving build practices through continuous integration and build provides a great first step to Agile adoption. The backlog-driven delivery process, continuous review, and frequent software delivery provide a firm foundation for additional practices associated with testing, measurement, and planning.
- **Involve architecture teams in the process.** In most IT development shops, architects play a pivotal role, understanding how the systems work and designing changes. It is easy to discount them from the team, focusing on activities associated with delivery rather than considering architectural concerns. Agile teams should counter this tendency, however, and instead involve architects as active team members, giving them responsibility for technical backlog items and delivery of long-term system quality.

SUPPLEMENTAL MATERIAL

Methodology

The Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009, was fielded to 1,298 application development and program management professionals who are readers of Dr. Dobb's magazine. For quality assurance, respondents are required to provide contact information and answer basic questions about themselves. Forrester fielded the survey from July 2009 to August 2009. Respondent incentives included a summary of the survey results and a chance to win one of five \$50 gift certificates.

Forrester fielded its Q3 2009 Global Agile Adoption Online Survey to 60 technology professionals from our ongoing Technology Industry Research Panel. The panel consists of volunteers who join on the basis of interest and familiarity with specific technology industry topics. For quality assurance, panelists are required to provide contact information and answer basic questions about their firms' revenue and budgets. Forrester fielded the survey from August to October 2009. Respondent incentives included a summary of the survey results.

Forrester's December 2008 Global Agile Company Online Survey was fielded to 241 technology industry professionals from our ongoing Technology Product Management & Marketing research panel as well as other Forrester clients and selected users of Agile. The panel consists of volunteers who join on the basis of interest and familiarity with specific technology industry topics. For quality assurance, panelists are required to provide contact information and answer basic questions about their firms' revenue and budgets. Forrester fielded the survey during December 2008. Respondent incentives included a summary of the survey results.

Exact sample sizes for the surveys used in this report are provided on a question-by-question basis. Surveys are not guaranteed to be representative of the entire application development population. Unless otherwise noted, statistical data is intended to be used for descriptive and not inferential purposes.

If you're interested in joining one of Forrester's Research Panels, you may visit us at <http://Forrester.com/Panel>.

ENDNOTES

- ¹ The Manifesto for Agile Software Development describes the principles of Agile software development and can be found at <http://agilemanifesto.org/>.
- ² For a further discussion of Scrum, see: Ken Schwaber, *Agile Project Management With Scrum*, Microsoft Press, 2004.
- ³ “Continuous integration and build” is a practice recommended in the eXtreme Programming methodology and first written about by Martin Fowler. His work can be found at <http://martinfowler.com/articles/continuousIntegration.html>.
- ⁴ Test-driven development (TDD) is a method described by Kent Beck to drive high-quality code and reduce opportunities for misunderstanding between requirements and design. See: Kent Beck, *Test Driven Development: By Example*, Addison-Wesley, 2002.
- ⁵ Refactoring has been employed informally for a number of years, but Martin Fowler described the practice in more detail. See Martin Fowler, *Refactoring: Improving The Design Of Existing Code*, Addison-Wesley, 1999.
- ⁶ Forrester published a report that gives a more complete picture of how to build harmony and balance between architecture and agility. See the July 10, 2009, “[Balance Architecture And Agility To Sustain Value Delivery](#)” report.
- ⁷ Forrester published a case study that describes one way that Microsoft manages its development. See the March 27, 2009, “[Case Study: Microsoft Speeds Tool Delivery With Agile Development](#)” report.
- ⁸ Forrester published a report that describes the ideas associated with Agile approaches to requirements management review. See the April 15, 2009, “[Just Do It: Modernize Your Requirements Practices](#)” report.
- ⁹ Forrester published a report that discusses in detail the debate between traditional architectural approaches and those that embrace Agility. See the July 10, 2009, “[Balance Architecture And Agility To Sustain Value Delivery](#)” report.
- ¹⁰ Scrum of Scrums has been described as a mechanism to scale Scrum, and Ken Schwaber defined it in his book, *Enterprise Scrum*. A great reference on the Scrum of Scrums can be found at: <http://www.scrumalliance.org/articles/46-advice-on-conducting-the-scrum-of-scrums-meeting>.
- ¹¹ Forrester published a report as well as a case study that describe the use of a lightweight life cycle. See the April 15, 2009, “[Best Practices: Software Development Processes](#)” report, and see the March 27, 2009, “[Case Study: Microsoft Speeds Tool Delivery With Agile Development](#)” report.

FORRESTER®

Making Leaders Successful Every Day

Headquarters

Forrester Research, Inc.
400 Technology Square
Cambridge, MA 02139 USA
Tel: +1 617.613.6000
Fax: +1 617.613.5000
Email: forrester@forrester.com
Nasdaq symbol: FORR
www.forrester.com

Research and Sales Offices

Forrester has research centers and sales offices in more than 27 cities internationally, including Amsterdam; Cambridge, Mass.; Dallas; Dubai; Foster City, Calif.; Frankfurt; London; Madrid; Sydney; Tel Aviv; and Toronto.

For a complete list of worldwide locations visit www.forrester.com/about.

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or clientsupport@forrester.com.

We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Research, Inc. (Nasdaq: FORR) is an independent research company that provides pragmatic and forward-thinking advice to global leaders in business and technology. Forrester works with professionals in 20 key roles at major companies providing proprietary research, customer insight, consulting, events, and peer-to-peer executive programs. For more than 26 years, Forrester has been making IT, marketing, and technology industry leaders successful every day. For more information, visit www.forrester.com.