

July 30, 2010

Agile Systems Integrators: Plausible Or Paradoxical?

by Dave West

for Application Development & Delivery Professionals



July 30, 2010

Agile Systems Integrators: Plausible Or Paradoxical?

Mixing Agile Methods And Systems Integrators Requires An Artful Approach

by **Dave West**

with Jeffrey S. Hammond and David D'Silva

EXECUTIVE SUMMARY

Agile methods have become popular because, on average, they produce better outcomes. Agile's focus on learning by execution and a condensed set of engineering practices increases the cadence of delivery when businesses need to supply innovative ideas in unknown business areas. However, the landscape of modern development organizations is a combination of in-house development resources and external systems integrators (SIs) that are both on and offshore. Forrester's research has found that SIs are increasingly looking to Agile methods to better partner with their customers and deliver innovative software. But the overall adoption of Agile within SIs is uneven: Specialist vendors such as Ci&T, emergn, and ThoughtWorks use Agile methods in the majority of their development projects, while large, generalist SIs are using Agile methods more sparingly. The differences among adoption rates are not solely the result of each individual SI's desire to be more Agile but are actually driven by the client's ability to adopt Agile practices in the context of the problems it has hired the SI to work on. Application development professionals need to change their working practices to work with their SI in an Agile way. These changes will not only enable the adoption of Agile processes but will also deliver improved innovation and increased business value.

TABLE OF CONTENTS

- 2 **Agile Is Going Mainstream, But Waterfall Is Still The Favorite With SIs**
- 7 **Systems Integrators That Adopt Agile Do Some Things Differently**

RECOMMENDATIONS

- 12 **Learn How Best To Work With An Agile SI**
- 13 **Supplemental Material**

NOTES & RESOURCES

From February to June 2010, Forrester surveyed 14 systems integrators, including Capgemini, CGI Group, Ci&T, Cognizant, Computer Sciences Corporation (CSC), emergn, HCL Technologies, HP, IBM, Infosys Technologies, Larsen & Toubro Infotech, Tata Consultancy Services, ThoughtWorks, and Wipro.

Related Research Documents

"Case Study: Ci&T's Path To Lean Methods"
July 1, 2010

"Tech Vendors Supporting Agile Must Be Adaptive"
May 5, 2010

"Agile Development: Mainstream Adoption Has Changed Agility"
January 20, 2010

AGILE IS GOING MAINSTREAM, BUT WATERFALL IS STILL THE FAVORITE WITH SIS

The Agile Manifesto describes an approach to software that is clearly at odds with the many characteristics of a software project that includes an external vendor.¹ It describes collaborative interactions, rapid change, and frequent delivery of working software instead of formal process phases, defined documentation, and heavy emphasis on planning (see Figure 1). Therefore, it comes as no surprise that systems integrators do not use Agile development approaches as frequently as the broader systems development community uses them (see Figure 2). Developers tell us that delivering high-quality software faster is the key driver behind their adoption of Agile approaches, but Forrester also sees Agile methods gaining traction in situations with a large amount of unknowns.² When there's a general lack of clarity, the ability to apply a method based on experimentation and observation enables development teams to gain better insights into the problem space and make the appropriate project course corrections — in other words, to fail faster (see Figure 3).

