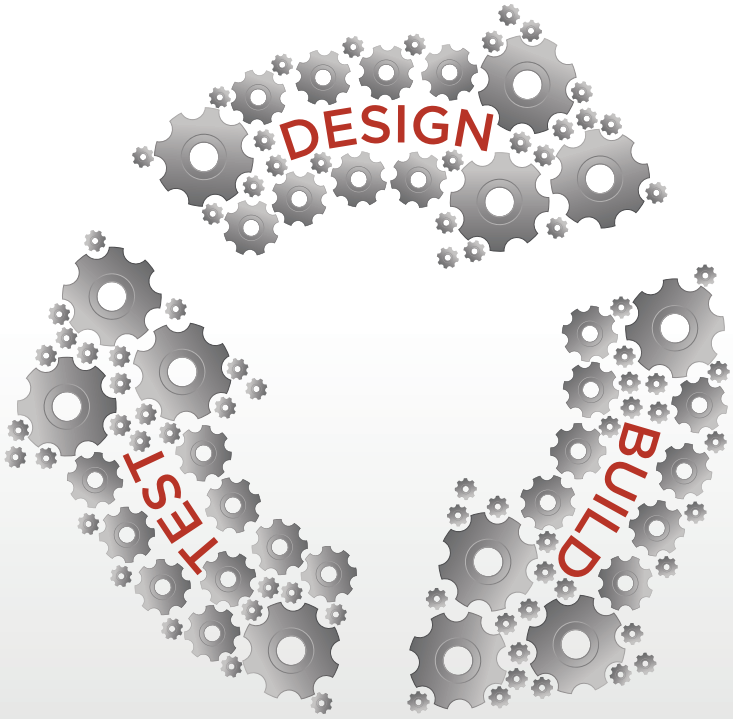
RMC Learning™ Solutions

TOP 10 Agile Techniques Every Team Should Know



A White Paper from
RMC Learning Solutions
www.rmcls.com

RMC Learning™ Solutions

10953 Bren Road East, Minnetonka, Minnesota 55343, USA

Main +1 952.846.4484

Fax +1 952.846.4844

E-mail info@rmcls.com

Table of Contents

Meet The Author

Barbara A. Carkenord, CBAP®, PMP®, PMI-PBA®, PMI-ACP®

Practice Director and Trainer—Business Analysis

As the Director of Business Analysis at RMC Learning Solutions, Barbara combines her entrepreneurial and management experience with her love of education and business analysis to promote the development of the business analysis practice. During her career Barbara co-founded two successful companies and worked in varied industries including manufacturing, financial services, and software development. Barbara has worked as a leader, manager, mentor, consultant, trainer, and technical writer. She has written numerous books, articles, blogs, and training manuals all aimed at helping other professionals enhance their skills. In 2010, she was named the Small Business Woman of the Year by the Georgia Women in Technology. In 2016 Barbara was elected to the International Board of Directors for the International Institute of Business Analysis (IIBA).

Throughout her career Barbara has been passionate about enabling people and organizations to succeed through analysis. Analytical thinking allows organizations to increase their process efficiency and improve the quality of their products. Beginning her career in software development, early assignments allowed her to design systems which streamlined employee processes, increased the quality of information, and improved customer relationships.

Barbara's focus on critical thinking and increasing business value drove her to help define the business analysis profession. As an early IIBA Member, she worked on the development of the worldwide standard for business analysis, the BABOK® Guide.

Barbara A. Carkenord,
CBAP®, PMP®, PMI-PBA®, PMI-ACP®

Connect with Barb

twitter.com/bcarkenord
linkedin.com/in/bcarkenord
facebook.com/bcarkenord

Introduction

Teams are the building blocks of successful organizations. Teams build new products and design changes that move the organization toward its long-term goals. Any team's productivity can be improved by incorporating some of the agile techniques and tools. Many agile techniques and practices are not new. They are based on existing techniques from other methodologies and processes and have been revised and sometimes renamed by agile. This article will focus on ten agile techniques and philosophies that can provide value for every team. Many teams are already practicing some of these techniques. Most people have heard about agile techniques like daily standups, but there are lots of other techniques that are not yet well known. Some of them are new and will require further education and coaching. Anyone working on a team should learn agile fundamentals as one of his or her core competencies, as agile practices can be used to help improve team productivity in solving business problems and exploiting opportunities.

Even if your organization isn't ready to embrace a pure agile process, there are many small changes a team can incorporate to increase its effectiveness.

Some people will argue that a team must incorporate all of the agile practices and principles to be successful and accurately claim to be "agile." Even if your organization isn't ready to embrace a pure agile process, there are many small changes a team can incorporate to increase its effectiveness.

To use this selective approach, the team must understand agile principles, the technique being used, and how it might be adapted for a particular situation. Setting management and stakeholder expectations for results is also critical. Simply employing a couple of agile techniques will not change the delivery schedule or outcome of your current project! But moving your team towards more agile practices will increase team cohesion, clarify stakeholder expectations, and increase the value of the team.

Know How and When to Use Each Technique

Excellent teams understand that agile is a *product* development approach often used for software development. A *product* is something tangible, which will be useful to its customers. Agile is great for software development projects where new software is a standalone product. When the business solution requires a change to existing software with numerous complex interfaces, agile will be more challenging to apply. When a business problem is best solved by a process change, an agile approach is probably not applicable.

Agile is not a business modeling or program management tool. Agile approaches assume that the business stakeholders already understand the root cause of their problem or the characteristics of an opportunity and that they need a *product* to be developed. Business models, business analysis, and business cases must be developed *before* an agile team starts working on a product. Sponsors, project managers, program managers, business analysts, developers, and all project professionals need to consider the characteristics of their projects to determine when and how agile techniques will be appropriate.

The Ten Techniques

1. Servant Leadership
2. Deliver Quick Business Value
3. Product Vision
4. Planning
5. Small Scope And Time-boxes
6. Prioritize Requirements
7. Fast Failure
8. Barely Sufficient
9. Visualize Results
10. Sustainable Pace

Overcome Reluctance

Agile approaches focus on team productivity rather than individual contributions. Team members may be fearful that being rewarded as a team, rather than as individuals, puts their success in the hands of others. Some team members may worry they won't be able to work collaboratively. Others will fear their work being exposed before they are finished and be concerned about the shorter time periods of sprints. The key to overcoming natural reluctance to change is knowledge and experience. Once team members try a technique in a safe environment, they will be able to determine its value. An important tenet of agile is team self-management. The team will be able to decide if a technique works well for them. Well-trained, enthusiastic team coaches and leaders assist the team members as they adapt the technique to their team. These *servant leaders* (Technique #1) will be the key to the success of any agile implementation.

1. Servant Leadership Teamwork and Trust

The concept of servant leadership shifts power from the team leader to the team members, focusing on collaboration and teamwork. Effective teams collaborate to keep each other informed of progress, challenges and decisions. Open communication ensures the development of product components that will work together and be consistent. Frequent collaboration with product owners and business stakeholders ensures the product development supports the underlying business need and allows product users to suggest adjustments to the product design during development.

Team members choose their assignments from the task list and commit to completion dates.

An underlying assumption of the agile manifesto is that a knowledgeable, trusting team can together produce a product faster and of higher quality than individuals working on their own. Servant leadership implies that agile teams can organize and manage themselves with the guidance of a coach or mentor. The leader acts as a servant to the team, providing suggestions and helping team members to develop productive, supportive team habits, norms and behaviors.

For example, instead of a project manager assigning tasks to team members, team members choose their assignments from the task list and commit to completion dates. If a team member is not keeping his or her commitment to the team, the coach helps the team discuss the problem and come to a team solution. Dealing with issues as a team helps team members build better communication and conflict resolution skills. This collaborative approach has been used for years by many project managers who have discovered that when people choose their own work and set their own deadlines, they are more successful.

The term *generalizing specialists* is used to describe successful agile team members. This agile phrase is a little tricky; many students jump on the word *generalizing* and assume everyone on an agile team needs to be able to do everything. But the focus is on both words and their meaning together. Agile team members are specialists. Each member has been assigned to the team because he or she has a unique and necessary skill. For example, one team member knows how to design and build XML files. Another member is selected because he knows how to use automated testing tools; another because she is an expert in product design. These experts are also expected to generalize, which means they can help each other as needed. If the testing of a feature is more complex than expected, the testing specialist may ask another team member to help with executing test cases or looking over test results. Generalizing means team members are ready and willing to help in whatever way they can, for the good of the team.

1. Servant Leadership Teamwork and Trust *(continued)*

Organizations that have traditionally rewarded individual accomplishments and promoted people based on individual output will have a difficult time suddenly expecting employees to work as a team and trust each other. This is the number one stumbling block to success in agile approaches. According to Version One's most recent survey on the *State of Agile Development*, two thirds of the failures of agile initiatives are a result of failure to integrate the right people or teach a team-based culture. Training and practice are essential to enable this transition. To read more about this survey, go to: http://www.versionone.com/state-of-agile-survey-results/

Team members will have more responsibility to manage their own work and participate in overall product completion by the team.

For teams to adopt the servant leadership model, team members must understand the differences from traditional project management. Team members will have more responsibility to manage their own work and participate in overall product completion by the team. This requires each team member to have the courage to question other team members about their progress. For team members to have open communication demands a high degree of trust. Building trust and a collaborative work environment is not something that happens overnight.

New teams working in an agile environment need time to adjust and experience this new mode of operation. The short sprints characteristic of agile give new teams the opportunity to learn to work together on small deliverables, adjusting their practices and improving their productivity on each sprint by conducting retrospectives. Agile principles emphasize stable teams. Once a team learns to work well together, management should try not to change team membership unless requested by the team itself.

For organizations and teams that have been working collaboratively on traditional projects, agile is a welcome acknowledgement of their best practices. Almost every business solution involves multiple stakeholders in its design and development. The agile techniques related to improving stakeholder involvement and collaboratively making prioritization decisions are techniques that can often be adapted to various types of projects and products.

Any team can benefit from these practices. Improving communication skills, conflict resolution skills and getting comfortable with team accountability builds strong teams. It also builds strong individuals who will be more productive and eventually be able to coach other teams.

To learn more:

1. Greenleaf, Robert K. *Servant Leadership: A Journey into the Nature of Legitimate Power and Greatness* 25th Anniversary Edition Hardcover – Deluxe Edition. Paulist Press, 2002.

2. Greenleaf, Robert K. *The Power of Servant Leadership.* Berrett-Koehler Publishers, 1998.

3. Adkins, Lyssa. *Coaching Agile Teams: A Companion for Scrum-Masters, Agile Coaches, and Project Managers in Transition.* Addison-Wesley Signature Series (Cohn) Kindle Edition, 2010.

2. Deliver Quick Business Value

Probably the most appealing aspect of agile, to executives and business managers, is the quick delivery of business value. The two-week sprint implies products will be delivered very quickly. But, as we all know, there is no magic that can instantaneously speed up development time. Delivering quick value is accomplished through scoping and prioritization, not by asking the team members to work more hours!

In agile, "Quick Wins" are the norm!

In many traditional organizations, team members are not encouraged to ask "Why am I building this?" or "What value does my work bring?" In agile, the question is asked at every planning activity and every prioritization discussion. How can we bring the most value to the business in the shortest amount of time? These have traditionally been called "quick wins" and rather than being the exception, in agile they are the normal course of business.

To provide true business value, at least one team member must have an intimate understanding of the business. This should be the product owner, assisted by a business analyst if there is one on the team. Although agile approaches do not specifically call for the involvement of a business analyst, or define the role of a business analyst, those skills are critical to success using an agile approach. Business stakeholders will request hundreds of features and capabilities, each one important to the requestor. The product owner and analyst need to ask questions, gather metrics, and do the research necessary to determine which requests will bring the most business value to the organization as a whole, rather than to a single user.

To deliver business value quickly, everyone on a team must be constantly monitoring the business value of each feature.

Business value can be measured by increased revenue, decreased costs, increased customer satisfaction, or any other factor that supports the organization's vision and goals. Every request has perceived business value or benefits to the requestor. The product owner's job is to evaluate the benefits to one stakeholder group against the benefits to other stakeholder groups.

To deliver business value quickly, everyone on a team must be constantly monitoring the business value of each feature. Team members must stay focused on building and delivering the features that have been prioritized for the current sprint. Team members need to understand the business value of each request and let the team know if development of a feature is taking longer than expected. By talking with the product

2. Deliver Quick Business Value *(continued)*

owner and collaborating on the product vision, team members have tangible knowledge of how each feature will help the customer and will be able to determine if adjustments to the feature during development will negatively impact the expected value.

Of course, the anticipated benefits of each new feature must be weighed against the costs of the new feature. This cost-benefit analysis at the feature level is a critical success factor in making prioritization decisions. Cost-benefit analysis at the feature level is not easy. It requires a deep understanding of the business processes and the organization's short- and long-term goals. A feature that would save time for one group of employees might cause problems for another group in a different business unit. Product owners and business analysts need to perform interface analysis, impact analysis and customer satisfaction analysis. Understanding how each feature impacts the business requires a clear product vision, our next technique. Delivering business value faster is achieved through small scope (Technique #5 Small Scope and Timeboxes) and accurate prioritization (Technique #6 Prioritize Requirements).

To learn more:

1. Pichler, Roman. *Agile Product Management with Scrum: Creating Products that Customers Love.* (Addison-Wesley Signature Series (Cohn)) (Kindle Locations 8-10). Pearson Education. Kindle Edition, 2010.

2. Cohen, Greg. *Agile Excellence for Product Managers: A Guide to Creating Winning Products with Agile Development Teams.* Super Star Press Kindle Edition, 2010.

3. Gottesdiener, Ellen and Gorman, Mary. *Discover To Deliver: Agile Product Planning and Analysis.* EBG Consulting, 2012.

3. Product Vision

Software developers who champion agile do so because they want to be a part of improving the business and designing new products. Frustrated by traditional requirements, which tell them exactly what to build without allowing them to be creative, agilists want to be involved in helping to envision the product from the beginning. A clear product vision is a product description that not only describes what the product should look like, but also defines why the product is important to achieving the business goals. Product visioning is a collaborative activity that includes all team members and business people in the initial design of the product idea. Business people describe their problem or opportunity to the team. They describe the product as they imagine it and allow the team to ask questions and make suggestions about how the product might be designed and built. The product vision also includes the high-level benefits of the product.

The Product Box is one of many ways to depict product vision. It combines the creativity of the developers with the business knowledge of the domain SMEs to present a vision for the new product. This is a good example of an agile, simple style of collaboration. A child's cereal box is the inspiration for this technique, which asks the team to work together to design the "product box." Just as a cereal box has an attractive photo or graphic and headlines that entice the customer to consider purchasing it, the product box gets a product name, picture and headlines or highlights of its features. Tastes great, free prize in every box, etc. on the front of the box encourages a potential customer to turn the box over and read more on the back. The team writes a list of more detailed features on the back of the box, which will convince the customer to buy. The high-level project vision is broken down into potential features of the system, which are then prioritized based on business value and risk.

With a strong facilitator (often a project manager or business analyst), product visioning sessions educate team members on user needs and business priorities in addition to refining the description of the product. Technical team members learn about customer desires from the business people and get to suggest creative, unexpected features which often delight the business people. They learn business terminology and priorities. Business people get to know the technical team and feel their enthusiasm for helping improve the business. A positive product visioning session acts as a strong project kickoff activity, motivating and empowering the team to work closely together during the entire product development process.

To learn more:

1. Pichler, Roman. *Agile Prouct Management with Scrum: Creating Products that Customers Love.* (Addison-Wesley Signature Series (Cohn)) (Kindle Locations 8-10). Pearson Education. Kindle Edition, 2010.

2. Cohen, Greg. *Agile Excellence for Product Managers: A Guide to Creating Winning Products with Agile Development Teams.* Super Star Press Kindle Edition, 2010.

3. Gottesdiener, Ellen and Gorman, Mary. *Discover To Deliver: Agile Product Planning and Analysis.* EBG Consulting, 2012.

4. Pixton, Pollyanna, Nickolaisen, Niel, Little, Todd and McDonald, Kent. *Stand Back and Deliver: Accelerating Business Agility.* Addison-Wesley, Pearson Education, 2009.

4. Planning

Here's one technique that agile and traditional approaches have in common: planning. You may be surprised to learn the importance of agile planning. It is an aspect of agile that doesn't get much attention. Agile planning is different from traditional planning in a couple of key ways. Agile planning does not result in long, detailed planning documents; agile plans are often represented with post-it notes on a wall. This informal representation of the plan does not make it less valuable or necessary.

Agile planning is an iterative activity, rather than being completed early in the project. High-level planning is performed with the team after the product vision is agreed upon. More detailed planning is performed at the beginning of each release, and at the beginning of each sprint. You can think of these plans as being at various *levels of abstraction*. Agile planning is based on the principle of rolling wave planning and estimating, which is also used by many traditional project managers. This approach acknowledges that we can only see a short way into the future, so detailed planning should only take place for the near-term work. Longer-term plans are high-level and will be refined as the project progresses. This approach to planning is the characteristic of agile that allows the team to respond to business changes during the product development. The team plans as much as it can at the beginning of an assignment and replans as requirements are detailed and product components are developed.

Another difference in agile plans versus traditional plans is that agile plans are built around specific time periods, with scope being the variable. For example: Release 1 will be implemented in two months. The items in release 1 will be prioritized and completed based on team velocity (how much work the team can likely deliver in an iteration). The release dates don't change, but the features included in the release may vary based on team velocity and prioritization decisions. This gives business stakeholders clarity about when to expect the product and allows them to participate in deciding which features are included in each release. Setting these firm dates, or *timeboxes*, is the next technique we'll talk about.

The release dates don't change, but the features included in the release may vary based on team velocity and prioritization decisions.

To learn more:

1. Highsmith, Jim. *Agile Project Management: Creating Innovative Products* (2nd Edition) Agile Software Development Series. Pearson Education. Kindle Edition, 2009.

2. Cohn, Mike. *Agile Estimating and Planning.* Pearson Education. Kindle Edition, 2007.

5. Small Scope and Timeboxes

Iterative or incremental development and delivery is a radical change for some business people who can't imagine getting half of a product. But many products can be delivered in pieces or in increasing levels of complexity based on business value. Websites are the best example of this type of product and one of the most common products developed using an agile approach. A new website can be delivered with limited functionality and content. While the simple website is providing a basic level of service to customers, enhancements and additions can be added and layered on top of the original site, providing more functionality. But to implement a product in pieces requires significant creativity and a deep understanding of customer needs. Will one feature be useful if related features are not yet available? How will the customer be supported for processes not yet built? These can be complex decisions the team needs to agree upon when determining what can be delivered and when.

Agile turns the traditional scope-time relationship upside down.

In traditional approaches, scope definition is very important; we clearly delineate what will be included in the final product. Scope definition is often used to determine the project timeline – how long will it take for the product to be built. Agile turns this traditional scope-time relationship upside down. Rather than deciding which features to include in the product, the team decides first *when* to deliver. Two week sprints are usually compiled into a release plan. Once the time period is agreed upon (Example: Six weeks for release 1), then features that can be built and tested within the timeframe become the planned scope. Here's the big philosophy change: scope can change, time cannot. This is a tough concept for traditional project managers who rigorously enforce change control processes to adhere to scope and business analysts who want to define all requirements for the entire product upfront.

Agile teams combine the concept of smaller scope with the technique of timeboxing to manage their work. Timeboxes have been used successfully for years in other professions. Knowledge workers like lawyers must be ready for a case by the court date – few delays are allowed. Writers work to a due date given by their publishers. When the due date arrives, the product is done (even though the writer could write more). Most writers will admit they would never finish anything if not given a firm due date. Software developers are similar to writers. They can always think of more features to add and more creative ways of developing software. This is why most traditional software

5. