...it’s not a question of whether your firm will adopt Lean and Agile techniques but how soon and how broadly it will adopt them. Firms that charge into the transformation blindly will meet an untimely but fairly predictable end.

— FORRESTER RESEARCH
Table of Contents

4
Survive and Thrive with Agile

5
Why Project Managers Should Go Agile

7
Mapping Waterfall Project Management to Agile Practices

14
Agile at Scale: Program Management

17
How to Get Started
Survive and Thrive with Agile

Agile refers to practices that encourage rapid development of working products, a focus on quality and value, and integrated cycles of feedback.

More and more organizations are adopting Agile methods to help manage complex work and improve project delivery. Agile projects are iterative and incremental, delivering working code and collecting feedback in phases to keep development aligned with changing business needs. Agile projects evolve through collaboration between self-organizing, cross-functional teams.

This guide provides tips for how project managers can use Agile approaches to plan, manage, and deliver high-value work. Use it to learn more about the differences between traditional project management and Agile, find the Agile role that’s best for you, understand the dynamics of Agile teamwork, and get started on the transition to becoming an Agile practitioner.

If you’re a project manager, you may be wondering: How is Agile different or better than traditional project management? Do I have a role on an Agile team, and if so, what is my role? How do I make Agile work for me?
Why Project Managers Should Go Agile

Once considered a fad, Agile has matured into a popular and respected set of development methods.

In fact, you may have seen Agile expanding outside of software development and IT into sectors like banking, management consulting, automotive manufacturing, and healthcare. Companies are moving to Agile methods because the global marketplace demands they bring products to market that better reflect their customers’ needs. Where the traditional “waterfall” approach — with its sequential phases and heavy investment in large-scale, up-front design — lacks the flexibility to respond swiftly to changing markets, Agile approaches offer faster delivery, higher quality, and an engaged development team that can deliver on its commitments.

Agile can improve your efficiency. Instead of designing an end-to-end, upfront solution (a solution that may be infeasible or outdated before it’s even implemented,) Agile teams build the solution incrementally — allowing you to mitigate risks, accommodate changing market needs, and deliver valuable features more quickly. Agile methods have been shown to cut time-to-market by 50% and increase productivity by 25%

According to the most recent global survey from PricewaterhouseCoopers\(^2\) on the state of project management, 34% of you now use Agile PM methods within your companies and a majority of PMs (62%) are certified Agile practitioners.

The PwC survey further comments that “Not only have organisations raised the bar in order to stay competitive in the turbulent business environment, but PM standards have also significantly increased … more practitioners are becoming certified in PM with an increased adaptation of Agile PM and EVM.” Job board Indeed.com reflects this rise in Agile roles, citing a 1,750% growth in Agile-related jobs since 2005\(^3\).

It’s a common misperception that Agile doesn’t need project managers. On the contrary, Agile doesn’t want your job; there’s plenty of work to do, and Agile needs all of you to help do it. The only people in danger of losing their jobs from Agile are those who’ve been hiding within the inefficiency of the system (Agile is remarkably good at holding up a mirror to an organization and exposing its waste, inefficiency, and dysfunction.) What matters most in Agile project management is not your title or your role, but your contributions: what you do to add value and keep the organization moving forward.


\(^3\) Indeed.com trend data, pulled 2013: http://www.indeed.com/jobtrends?q=Agile&l=&relative=1
Mapping Waterfall Project Management to Agile Practices

Agile and waterfall development aren’t as different as people imagine.

Both approaches recognize the triple constraints of cost, schedule, and scope; where they differ is in the implementation.

First, waterfall development encourages locking down scope requirements so that schedule and cost can be planned, and sees feedback as “rework” -- something to be avoided through better planning. Agile, on the other hand, recognizes that scope is always variable, and sees feedback as a critical and inextricable part of a planning process that continues through execution.

Second, Agile eschews waterfall’s traditional directive tactics in favor of collaboration and facilitative support—or what's known as “servant leadership.”

The idea is that if you give teams the tools they need to succeed, help them understand the business value of their problem-solving, and provide them with the space and time to complete their work, the need for command-and-control management disappears.
In traditional project management, you—the project manager—are responsible for balancing scope, cost, and schedule, as well as managing quality, reporting, and interpersonal issues. In Agile project management, the whole team commits to shared decisions and collaborates in its work to meet these commitments. Your Agile project management support equips the team to become fully engaged and motivated contributors, who can produce high-quality work at a faster pace.

Despite the differences between Project Management Institute (PMI) and Agile approaches, many of the practices identified in the Project Management Book of Knowledge (PMBOK) are quite compatible with Agile practices. In fact, when followed

### PMBOK Practices Mapped to Agile Practices

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiating</td>
<td>Envisioning</td>
</tr>
<tr>
<td>Defines and authorizes the project.</td>
<td>Defines the product sufficiently to provide a sandbox with borders in which to work.</td>
</tr>
<tr>
<td>Planning</td>
<td>Roadmap</td>
</tr>
<tr>
<td>Describes how the project will be managed.</td>
<td>Translates the vision into a set of features and an expected timebox in which to deliver them.</td>
</tr>
<tr>
<td>Executing</td>
<td>Release</td>
</tr>
<tr>
<td>Helps the project groups work together to complete the work.</td>
<td>Helps the team incrementally and iteratively develop potentially shippable code.</td>
</tr>
<tr>
<td>Monitoring and Controlling</td>
<td>Adapting</td>
</tr>
<tr>
<td>Checks the progress of the project and corrects problems.</td>
<td>Integrates planned stopping points to inspect and adapt the process and product.</td>
</tr>
<tr>
<td>Closing</td>
<td>Closing</td>
</tr>
<tr>
<td>Formally closes each phase or the project and receives approval of the project work.</td>
<td>Team reflects on achievements and decision-making per lessons learned.</td>
</tr>
</tbody>
</table>
with discipline and rigor, Agile methods are just as compliant with the Capability Maturity Model Integration (CMMI) as traditional waterfall methods. The differences lie in when and how these practices are executed and the lexicon used by their practitioners.

The PMBOK identifies Initiating, Planning, Executing, Controlling, and Closing as the process groups within project management. The Agile process phases of Envisioning, Roadmap, Release, Adapting, and Closing are similar to the PMBOK phases, but better reflect the reality of how software is actually developed.

These Agile project phases also map nicely to an iterative development environment, scaling from short iterations up to longer-term releases.

PMBOK identifies nine key knowledge areas: Integration, Scope, Time, Cost, Quality, Human Resources, Communications, Risk, and Procurement. Let’s take a look at how some of these knowledge areas map to Agile practices.

**Integration Management: A Collaborative, Iterative Process**

<table>
<thead>
<tr>
<th>Integration Management Process</th>
<th>Agile Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>Agile</td>
</tr>
<tr>
<td>Develop project charter or plan</td>
<td>Develop roadmap and backlog</td>
</tr>
<tr>
<td>Execute the project plan</td>
<td>Do iteration work</td>
</tr>
<tr>
<td>Direct, manage, monitor, control</td>
<td>Facilitate, lead, collaborate</td>
</tr>
<tr>
<td>Integrated change control</td>
<td>Constant feedback, ranked backlog</td>
</tr>
</tbody>
</table>
In integration management, a key deliverable is the project charter prepared and owned by the project manager. In Agile development, with its emphasis on just-in-time design and participatory decision-making, planning becomes a collaborative, iterative process. The entire Agile team creates, owns, and commits to developing a roadmap and backlog, and the envisioning process and its results are shared with all stakeholders in the most visible possible way. The Agile project manager facilitates the team in its ongoing, iterative planning and facilitates the communication of important information.

In keeping with the idea of minimum process to achieve maximum value, rigorous change control is integrated into the daily routine of Agile teams. Product changes are managed using a ranked backlog of features, with features that provide the most business value to the customer ranked highest. During release and iteration planning, the highest-ranked items move from the backlog into iterations, to be coded, tested, and then accepted by users. Feedback-gathering and process enhancements are integrated into each iteration cycle to improve not only the product but the way the stakeholders work together. The team—customer, developer, tester, analyst, technical writer, project manager—becomes the equivalent of a change control board, doing collaborative, continuous planning at a regular cadence.

**Scope and Time Management: Fixed Timeboxes**

“Scope creep” has always been the bane of traditional project managers. Plan-driven approaches work hard to prevent changes in scope, even as customer and business needs evolve; but Agile approaches expect and embrace scope change.
The most common question asked by project managers regarding this facet of the Agile approach is, “What am I supposed to tell my supervisor/customer about when this project will be finished?”

Remember: the Agile strategy is to develop within a timebox of fixed schedule and cost, so that the highest-value features for the customer can be implemented.

This is the area of Agile that tends to fluster project managers the most. The box that we use is still time; however, instead of stuffing more feature “bricks” into a single, flimsy box until that timebox explodes, we’re now using multiple timeboxes made of steel, and we stop dropping in bricks when the box is full. We then close the box, padlock it for the entirety of this iteration, and work the features through the acceptance. Because we’re doing the work one timebox at a time, it’s difficult to understand how much work will be completed over a longer timeframe.

As a project manager, you are not a fortune-teller: regardless of your PM approach, you can only make educated guesses based on judgment and historical analysis. The key difference in Agile development is that scope definition is done as part of iteration planning, where features are tasked out, estimated, and assigned. As product owners and customers get to review, test, and accept the implemented features, the original project scope can be checked against accurate progress metrics and communicated to stakeholders.

**Quality Management: Integrated from the Start**

Quality assurance and control staff are accustomed to being the caboose on a runaway train: they expect shortened time frames for testing, specs that don’t match the product delivered, and little interest in their input until the final weeks of the project. Agile development brings QA into the process at the very beginning with analysis and design of
the product, and keeps them involved in decision-making throughout the entire lifecycle. With incremental code development, QA is testing at the beginning of the project instead of waiting for something to be “thrown over the wall.”

In Agile practices, everyone contributes to defining, maintaining, and enhancing the quality of the product. Developers contribute to testing by writing unit tests and helping with the framework for automating regression and acceptance testing; product owners help to define and run acceptance tests. To prevent defects, quality assurance members participate in the development team decision-making on a daily basis: their input helps developers write better code.

To control quality, bug-checking is done within the iteration. In Agile development, you will all take responsibility for ensuring that the features being coded are high-quality and meet the customer’s expectations.

**Human Resources Management: Self-organizing Teams**

The PMBOK defines human resource planning as the process where roles and responsibilities are assigned, and it defines team development as the development of each team member's appropriate skillset and competencies. The Agile equivalent to this is to establish cross-functional teams, and allow them to self-organize.

Establishing a cross-functional team in Agile means building a team comprised not just of programmers, but all the key players needed to create an increment of working code: testers, analysts, architects, technical writers, subject matter experts, customer/product owners, and the team lead (project manager, Scrum Master, etc.) Each individual brings a particular skillset, but as the team matures each member will gain an understanding about the others’ tasks and efforts. As a result, the team is better able to take responsibility for
### Human Resources Management

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Resource Planning</td>
<td>Dedicated team of 7 +/- 2 members</td>
</tr>
<tr>
<td>Team Development</td>
<td>Cross-functional, collaborative teams</td>
</tr>
<tr>
<td>Team Management</td>
<td>Servant leadership, self-managing teams</td>
</tr>
</tbody>
</table>

the quality and speed of its progress. The team’s total ownership of planning, execution, and review leads to self-directed performance that’s maintained through continuous reflection and improvement.

Traditional project managers sometimes struggle to grasp this concept of collaborative ownership and self-directed teams. Here is where the shift from command-and-control management to “servant leadership” comes into play. Becoming a servant leader involves learning how to foster collaboration and reflection, providing guidance and facilitation, and removing roadblocks so that your team can do the best work it possibly can.

Team dynamics don’t change overnight, and teams new to Agile benefit from strong leadership and well-trained managers.
Agile at Scale: Program Management

For many of you, the shift from traditional to Agile project management practices is part of a cultural shift within your organization.

Regardless, framing the shift as a “waterfall vs. Agile” debate is polarizing and unnecessary. The shift should be seen as a continuum on which organizations can adopt Agile at a speed that works best for them, and can scale their Agile practices in a customized way.

Successfully adopting Agile at the enterprise or portfolio level means making organizational changes in process, structure, people, and metrics around program management—where projects are approved, prioritized, allocated, and reported on. Program managers have an essential role here, serving as liaisons between divisions, negotiating compliance with best practices among self-managing teams, and acting as the communications channel to stakeholders. Program managers can assist in identifying the right platforms and processes for reporting on project status, metrics, and compliance, so that useful information can be delivered to executive management in a way that’s not unfairly burdensome to development teams.

Some of you will get your feet wet with Agile by piloting just a single team, while others will implement Agile at the portfolio level, across your entire organization with hundreds of development teams.
In large organizations that adopt an enterprise Agile framework like SAFe™, program managers perform the critical function of learning what changes need to be made at the enterprise level and working with people and processes to get these changes implemented. Program managers can help keep a strict eye on the difference between the product requirements and the project requirements, managing the project to the scope schedule or budget as mandated by the project sponsorship. They also can help the organization address ongoing needs for training and performance evaluation.

The transition from traditional approaches to Agile approaches takes time, and program managers can help by preparing themselves and others for the transition.

Despite the fact that your role may change as your organization shifts its cultures to adopt Agile practices, your contributions as a program manager remain vital.

You can exhibit servant leadership by facilitating teams in self-management, acting as keeper of Agile process and practices, and communicating clearly both up and down the chain.
The question isn’t who can and can’t be Agile. Everyone can. The real question is: How can I make Agile work for me and my organization?

— AGILE COACH JULIE CHICKERING
How To Get Started

Project managers willing to make the change to Agile servant leadership will find the journey exciting and rewarding.

With the leadership skills to facilitate self-managing teams and guide them with the right questions at the right time, you can watch your teams produce high-quality, collaborative work and feel proud of their achievements. All the other skills needed to excel in Agile project management—guiding teams through process decisions, scheduling meetings, sharing information, and providing the tools and environment for great work—are already part of your project manager’s responsibilities today.
Step One:
Learn the Agile Process

The first step in transforming to Agile project management is to learn the Agile process. Whether you’re doing Scrum, Extreme Programming, or another style of Agile development, you will become the knowledge repository for your team’s practices and principles. Team members will look to you for input when they have process questions and for understanding when they have questions or concerns. Self-educating with books and articles on Agile methods will enable you to respond thoughtfully to your teams and guide them successfully on the Agile transformation path.

Other Helpful Resources

Internet

http://www.agilemanifesto.org Read the Agile Manifesto for more about the Agile movement’s origins, principles, and how its founders came to develop a new approach to development work.

http://www.agilealliance.com The Agile Alliance provides a vast library of articles and will help you stay up-to-date on current trends.

http://www.apln.org The Agile Project Leadership Network

http://www.scrumalliance.org Scrum Alliance

http://www.rallydev.com/resource/build-your-agile-business-sourcebook Rally’s definitive guide to scaling Agile from the Team, to the Program, to the Portfolio level, packed with over 30 pages of practical content, agendas, and real-world advice from experts and your peers.

Traditional Media

The Software Project Manager’s Bridge to Agility, by Michele Sliger.


Agile Project Management: Creating Innovative Products, by Jim Highsmith.

Agile Estimating and Planning, by Mike Cohn.

The Power of Servant Leadership, by Robert Greenleaf.
Step Two:
Find Your Role

Far too many organizations cripple their Agile teams by uniformly assigning those with a given traditional title to an Agile role they assume is roughly equivalent. Some people are wildly successful in these new assignments; others are abject failures; while still others are left wondering if they still have a job, since Agile doesn’t assign a role for their traditional title.

It’s much more valuable for both you and your organization to look at the skills, capabilities, and inclinations that make individuals successful in Agile roles, and which traditional titles these people often held. For example:

You might be a Product Owner if...
You are good at working with stakeholders, managing expectations, and balancing competing priorities; you’ve got a strong sense for business priorities and value; you’re comfortable driving the vision and strategic direction for a product or system while simultaneously collaborating with the team to help make day-to-day tactical decisions.

Your traditional title might be: Product Manager, Business Analyst, Project Manager, Application Manager, Marketing Manager, Solution Architect, etc.

Ultimately, Agile values contributions over roles, and roles over titles. A title shouldn’t be the determining factor in deciding roles on an Agile team, or how individuals add value for that team. What matters are your unique skills, capabilities, and inclinations.
You might be a Scrum Master if...
You understand people – their motivations, fears and desires; you understand group dynamics – what makes a team better (or worse) than the sum of its members; you tend to focus on the long-term – in other words, developing the capability to deliver – rather than on delivering the priority or fire of today; you’re part mentor, part counselor, part teacher, and coach.

Your traditional title might be: Project Manager, Technical Lead, Development Manager, QA Manager, Business Analyst, QA Analyst, Release Manager, etc.

You might be a Delivery Team Member if...
You add value to a product or solution directly; you must collaborate with others to effectively build and deliver a product or solution; you can commit yourself, personally and professionally, to building and delivering that product or solution; you write code, write documentation, analyze, test, integrate, deploy, design, architect, etc.

You might be a Stakeholder if...
You’re a customer, an end-user, a business leader, a technology leader; you have architectural, compliance, regulatory, legal, security, or other oversight responsibilities; you will support, market, sell, integrate, train, or otherwise be impacted by the product or solution delivered.

Your traditional title might be: Product Manager, Sales Executive, Security Analyst, Compliance Officer, Marketing Manager, Development Manager, Lead Counsel, Technology Architect, Enterprise Architect, etc.

You might be a Specialist if...
You have a skill set that is unique to a small number of people in the organization (relative to the number of Agile teams); many Agile teams will rely on your specialized work to build and deliver their product or solution; no one team will regularly have enough of your specialized work to require your dedicated involvement.

Your traditional title might be: DBA; Database Architect, Security Specialist, BI/DW Engineer, etc.
Step Three:
Get Training and Certifications

Most people are unaware of the nuances of good facilitation and believe they already know how to do it well; however, the directive, command-and-control style of leadership can prevent teams from achieving cohesion and participating in collaborative decision-making. Becoming skilled in how to properly facilitate team discussions is an art and a talent: one that can be learned, and one that’s vastly underestimated in ensuring the success of a team. A well-trained facilitator is a guiding force in helping teams share information, hear that information clearly, express their opinions, define their options, and reach consensus.

Other Helpful Resources

Internet

http://www.agileu.org Agile University provides training and certification in Scrum, facilitation, and other useful skillsets.


http://www.scrumalliance.org The Scrum Alliance provides training, certification, and peer support.

http://iaf-world.org/index.aspx International Association of Facilitators provides training, certification, and a coaching directory.

Traditional Media

http://www.iaf-world.org The International Association of Facilitators.

Collaboration Explained, by Jean Tabaka, explains the how, why and what of good facilitation.

Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects (Pragmatic Programmers), by Johanna Rothman.

Liftoff: Launching Agile Teams & Projects, by Diana Larsen.
Step Four:
Get Coached Through the Transition

An uncoordinated approach to Agile can result in disconnected teams, silos, project failures, and other risks. Once you have educated yourself on the principles and process of Agile, identified your role, and sought out training and certifications, you may want to bring in outside help to ease the transition for you and your organization.

Other Helpful Resources

Internet

http://www.rallydev.com/services/agile-coaching-and-training-services Rally Software provides coaching and consulting services to guide you through the process of a planned, coordinated, scaled transformation.

http://www.agileleadershipnetwork.org/local-chapters The Agile Leadership Network provides resources for training and coaching.