Figure 1 Conflicts Between The Agile Manifesto And Vendor Approaches

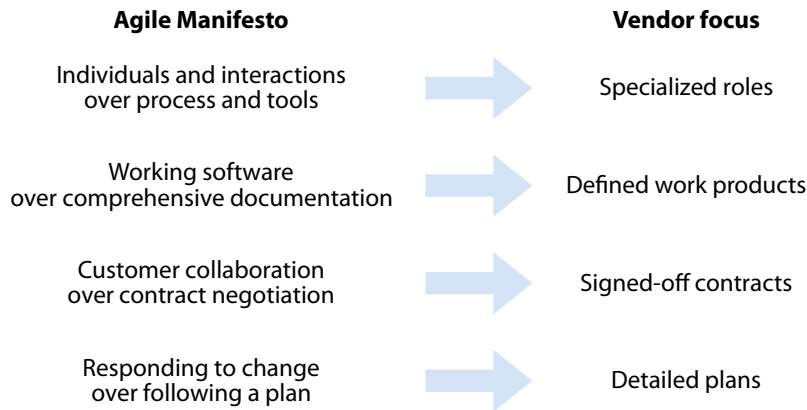
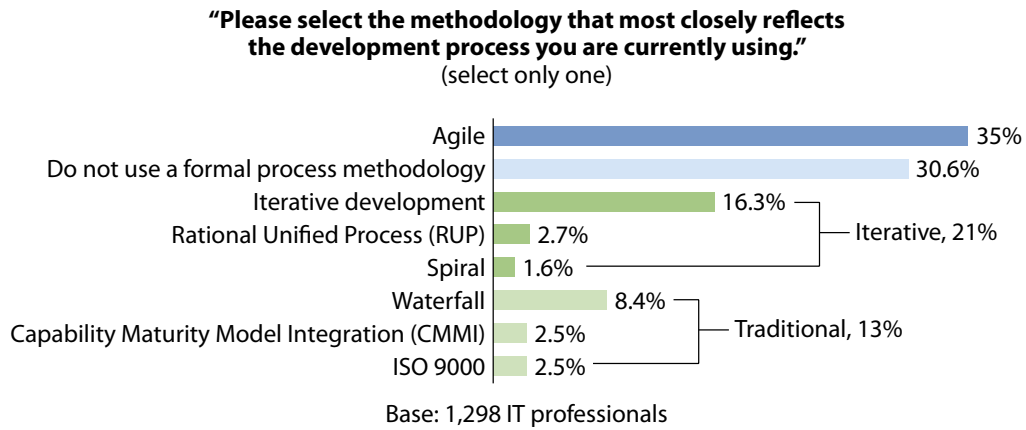
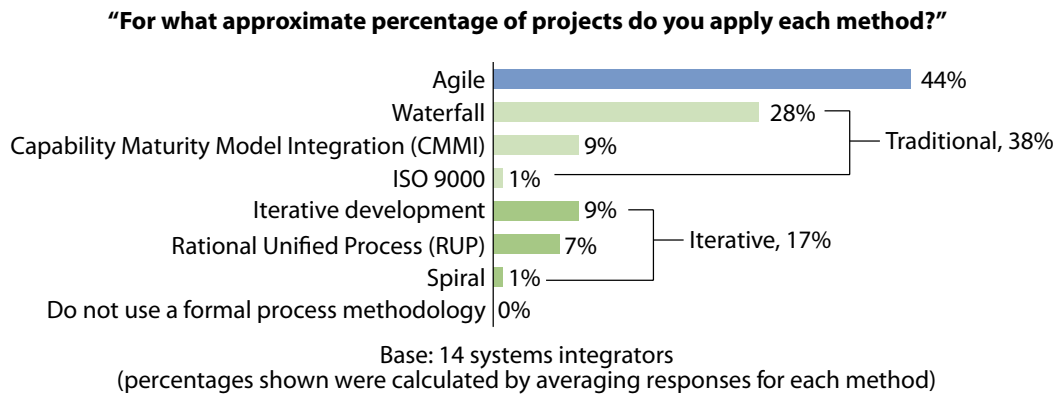


Figure 2 SIs Lag In Agile Adoption Compared With The Broader Development Community



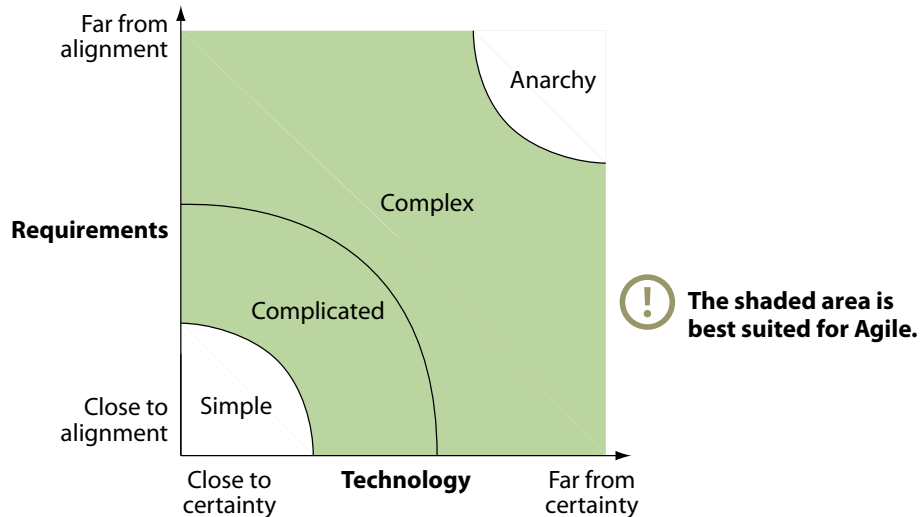
Source: Forrester/Dr. Dobb's Global Developer Technographics® Survey, Q3 2009



57353

Source: Forrester Research, Inc.

Figure 3 Managing Risk Is A Key Motivation To Move To Agile Methods



Source: Adapted from Ralph D. Stacey, *Strategic Management and Organisational Dynamics: The Challenge of Complexity*, Prentice Hall, 2000.

57353

Source: Forrester Research, Inc.

As Software Increases In Importance, Failing Slowly Is No Longer An Option

A project described by adjectives such as “unknown,” “complex,” “risky,” and “very important” does not sound like a good candidate for development; however, for many businesses, the post-recession economy requires a different way of applying technology. Growing revenues mean pursuing new markets, alternative sales channels, and previously untapped customer opportunities. Software development practices must evolve to:

- **Deliver business innovation rather than what was specified on paper.** On time and on budget are no longer software development projects’ key drivers; instead, projects must deliver business value and customer satisfaction. A project that delivers the wrong feature, even if that was what the contract specified, may still be considered a business failure.
- **Get feedback faster.** Speed can mean the difference between success and failure in many markets, but speed is not just associated with releasing software to production — it’s also how fast you can deliver working software internally. Internal and external customers, especially Millennials, have a more sophisticated appreciation of technology than do previous generations of users, so software development teams can benefit from increased feedback early in the life cycle.

- **Build quality into the application earlier.** Quality is no longer just about working software but now also covers how easy the software is to learn and use.³ Building great software is even harder today than it once was because today's problems require more than just simple automation projects; they deal with complex, often unknown business opportunities.
- **Increase the likelihood of project success.** If software increases in importance to a business, then so does the impact of failure. In many organizations, project success continues to be low, and the larger and more complex the project, the more likely it is to fail. The development leader of a large insurance company described the failure of large projects as the primary reason he is moving his developers to Agile approaches: "I did the numbers and found that all of my large projects were late and cost more money than I expected. That made the change to Agile an easy one for our company."

The gut reaction to increased complexity is to increase the rigor of the planning processes while increasing process discipline. Modern development processes instead suggest introducing approaches that encourage frequent planning in which teams take the experience gained during the course of the project and build it back into the next round of planning.

- **Better deal with uncertainty.** To better manage uncertainty, software development teams need to change from traditional, predictive approaches based on interim artifacts and defined process models to more flexible, observation-based approaches. The resulting software development processes are best described as a series of objectives rather than a list of activities to perform and deliverables to create. By focusing on the why rather than the how, modern approaches allow teams the flexibility to build the right process to deliver a successful outcome.

Agile Is A Bridge Too Far For Many SIs

As the volume of software to be written has increased, systems integrators have naturally filled in gaps in development capacity not only by lowering costs but also by increasing process and technology maturity.⁴ To prove the maturity of the process guiding their software development efforts, most SIs adopted standards such as Capability Maturity Model Integration (CMMI) or IT Infrastructure Library (ITIL). These processes stipulate a comprehensive set of process steps, voluminous documentation, and an extensive set of proof points for all projects, large and small. Initially, clients considered these comprehensive processes' overhead acceptable because the processes offered a way to manage the risk of working with an external vendor, especially when the implementation labor costs were substantially lower. Risk mitigation focused on two main areas: ensuring that the client knew what it wanted and guaranteeing that the vendor could deliver it. According to the 14 systems integrators we interviewed:

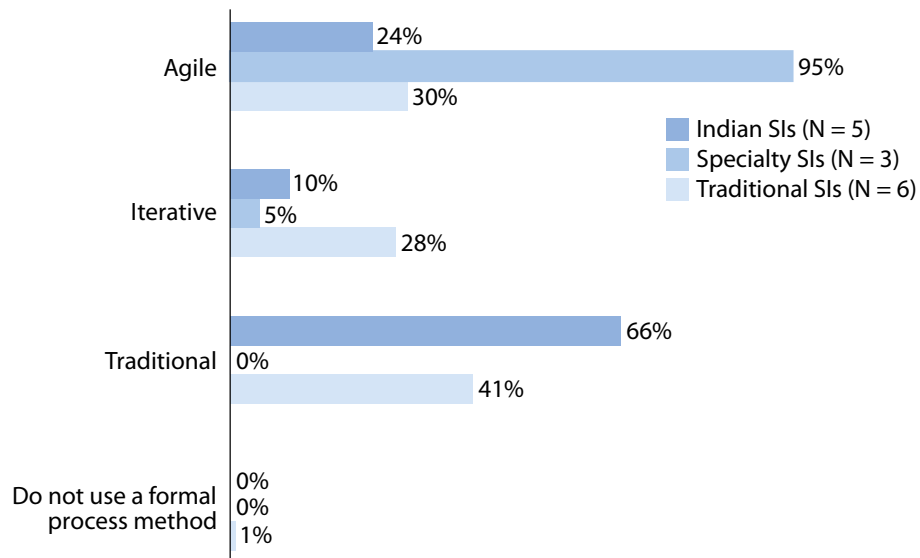
- **Waterfall is the preferred development approach, but only just.** Waterfall development, the traditional sequential approach to delivering software, is still the most popular approach with

the majority of the vendors we interviewed. The attraction of this approach for both SIs and customers lies in the simplicity of its execution. Stages based on discipline allow customers and vendors to focus on a particular set of skills at a certain point, and they also provide interim evidence of success via preapproved artifacts. Customer contracting departments also like the waterfall method because it's a good structure on which to build fixed-price contracts.

- **Iterative is a compromise for SIs looking for a more flexible process.** After waterfall, the next most popular development methodology among SIs is an iterative approach. SIs generally use an industry-standard approach such as the unified process or the spiral model as the basis of their approach.⁵ Iterative development is often described as mini waterfall, as it allows a team to break a problem down into a series of chunks and then apply a waterfall model to each smaller problem.⁶ The unified process extends this approach by focusing the initial mini waterfalls on risks associated with scope and architecture.
- **Most SIs use Agile sparingly, but specialists use it heavily.** A small number of SIs, including Ci&T, emergn, and ThoughtWorks, follow Agile approaches exclusively and focus their business, organizational, and commercial models on delivering projects using Agile methods. There is also a stark contrast between US-based SIs and Indian companies: Indian firms apply Agile less frequently than do their American counterparts (see Figure 4).

Figure 4 Use Of Agile Methods Varies Among SIs

“For what approximate percentage of projects do you apply each method?”



Base: 14 systems integrators
 (percentages shown were calculated by averaging responses for each method)

Staffing Challenges And A Bias Toward Waterfall Inhibit Adoption Of Agile Methods

When we asked SIs about what inhibited their adoption of Agile methods, the answers ranged from problems with tools to difficulties merging Agile process with clients' processes and procedures. The good news is that these are problems that SIs think they can solve. The bad news is that two other, more challenging problems still stand in the way of increased Agile adoption:

- **A strong cultural bias in favor of waterfall methods.** Years of working in a certain way make it hard to change to a new approach. Waterfall or sequential processes are often ingrained in all the aspects of an SI's relationship with its clients as well as deeply manifested in roles, meeting structures, document templates, and contracting models. Moving to Agile methods takes a lot more than simply retraining developers; it requires fundamental changes to the core structural, cultural, and political frameworks of the entire business, including the selling and marketing arms of the organization.
- **Lack of trained personnel.** Agile requires highly skilled people to work effectively together to deliver working software, and it helps if those people are trained in Agile methods. This can create a chicken and egg problem: Staff members within both the vendor and client will only be trained in Agile methods when the project calls for it, and the projects will not be Agile without trained individuals. Some SIs have succeeded in breaking this stalemate; for example, IBM has introduced an Agile night school program that educates staff members on the use of the Agile method while they are working on other projects. Agile SIs also note that the client's skills are often just as big a problem, as Agile methods require the client to undertake a number of key Agile responsibilities, including business ownership. To offset this risk, these Agile SIs undertake training as part of every engagement.

SYSTEMS INTEGRATORS THAT ADOPT AGILE DO SOME THINGS DIFFERENTLY

When an SI adopts Agile methods, it necessarily changes the way the SI works. These changes challenge traditional working practices and can be a shock to clients that have become comfortable with a traditional, sequential approach. Agile process changes are concentrated in the areas of requirements gathering, engineering practices, project reporting, and contract definition. These changes require application development professionals to challenge the status quo of how they approach working with SIs.

Requirements Process Change

The traditional development approaches SIs favor concentrate on large amounts of documentation as evidence of progress. These document-centric approaches encourage including numerous tasks and review sessions within the plan, often extending the project's timeline. A representative from a large aircraft manufacturer described this lack of flexibility in SI processes as the reason why the organization hired its own development resources and did not work with the regular

systems integrator, noting, “We find for really big projects our regular SIs do ok, but when it comes to more dynamic, smaller projects it is much easier to hire your own resources.” When SIs use Agile approaches, they must balance the need for demonstrated progress with organizational requirements and control points. The result is that SIs following an Agile approach do not discard all documentation but instead concentrate on:

- **Lightweight and incremental artifacts.** By delivering faster, Agile teams challenge the format and value of traditional requirements documents. The result is slimmer, audience-specific artifacts such as storyboards, visual prototypes, and even YouTube vignettes. Agile teams consider audience understanding and input more important than following any one document template.
- **Collaboration inside and outside of the team.** Instead of concentrating on executing a formal set of process reviews in which the SI meets with key stakeholders, Agile SIs concentrate on collaboration and frequent, informal discussions. They also take advantage of new media; 100% of the SIs we interviewed use wikis on their Agile projects.
- **Teams rather than specialists.** Agile SIs replace the traditional handoff-oriented, specialist-structured team with a cross-functional, collaborative team with multirole players. By defining shared goals and objectives, Agile teams are able to manage the resource peaks and troughs traditionally associated with working in an unknown situation. One interesting byproduct of Agile methods within SIs is the resulting change to the team structure in outsourced projects: Not only does Agile encourage more people being local, but it also creates vendor teams that have worked together before. The result is that Agile SIs upsell teams rather than resources.
- **Encouraging change rather than pushing back on it.** The backlog is a constantly changing artifact, with team members adding or removing items as they obtain a shared understanding of the project. Agile backlogs encourage change, allowing the addition of new items without committing to the work. Work is scheduled from the backlog for each iteration or sprint, allowing the team to focus on the work at hand. This approach removes the traditional conflict between the vendor and the client, where change is used as a bargaining chip in the blame game.

Modern Working Practices Are Adopted

By delivering more frequently, Agile projects often expose weaknesses in other practices. In particular, build, test, and architecture practices often require strengthening. Because Agile methods encourage development teams to improve the way they build software, Agile teams will naturally adopt better and better practices. Forrester has observed that Agile SIs tend to get the most bang for the buck from the following Agile best practices:

- **Continuous integration.** Build processes need to be automated and integrated with the delivery workflow to support increased throughput, even if delivery is internal. The goal is to create

automated builds that are triggered by a change in state of the tasks the team is working on. When this practice is employed, the client can observe the stability of the code by seeing the success of the builds and their frequency. And the ratio of successful builds and delivered tasks over time is a great external metric to use with clients as a predictor of overall project progress, status, and risk.

- **Test-driven development.** By employing a test-first process, Agile SIs deliver higher-quality software, but following this approach requires a different way of building requirements and testing infrastructure. For one thing, requirements need to be written with testing in mind, with a clear focus on expected results and test data. Testing infrastructure also needs to be ready much earlier in the life cycle, enabling testing to commence at the same time as the code is written. Clients need to work with the Agile SI to develop the right infrastructure and build requirements that include testing information in them.
- **Refactoring.** Frequent change and rework is one consequence of working in an Agile way. Agile SIs cannot insulate their customers from the need to refactor the architecture throughout the project, and this puts added pressure on the customer when dealing with integration into a broader enterprise architecture. Thus, many Agile SIs try to employ proven industry-standard models and open source technology to try to minimize the impact of change. For example, IBM's global services adopts business frameworks, such as the Insurance Application Architecture (IAA), that provide a certain base model around messaging infrastructure and the supporting data model.⁷

Transparent Reporting Is Key To Making Agile Work

“You can't change something that you cannot measure” is the maxim of Agile and Lean projects. Too often SIs hide internal measures and project status from clients, keeping two separate sets of books: one describing status to the client and the other expressing the true working status of the project. At best, this practice smoothes out the rough patches of a project; at worst, it precipitates catastrophic project failures or results in unprofitable contracts. Because Agile processes use a single set of measures, introducing them can actually prevent dysfunctional behavior, inviting straightforward discussion between customers and the SI. Transparent measurement results in:

- **Frequent — even daily — measurement of progress.** Agile methods replace periodic status reporting meetings with more-frequent status discussions. These sessions, such as Scrum meetings, provide the opportunity to observe work in progress, completed work, and obstacles to progress. These meetings involve everyone who is directly involved, allowing everyone on the project to have a more complete view of project status.
- **One set of books.** If a team is reporting status frequently, it is very hard to introduce additional views of the project for the customer. Instead, everyone sees the same view of the project, which allows both the customer and the SI to make decisions based on accurate information. In

addition, this enables the development team, inclusive of the client and SI, to make decisions concerning the process it is following and the work it is undertaking. This “tweaking” of the process is only possible if the team is reporting timely, accurate information.

- **The use of Agile application development management tools.** Agile methods do not mandate tool use, and some argue that the Agile Manifesto explicitly disregards tools in favor of people working together. However, when working with a vendor, which often does not share the customer’s locality, tools become helpful. Tools provide the ability to quickly harvest correct information and share that information with people who are not located in the same place.

Contracts Become Objective-Based Rather Than Deliverable-Based

SIs find that blending Agile approaches with their existing commercial models creates friction in the areas of resource optimization, planning, governance, and risk management. Widescale adoption requires changes to SI business models and also commercial relationships with clients. Application development professionals should look to challenge their existing contracts with a view toward:

- **Sharing success objectives with SIs.** Traditional approaches to managing relationships with SIs are by their very nature confrontational. The SI wishes to maximize time and effort while minimizing scope. The client wants more scope, but for less money and with quicker delivery. The result is often the need to specify in detail the tradeoff between these opposing forces with a contract that describes in some detail the functionality the SI will deliver or the activities it will undertake. Agile approaches are concerned with delivering understanding and value through execution. Thus, it is not realistic to specify how the Agile team is going to work or what it is going to deliver prior to the team undertaking work and achieving experience. Instead of concentrating on activity-oriented measures, application development professionals should think about including measures that describe value or business success. Having shared goals provides a great foundation for the vendor and the client to create long-term business success.
- **Paying for capabilities rather than activities.** Some Agile-oriented SIs interviewed in the survey such as Ci&T and ThoughtWorks employ a “pay for a team” approach. The customer selects a team with the right skills, and the vendor pays the team members to do a certain number of iterations or sprints. The team executes these iterations in a transparent way, allowing the customer to observe progress and provide feedback. Because it only commits the customer to a small number of iterations or sprints, this approach manages overall project risk. Furthermore, the SI employs traditional Agile management approaches to ensure that the project is going in the right direction.
- **Encouraging change rather than avoiding it.** When a contract includes a comprehensive description of a solution’s scope or specific details, these particulars become hard to change. Change is inevitable with Agile projects, as Agile teams strive to gain knowledge through execution and understanding. Application development professionals working on Agile SI

contracts should avoid including detailed content that is likely to change. Instead, they should focus on describing intent and required business objectives. This allows the SI to focus on delivering value, in line with the contract, rather than functionality that was originally thought — but may not turn out to be — valuable. A senior manager at a financial services company describes this different view as fundamental to success: “Traditionally we always spent weeks if not months describing the software we wanted. This resulted in that software being built, and only then did we realize that actually we really did not quite want that, but instead what we thought the original software would provide.”

- **Providing software-based, not paper-based, measurements.** Delivering working software is a great way to measure project success and should be encouraged over complex documents. This working software should be augmented with appropriate documentation to help the customer and vendor make decisions associated with project direction.

Where Agile Methods Are Applied Is As Important As When

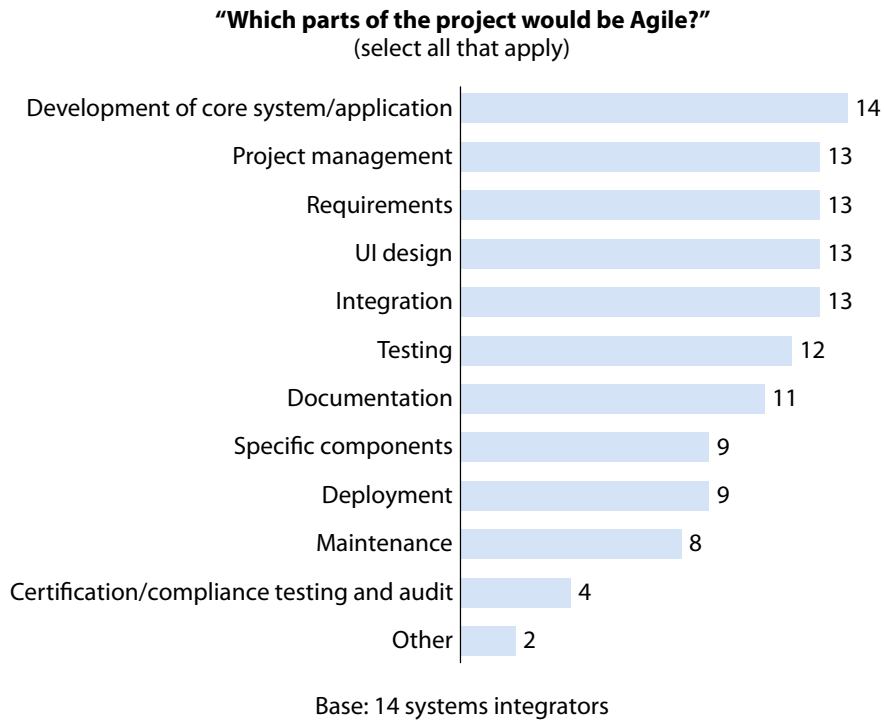
Increased use of Agile methods is pushing their application to new and different situations. When originally described by the Agile Manifesto, the goal for Agile practices was to help development teams build custom software better. Forrester is now seeing some SIs expanding their use of Agile techniques to include any situation where there is a team, a problem, and a large amount of risk (see Figure 5). These situations include core custom software development but also systems maintenance, project management, user interface design, and requirements gathering projects. A representative of one SI described this evolution, explaining, “We use Scrum to manage the majority of our work, even the work where we are not delivering software.”

The bottom line is that SIs are applying the principles of Agility, but with some modification around the practices they follow. For example, these project teams do not always deliver production-ready working software; instead, they may deliver a working prototype or a series of process flows. Pedantics would argue that this approach is not the true essence of Agile, as teams are not delivering software. But the reality is the pragmatic adoption of a process that leads both customers and vendors to blend the best bits of the original Agile Manifesto within their approach in a desire to deliver value to their customers. Agile practices are expanding to include:

- **Development of core systems.** It’s no surprise that teams are using Agile methods extensively as they develop core functionality associated with a project. What does come as more of a surprise is that SIs are applying Agile methods to both externally visible software development and more-back-office-oriented systems. These back-end development projects include customizing packaged applications and business intelligence (BI) solutions.
- **Maintenance of existing software.** A growing trend among SIs is using Agile approaches to better manage the maintenance of existing systems. The backlog best practice that Agile teams favor when organizing work is also a great tool for managing maintenance work, and the

frequent-release-oriented nature of Agile also lends itself well to the paradigm of maintenance work. Where maintenance is the focus, we also see popular Agile techniques such as Scrum augmented with flow-oriented approaches such as Kanban.⁸

Figure 5 Systems Integrators Report Using Agile For Many Different Project Types



57353

Source: Forrester Research, Inc.

RECOMMENDATIONS

LEARN HOW BEST TO WORK WITH AN AGILE SI

The application of Agile practices has fundamental implications for both the SI and you, their clients. The results we’ve seen from specialist SIs are promising in terms of productivity and quality, but reaping these benefits requires some fundamental changes to your working relationships. To take advantage of this new way of working, application development professionals must:

- **Encourage a different planning process.** Planning in detail at the start of the project without a clear understanding of what the project really is leads to lots of assumptions and padding in estimates. It can also lead to a low-trust environment as both parties try to protect their interests. Instead of following a “plan once, sign off, and execute” approach, application development pros should plan many times and feed execution experience back into the review process as an input for replanning at the start of each sprint.

- **Engage more frequently with the SI.** Traditional approaches to working with vendors often encourage a “throw it over the wall” approach. Adopting an Agile or Lean approach often requires frequent collaboration. In an outsourcing model, time zones can make this harder, as both organizations need to be in the office to collaborate. Thus, near-shoring models combined with embedded local resources can help enable this level of interaction.
- **Provide business leadership throughout.** Requirements documentation review and sign-off is a poor substitute for business leadership, and Agile and Lean approaches encourage vendors to provide clear product owners and include them in the cross-functional delivery team.
- **Focus on working software rather than documentation.** A key tenet of Agile development is that the client encourages the vendor to deliver software frequently but also has the ability to review the software. This means that vendor and client must have shared environments where the vendor can frequently deliver software and the client can review it.
- **Introduce shared measurement and dashboards.** Transparency is a key tenet of Agile development; thus, in Agile approaches, it is important to implement a simple shared dashboard that describes the real-time status of the project. Meetings should be conducted with this — and only this — information. If the dashboards do not provide adequate information, they should be changed to reflect the information that is considered important. By using a single shared version of the truth, teams eliminate wasteful reporting processes.

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.

Companies Interviewed For This Document

Capgemini	HP
CGI Group	IBM
Ci&T	Infosys Technologies
Cognizant	Larsen & Toubro Infotech
Computer Sciences Corporation (CSC)	Tata Consultancy Services
Emergn	ThoughtWorks
HCL Technologies	Wipro

ENDNOTES

- ¹ The Agile Manifesto can be found at <http://agilemanifesto.org/>.
- ² A recent survey of 52 development professionals doing Agile found that 64% cited the ability to make midcourse corrections as the key motivation for applying an Agile method.
- ³ Mike Gualtieri and Mary Gerush described the changing importance of design in a recent Forrester report. See the May 10, 2010, "[Business Analysts: Seize The Opportunity To Deliver Compelling User Experiences](#)" report.
- ⁴ In line with the continued growth in technology, more and more software is being written. Examples abound, but taking straight lines of code, the avionics system in the F-22 Raptor has approximately 1.7 million lines — compared with the latest Mercedes, which has somewhere in the region of 20 million lines in its navigation system alone. Source: Robert N. Charette, "This Car Runs on Code," Discovery News, February 5, 2010 (<http://news.discovery.com/tech/toyota-recall-software-code.html>).
- ⁵ The spiral model was developed by Barry Boehm in 1986. For a complete description, check out the ACM Web site (<http://portal.acm.org/citation.cfm?doid=12944.12948>).
- ⁶ Mini waterfall was a term coined by Walker Royce in his book *Software Project Management: A Unified Framework* (Addison-Wesley Professional, 1998).
- ⁷ To reduce the cost and impact of refactoring, IBM and other SIs offer a series of complete industry-specific models that reduce risk. An example of the IBM model can be found at the IBM Web site (http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=/com.ibm.ws.icp.insp_c.doc/ins/pc/p_cdev/concept/intro/c_ovrhl7std.html).
- ⁸ Kanban is a signaling system developed by Toyota to provide insights into production flow. A great description of the use of Kanban with Scrum can be found at the Lean Engineering Software Web site (<http://leansoftwareengineering.com/ksse/scrum-ban/>).

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 19 key roles at major companies providing proprietary research, customer insight, consulting, events, and peer-to-peer executive programs. For more than 27 years, Forrester has been making IT, marketing, and technology industry leaders successful every day. For more information, visit www.forrester.com.