



IBM Center for
The Business of Government

Using Technology Series

A Guide to Critical Success Factors in Agile Delivery



- Agile Delivery Champions
- Existing Knowledge
- Verbal Communication
- Product Owner
- Reviews
- Top Staff
- Tooling Needs
- Upfront Work
- Specialty Skills

Paul Gorans

IBM Global Business Services

Philippe Kruchten

University of British Columbia

A Guide to Critical Success Factors in Agile Delivery

Paul Gorans

IBM Global Business Services

Philippe Kruchten

University of British Columbia



TABLE OF CONTENTS

- Foreword** 3
- What is Agile?** 4
- What Are the Benefits of Adopting an Agile Approach?** 9
- Ten Critical Success Factors for Implementing Agile Delivery**..... 10
 - Success Factor One:** Changing the Acquisition Process to Support Agile Delivery 10
 - Success Factor Two:** Integrating Executive Champions and Stakeholders into an Agile Initiative 12
 - Success Factor Three:** Using Existing Knowledge and Not Reinventing the Wheel 13
 - Success Factor Four:** Implementing More Verbal Communication and Dashboards 15
 - Success Factor Five:** Including the Right Product Owner and Mission Subject Matter Experts 16
 - Success Factor Six:** Implementing Reviews that Support Agile Delivery 17
 - Success Factor Seven:** Selecting Top Staff for Lead Roles in the Agile Project 18
 - Success Factor Eight:** Planning for IT Infrastructure and Tooling Needs 19
 - Success Factor Nine:** Conducting “Just Enough” Upfront Work Before the Start of the Agile Project 20
 - Success Factor Ten:** Integrating Critical Specialty Skills to Support Agile Teams 21
- Acknowledgments**..... 23
- About the Authors**..... 24
- Key Contact Information**..... 26

Many complex IT programs are encumbered by requirements that continually change over lengthy timeframes. The results are often cost overruns and schedule delays. As a result, desired mission objectives are not achieved.

Numerous studies and years of implementation experience with software development within complex IT projects provide evidence that Agile approaches, when executed correctly, improve the delivery of software and large system integration projects. For optimal results, Agile approaches must be planned, implemented with discipline, and tailored to the need of the project and the organization.

Agile delivery approaches support the federal government's goals of doing more with less and improving the agency's ability to manage their budgets and delivery dates.

While the Agile movement started officially in 2001, and is relatively young, most Agile concepts and practices have been applied to projects for decades. They are still popular because they have been proven to work. However, far too many problem implementations of Agile exist, and these unsuccessful implementations have generated some negativity about the Agile movement.

The purpose of this *Guide* is to help mission executives and program leaders understand how best to leverage Agile values and benefits. Agile can be used as a tool to leverage IT in a way that minimizes time and cost and maximizes mission and operational effectiveness. This *Guide* sets forth 10 critical success factors for implementing Agile delivery. The critical success factors are based on lessons learned from delivering large, complex projects and programs, as well as formal assessments of troubled Agile initiatives. We hope that this *Guide* will be highly useful to executives throughout the federal government as they move toward implementing Agile projects.



Daniel J. Chenok
Executive Director
IBM Center for The Business of Government
chenokd@us.ibm.com



Angela Carrington
Partner and Vice President
Homeland Security, Justice and State
IBM Global Business Services
angela.carrington@us.ibm.com

Agile is a set of values and principles based on best practices for delivery of software and other IT projects. When implemented in a disciplined manner and scaled to the needs of the project, program, or portfolio, Agile values and principles facilitate and validate the demonstration of a working solution to stakeholders frequently and in small pieces. Agile provides the flexibility to adapt to changes over time. The key intent of Agile solution delivery is to provide value to an organization in increments, which are adjusted and built over time into a scalable solution.

The modern Agile movement started in 2001, when a group of thought leaders experienced with various approaches to deliver software rapidly and in small chunks (in a “lightweight” manner) converged on a set of values and principles. It was considered a rebirth of the Rapid development movement based on what this

Key Ideas in The Agile Manifesto

- **Individuals and interactions over processes and tools.** People are the most important success factor. Too much emphasis is often placed on coding knowledge and development tools. Instead, team members and their communication with each other should carry a much larger role.
- **Working software over comprehensive documentation.** Software documentation is important, yet information transfer is more effective through the code itself and through human interaction.
- **Customer collaboration over contract negotiation.** Successful software development requires frequent communication and collaboration between the user and the developer, rather than a traditional statement of work.
- **Responding to change over following a plan.** Long-term project plans are not adaptable; short-term plans provide more flexibility in responding to change. It is considered more effective to devise a detailed plan for a two-week period and a general plan for a three-month period.

Source: Anne L. Fruhling and Alvin F. Tarrell, *Best Practices for Implementing Agile Methods*, Washington, DC: IBM Center for The Business of Government, 2008.

group of people could agree on at that time, and was enshrined in a document known as the Agile Manifesto. The Manifesto was not intended to encompass all aspects needed for success—it was intended to be more of a starting point than an end stage. The Agile Manifesto, as published in 2001, has two basic components: Agile values and Agile principles (they can be found at www.agilemanifesto.org). The box on the previous page, *Key Ideas in the Agile Manifesto*, presents the four key concepts included in the Agile Manifesto.

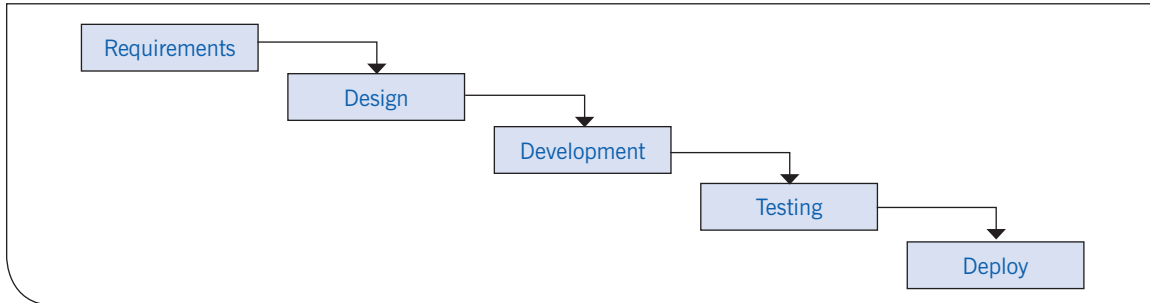
An Agile approach is especially relevant because the rapid pace of technology can cause a project's scope to change frequently. A basic project management principle is scope and resources will vary over time. More traditional development approaches often assume that scope and requirements are well understood, and exceptions can be managed by adding additional time and resources. In contrast, Agile approaches acknowledge the difficulty of anticipating all requirements upfront and promote efficiency and effectiveness by allowing some requirements to evolve over time while fixing the time and resources. Work is planned and implemented in smaller pieces, quickly providing evidence to stakeholders that teams are (or are not) producing what they want within a shorter timeframe.

Using an Agile solution delivery approach is very different than a traditional solution delivery approach, such as the “waterfall” approach. Figure 1 on page 6 presents the traditional waterfall approach. Figure 2 on page 6 captures an Agile approach, which is much more iterative. The box on page 7, *Differences Between Agile and Waterfall Approaches*, describes four key differences between the two approaches.

Two of the most popular, basic Agile methods are “Scrum” and “eXtreme Programming.” Both methods embody the intent of the Agile values and principles. Most Agile approaches have a mix of practices and techniques that provide further guidance, as well as light templates. The box on page 8, *Well Known Approaches to Agile Development*, describes the Scrum Process and eXtreme Programming.

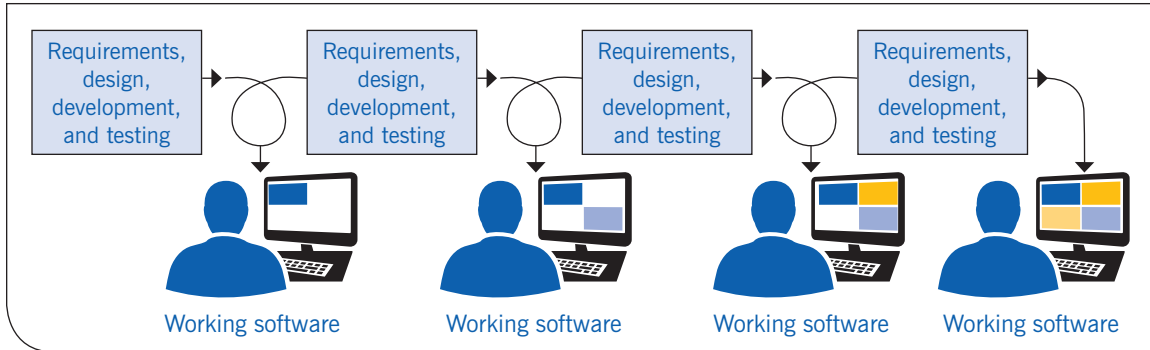
Years of implementation experience have demonstrated that when Agile practices are integrated with discipline and tailored to the scale of an initiative, business results are optimized. “Discipline” requires that sound practices are tailored to specific business needs and can be traced back to Agile principles and values. For example, an existing, stable application that requires enhancements may need only a basic Agile delivery method such as Scrum. However, to successfully deliver a larger, complex project requiring many Agile teams in an organization, Scrum alone will not provide a sufficient solution.

Figure 1: Waterfall Approach to Software Development



Source: Adapted from Royce, W. (1970) *Managing the Development of Large Software Systems*. Proceedings of the IEEE WESCON Conference, Los Angeles. Aug. 25–28, 1970, pp. 1–9.

Figure 2: Agile Approach to Software Development



Source: Adapted from Government Accountability Office, *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods*, July 2012 (GAO-12-681)

Differences Between Agile and Waterfall Approaches

- **Timing and scope of software development and delivery.** In an Agile project, working software is produced in iterations of typically one to eight weeks in duration, each of which provides a segment of functionality. To allow completion within a short timeframe, each iteration is relatively small in scope ... In contrast, waterfall development proceeds in sequential phases of no consistent, fixed duration to produce a complete system ... Waterfall phases typically address a single step in the development cycle.
- **Timing and scope of project planning.** In Agile, initial planning regarding cost, scope, and timing is conducted at a high level. However, these initial plans are supplemented by more specific plans for each iteration and the overall plans can be revised to reflect experience from completed iterations ... In contrast, in traditional waterfall project management, this analysis is documented in detail at the beginning of the project for the entire scope of work.
- **Project status evaluation.** In Agile, project status is primarily evaluated based on software demonstrations ... In contrast, in traditional project management, progress is assessed based on a review of data documents at predetermined milestones and checkpoints.
- **Collaboration.** Agile development emphasizes collaboration more than traditional approaches do ... In contrast, with traditional project management, customer and technical staff typically work separately, and project tasks are prescribed and monitored by a project manager.

Source: Government Accountability Office, *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods*, July 2012 (GAO-12-681).

Well Known Approaches to Agile Development

- **The Scrum Process.** Scrum is a project management method for Agile software development. The Scrum process consists of a combination of daily meetings, Scrum “sprints” to the next incremental delivery point, time boxing ongoing development work, and processes for managing the backlog ... Organizations can achieve several benefits when effectively implementing Scrum.
- **eXtreme Programming.** One of the most prominent approaches adhering to the principles of Agile software development is eXtreme Programming, or XP. A lightweight software development methodology, XP was originally designed for teams of up to 10 people and for projects that need to develop software quickly in an environment of vague or rapidly changing requirements.

Source: Adapted from Anne L. Fruhling and Alvin F. Tarrell, *Best Practices for Implementing Agile Methods*, Washington, DC: IBM Center for The Business of Government, 2008.

While many agencies are having success with basic Scrum practices, others still struggle to leverage the power of Agile for their key initiatives, or have not optimized their practices to realize full lifecycle benefits from Agile. Many have a long requirements phase, then distribute work to Agile teams for development, followed by a multiple-month testing and deployment phase; one term for this approach is “Water-Scrum-fall.” Such an approach may improve the quality of the software being produced, but provides only limited speed-to-value benefits for the agency.

WHAT ARE THE BENEFITS OF ADOPTING AN AGILE APPROACH?

Agile approaches, when well planned, implemented with discipline, and tailored to the scale of the project or organization, can produce better-quality software, faster, and at an overall lower cost. Agile presents an opportunity for tangible benefits and cost savings to the federal government if governance models for managing the technology program are adapted to support lighter deliverables, fixed resources and time, and variable scope. Benefits include:

- An Agile project can be completed on budget, delivered on time, and satisfy the organization overseeing the project, providing the scope is not fixed from the start.
- With Agile, the organization sponsoring the project gets the features they want via an incremental approach, providing more flexibility to adjust the software features contained in each release and each iteration as organizational priorities evolve.
- Fewer defects remain in the delivered product because they have been caught and corrected earlier.
- Less money is spent analyzing, designing, and planning features a year or more in advance that could change by the time of implementation, become a lower priority, or may never need to be developed at all.
- Federal government agencies can adopt new technology as it emerges, as opposed to waiting for results from a two- or three-year waterfall delivery cycle where the technology may be out of date once implemented.

Using Agile in the Federal Government

At a federal agency, a large program was transitioned to an Agile approach after two years developing requirements and 16 months delivering a first release. Initial Agile planning included securing executive stakeholder support for a disciplined approach, defining quarterly development release cycles, securing resources and stakeholder support and changes from the various program entities, defining Agile practice usage and documenting it all in an Agile Charter. Teams were selected and trained in the tailored approach, release preparation activities were conducted, and then one-month iterations with multiple Agile teams began. This resulted in the delivery of all planned capabilities on time in the first, four month, Agile release. Shortly afterward, the monthly maintenance release team was transitioned to an Agile approach, which resulted in an increase in the number of service requests delivered each month. The agency was very satisfied and trust and credibility with the delivery team was increased. That is what an Agile approach, with discipline and to scale, can achieve.

Based on experience implementing and evaluating numerous Agile projects over time, this *Guide* offers 10 critical success factors for government executives and managers to leverage in making Agile successful in their organizations. The [GAO report](#) (GAO-12-681) on Agile cites success with Agile at multiple government agencies and makes recommendations for success that either support or identify a need for the 10 success factors discussed on the following pages.

Success Factor One: Changing the Acquisition Process to Support Agile Delivery

It is difficult for a federal government agency to execute a project or program with a contractor using an Agile approach without changing several aspects of how the agency engages with the contractor. Acquiring Agile solution delivery with partners requires more than adding a bullet such as “The contractor will use an Agile method” to the standard, traditional, waterfall RFI or RFP template that contains traditional phases, gates, deliverables, and roles and responsibilities.

Agile requires a change in how work is procured, executed, and monitored. Proposal and contract language needs to be adapted to enable partners to deliver with lighter deliverables, fixed resources, fixed time, and variable scope. Mission leadership and subject matter experts (from both parties, government and contractors) must be committed to participate directly in the Agile approach.

For larger projects, it is necessary to consider how the engagement and first release will be initiated. It should be clearly stated that the contractor will lead initial work sessions to kick off the Agile engagement and conduct “Release Planning” activities with “just enough” requirements, design, architecture, and planning to initiate and support multiple Agile delivery teams. Further, the agencies will work with the contractor to develop a capacity-based estimate and right-size the Agile teams for a fixed capacity, fixed time (time boxed releases or iterations) variable scope release. If Agile or Scrum is requested without addressing how the preceding and succeeding waterfall phases, gates, and deliverables will be adapted, an opportunity to improve the end-to-end delivery is lost. Furthermore, requesting Scrum certified resources does not assure contractor-provided resources are qualified in their basic solution delivery skills, or that they have the Agile training and experience commensurate with the need.

TEN CRITICAL SUCCESS FACTORS FOR IMPLEMENTING AGILE DELIVERY

Most agencies have multiple contractors engaged. Recently, an increasing number of agencies are including multi-contractor procurements for every project. This causes potential constraints to executing an Agile approach. For example, having one contractor responsible for a phased requirements process, with a second responsible for development using Agile, and perhaps a third responsible for final testing and deployment, can be complex to manage and result in additional agency and contractor overhead cost. To execute an Agile approach, all three contractors must have resources integrated in activities that the others are responsible for. If multiple contractors are going to be leveraged for Agile project delivery, then clearly define the roles of each party, who will play the role of systems integrator, and how they will all work together as one team.

The agency mission, IT, and procurement leadership should meet upfront to jointly define their intent and work to assure that all entities have the contractual ability to execute together as one team. Post award, these individuals should meet on a routine basis to monitor program progress.

Success Factor Two: Integrating Executive Champions and Stakeholders into an Agile Initiative

Just like any initiative that impacts multiple groups across a large organization, an Agile initiative requires executive champions from the mission, IT, and other stakeholder offices who are willing to challenge the status quo, empower Agile initiative leaders to make changes to operating procedures, commit resources, and actively participate in resolving challenges.

It is difficult to succeed with an Agile approach without strong executive leadership. One way to secure that sponsorship is to develop a charter to define the initiative, goals, objectives, organizational scope, impacted entities, roles and responsibilities, and resources (human and other capital expenditures) that are necessary for success; this should involve key participants from both the business and technical groups including legal, security, privacy, and stakeholder readiness. The executives should sign the charter to formally commit themselves and their resources to the success of the Agile initiative.

Success Factor Three: Using Existing Knowledge and Not Reinventing the Wheel

Far too often, organizations hire coaches who begin with basic Agile training on simple Scrum practices and have the teams re-learn lessons through execution, as opposed to considering what has worked elsewhere, and planning for the avoidance of common challenges to Agile that exist in most traditional organizations. This coaching approach often leverages retrospective assessments at the end of each iteration. Retrospectives are important and leverage each team member in the identification and implementation of improvements for their team and the project. However, since the start of the Agile movement in 2001, numerous lessons have already been learned, shared, and communicated via Agile conferences, books, and other forums. Those lessons learned should be leveraged during Agile planning to identify barriers to agility within the organization that could impact the project, and resolve many of them prior to the initiation of a first Agile project team.

In addition, delivery frameworks or processes must be commensurate with the agency's challenge. Agile per se is not the objective; rather, Agile can provide a better solution for the organization. Advanced frameworks, or delivery processes that leverage other proven practices and light deliverables, may be useful as a starting point. Doing "just enough" upfront work for a given release can better set stakeholder expectations, enable extensibility and scalability of the solution, and provide a roadmap for the many Agile teams to follow. The ultimate results are better quality products and higher productivity from each Agile team.

Example Frameworks and Resources

- **Lean Development Principles.** These principles complement and supplement some of the gaps in simple Agile delivery approaches, and have been woven into other disciplined, scalable Agile frameworks. For larger, more complex projects, Lean development practices bring common sense and economies of scale to overly simplistic Agile approaches. For example, the Poppendiecks include the principle of “Eliminate Waste” and offer examples such as “Building the Wrong Thing” or “Building the Thing Wrong.” Waste and failure to “optimize the whole” have been recurring issues discovered through Agile assessments that can be addressed by integrating Lean Development Principles (<http://www.poppendieck.com>).
- **Disciplined Agile Delivery (DAD).** This decision process framework is a hybrid agile approach to enterprise IT solution delivery that provides a solid foundation from which to scale. (<http://disciplinedagiledelivery.com>).
- **Scaled Agile Framework (SAFe).** This framework includes a mix of Agile and Lean practices that scale to the enterprise (<http://www.scaledagileframework.com>).
- **Agile With Discipline Delivery Process (AWD).** This IBM delivery process integrates Agile and Lean practices, including step-by-step guidance and light templates for planning and delivering large, complex projects comprised of many Agile teams, from start to finish. It is tailored fit-to-purpose for solution delivery engagements.

Success Factor Four: Implementing More Verbal Communication and Dashboards

Organizations implementing an Agile project should establish verbal communication, including at least one weekly meeting with key agency executives and technical stakeholders. This helps assure those stakeholders have an awareness of challenges and an opportunity to assist in their resolution. Agencies should develop a communication plan that encompasses all stakeholders and participants to create easy access and visibility of progress at all levels. The plan should include and refine the following communications:

- Verbal communication at the Agile team and cross-team level through daily standup meetings
- Light documentation or tooling for communicating the planned scope of the release to stakeholders, including the teams that will support the product as it is incrementally released. Examples of this are architectural decisions, light design for each story, test approach, final testing outcomes, and deployment preparedness.
- Demonstrations of working features to key stakeholders at the end of each iteration in order to receive ongoing feedback
- Wikis or other Agile tools that provide central access to all information to all Agile team members, support resources, and stakeholders
- Metrics that show capabilities committed to a given release, and a summary of how many stories were closed by each team against their commitments for the given iteration

Success Factor Five: Including the Right Product Owner and Mission Subject Matter Experts

The best teams without the right product ownership may not create what end users need. It is important that the right mission subject matter experts (SMEs), who can truly represent the product end-users, be assigned to each Agile team. The subject matter experts should have functioned in a mission role and should possess skills to solicit the input of other mission SMEs and evaluate and negotiate to mission needs.

For enterprise initiatives, it is difficult to find one product owner who can know all requirements. The product owner should have a breadth of knowledge and be respected by the end users, but also needs to be able to involve and weigh the intent of the executive stakeholders, end users, and staff. Those decisions could mean the difference between the application being deployable or not. If the intent of the solution is to change how the business works today, then other stakeholders may need to be involved to address those gaps, as opposed to the involvement of only the mission and information technology SMEs who may be overly comfortable with the current business process and IT solution. The implementation of a product management team should be considered to provide structure to the product vision and implementation, through product owners and mission SMEs on each Agile team.

Success Factor Six: Implementing Reviews that Support Agile Delivery

A traditional System Delivery Life Cycle (SDLC) includes sequential phases of analysis, requirements, design, construction, and testing before a solution or release of a solution is implemented. This is also known as water-fall development which was discussed earlier in this *Guide*. At the conclusion of each phase, mandatory documents are reviewed, assumed to be complete, and then approved before the project can proceed to the next phase. Agile teams execute many activities from these traditional phases within Agile iterations; as a result, traditional phases and gates do not support the needs of Agile teams.

Agencies should require—even with an Agile approach—checkpoints to assess progress, assure conformance with relevant standards, and evaluate planned vs. actual scope delivered against the allocated budget. Some agencies that use a four-phase Rational Unified Process (RUP) governance framework are a step ahead in reducing the number of phases and gates in their lifecycle. However, many organizations have implemented RUP governance cycles with rigid, long time-boxes, without allowing Agile project teams the flexibility to tailor the iterations to fit the need of the project. Some organizations have implemented Agile practices within their RUP governance framework and achieved some degree of improvement.

One approach that has worked in a large federal program is the execution of “just enough” release preparation activities and a formal review prior to the start of the first iteration of each Agile release. This approach allows the product owner and Agile teams to make a commitment to stakeholders on the budget that will be spent during the fixed release duration and a commitment of capability that allows for adjustment. At that point, other business and IT stakeholders, such as security, can participate, review “just enough” documentation, and raise concerns. At the conclusion of each iteration, stakeholders participate in a formal review and demonstration of working code, where they can verify that their requirements or standards have been met. Then, a final review is conducted to formally approve the deployment to production.

Success Factor Seven: Selecting Top Staff for Lead Roles in the Agile Project

The rewards of implementing the right Agile approach can be substantial, but will challenge the current state and culture of most government agencies. Like any transformational endeavor, resources with strong leadership skills will be required to:

- Execute and lead the charge
- Broker the necessary changes
- See the vision through from start to finish

Agile project management requires more use of project management “soft skills” and aggressive issue and risk management (identifying and resolving impediments), as opposed to logging and tracking status over weeks via e-mail and updating a Gantt chart. The environment is fast-paced, and all project managers may not be well suited to the change. Agile teams integrate a mix of skills required to deliver demonstrable code, so people will likely take direction from a project manager who is not their direct supervisor.

Experienced staff must be selected to plan, guide, execute, and refine an implementation, commensurate with the work to be executed. A very small, self-contained project, on a proven architecture, with no external dependencies, may be able to hire an Agile coach, participate in a basic Scrum course, and do fine. But when delivering a mission-critical solution for a complex business and technical environment, with a commitment to when releases of that solution will be delivered, planning and executing the initiative through an end-to-end, disciplined, scalable, Agile approach is required.

Success Factor Eight: Planning for IT Infrastructure and Tooling Needs

Implementing Agile practices will challenge how infrastructure is used to support the rapid build, deployment, testing, and release of software from many developers and Agile teams into a production system. This includes Agile Development and Operations (DevOps) practices, and tooling that optimize the build, testing, deployment, and release of software, remove defects from the code earlier, and provide the opportunity to release software to the end users more frequently. Design, planning, and implementation of the functional development products, tooling, and support are critical for Agile teams to become productive quickly. Agencies can access these products as a service within a cloud environment at project initiation to speed the start of the project, while planning the production implementation for a first release.

Security is of paramount importance in the federal government, and is one example where automated tools are available and incorporated into the build process, and configured to identify security risks each time a developer's code is integrated into the solution. This helps to identify issues that might be missed by normal code reviews and provides more frequent opportunities to address those issues in early development iterations.

Success Factor Nine: Conducting “Just Enough” Upfront Work Before the Start of the Agile Project

If multiple Agile teams are initiated without “just enough” pre-work to plan and establish basic high-level requirements, design, architecture, and standards for a first release, it is wasteful. Without some “just enough” preparation, many Agile teams will spend a lot of effort and money communicating across their small teams and defining what they all need prior to starting, as opposed to elaborating on and delivering stories that had been defined at a high level and selected to support the goal of the release. This can also lead to “building the thing wrong” as teams are pressured to deliver a prototype that functions on the surface, but may not be designed to the same standards as another Agile team or to handle the volume of transactions that are required for a first or early release.

By conducting “just enough” work for the release ahead of time, the development teams can be far more productive and attain a more predictable velocity sooner. Defining “just enough” requirements, design, and architecture is a delicate balance. It requires resources who understand the agency’s technical standards and requirements, and can work with the business to continuously weigh the short and long term needs of the mission, as well as the risk and cost of rework at a later point in time. The intent is not to conduct all upfront requirements, design, and architecture, as is prescribed in a traditional waterfall approach, but to define enough of a technical foundation on which functions or features can be built without repeatedly spending money to rethink everything the Agile teams develop.

Facilitated work sessions should be conducted that incorporate other applicable practices (e.g., User Centered Design, Business Process Modeling [BPM], Architecture Modeling) to shape capabilities for a given release, then decompose those capabilities into stories that can be prioritized and elaborated within each iteration of a release by each Agile team.

Success Factor Ten: Integrating Critical Specialty Skills to Support Agile Teams

In general, each Agile team should have a mix of resources with the skills and knowledge required to develop a production-ready solution. However, federal government projects may require niche skills including:

- Organizational Change Management
- Application and Infrastructure Security
- Application Usability (U.S. Federal Section 508 requirements)
- Legal and privacy issues
- Plain language
- Independent Validation and Verification (IV&V) testing
- Training

Most of the above skills will be required for federal contracts, may be in short demand, and if ignored, could result in delays or risk to the agency. For example, application security requirements related to the Federal Information Security Management Act (FISMA) and National Institute of Standards and Technology (NIST) guidance are critical to protect federal government interests. Without the representation of experience with FISMA and NIST guidance across Agile teams, an agency may introduce significant security risk, or may experience delays and rework if those requirements are not considered early in the project.

One way to incorporate niche skills required by all Agile projects is to dedicate subject matter experts across multiple Agile teams. These SMEs can participate in facilitated work sessions to contribute their subject matter expertise to the development of stories, assumptions, and acceptance criteria, as well as other related planning for the program and release. Then, during each Agile iteration, they can participate in the daily standup meetings of each Agile team, have review tasks assigned to them, and execute their tasks as stories are completed.

Agile teams should receive an orientation on select standards by the SMEs who own the standards and work with the Agile teams to assure they are implemented, adapted and automated into tooling in some cases. For

TEN CRITICAL SUCCESS FACTORS FOR IMPLEMENTING AGILE DELIVERY

example, a team should be oriented on application security concerns and standards. Then, during each Agile iteration, lead developers should conduct code reviews or review automated tooling feedback to assure that the standards have been applied; then, a Security SME can validate that those concerns have been addressed before the story is “done.” If standards are deemed onerous, then at the end of each iteration, as the teams conduct retrospectives, they can work with SMEs to formally refine those standards fit-to-purpose.

ACKNOWLEDGMENTS

The authors would like to thank the following individuals from IBM who assisted in the preparation of this report: Mark Bolter, Thomas Coleman, Greg Day, Christopher Egan, Adrienne Fadoul, Lori Feller, Matt Ganis, Michael Hicks, William Mead, Ian Morrison, Darrell Riggs, Jacob Waheed, and Amy Wilson.

ABOUT THE AUTHORS

Paul Gorans is the Agile Competency Lead for IBM Global Business Services, U.S. Federal sector. In addition to growing IBM Agile capabilities, he works with IBM teams and their clients to optimize solution delivery through the pragmatic use of integrated Agile, Lean, rapid, and traditional practices.

Paul has over 22 years of experience delivering and supporting the sale of Application Development and Maintenance services. His diverse background includes delivery roles in systems engineering, project management, and business process reengineering in multiple industries, and Agile sales and consulting across all sectors.

For the past 14 years, Paul has been focused on Agile delivery and has consulted on over 400 Agile engagements. This includes planning and leading complex Agile projects and programs, over a dozen client Agile assessments, and planning/guiding the transformation of a large federal program from a traditional to an Agile delivery approach.



ABOUT THE AUTHORS

Philippe Kruchten is a full professor of software engineering in the department of electrical and computer engineering of the University of British Columbia, in Vancouver, Canada. He holds an NSERC Chair in Design Engineering. He joined UBC in 2004 after a 30-year-plus career in industry, where he worked mostly with large software-intensive systems design, in the domains of telecommunication, defense, aerospace and transportation. Some of his experience is embodied in the Rational Unified Process (RUP) whose development he directed from 1995 to 2003, when Rational Software was bought by IBM. His current research interests still reside mostly with software architecture, particularly architectural decisions and the decision process, as well as software engineering processes, particularly the application of Agile processes in large and globally distributed teams. He teaches courses in entrepreneurship, software project management, and design.

He is a senior member of IEEE (Computer Society), an IEEE Certified Software Development Professional (CSDP), a member of ACM, INCOSE, CEEA, the founder of Agile Vancouver, and a Professional Engineer in British Columbia. He has a diploma in mechanical engineering from Ecole Centrale de Lyon, and a doctorate degree in information systems from Ecole Nationale Supérieure des Télécommunications in Paris.



To contact the authors:

Paul Gorans

Agile Competency Lead—IBM Federal
IBM Global Business Services
600 14th Street, NW
Washington, DC 20005-2012
(313) 574-5973

e-mail: pgorans@us.ibm.com

Philippe Kruchten

Professor
Electrical and Computer Engineering
Faculty of Applied Science
The University of British Columbia
4046 - 2332 Main Mall
Vancouver BC V6T 1Z4
Canada
(604) 827-5654

e-mail: pbk@ece.ubc.ca



REPORTS from
The IBM Center for The Business of Government

Recent reports available on the website include:

Acquisition

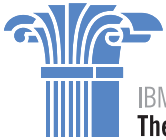
Eight Actions to Improve Defense Acquisition by Jacques S. Gansler and William Lucyshyn
A Guide for Agency Leaders on Federal Acquisition: Major Challenges Facing Government by Trevor L. Brown
Controlling Federal Spending by Managing the Long Tail of Procurement by David C. Wyld

Improving Performance

Incident Reporting Systems: Lessons from the Federal Aviation Administration's Air Traffic Organization by Russell W. Mills
Predictive Policing: Preventing Crime with Data and Analytics by Jennifer Bachner
The New Federal Performance System: Implementing the GPRA Modernization Act by Donald Moynihan
The Costs of Budget Uncertainty: Analyzing the Impact of Late Appropriations by Philip G. Joyce

Using Technology

Realizing the Promise of Big Data: Implementing Big Data Projects by Kevin C. Desouza
Cloudy with a Chance of Success: Contracting for the Cloud in Government by Shannon Howle Tufts and Meredith Leigh Weiss
Federal Ideation Programs: Challenges and Best Practices by Gwanhoo Lee
Rulemaking 2.0: Understanding and Getting Better Public Participation by Cynthia R. Farina and Mary J. Newhart
The Use of Data Visualization in Government by Genie Stowers
Mitigating Risks in the Application of Cloud Computing in Law Enforcement by Paul Wormeli
Working the Network: A Manager's Guide for Using Twitter in Government by Ines Mergel



IBM Center for
The Business of Government

About the IBM Center for The Business of Government

Through research stipends and events, the IBM Center for The Business of Government stimulates research and facilitates discussion of new approaches to improving the effectiveness of government at the federal, state, local, and international levels.

About IBM Global Business Services

With consultants and professional staff in more than 160 countries globally, IBM Global Business Services is the world's largest consulting services organization. IBM Global Business Services provides clients with business process and industry expertise, a deep understanding of technology solutions that address specific industry issues, and the ability to design, build, and run those solutions in a way that delivers bottom-line value. To learn more visit ibm.com

For more information:

Daniel J. Chenok

Executive Director

IBM Center for The Business of Government

600 14th Street NW

Second Floor

Washington, DC 20005

202-551-9342

website: www.businessofgovernment.org

e-mail: businessofgovernment@us.ibm.com

Stay connected with the IBM Center on:



or, send us your name and e-mail to receive our newsletters.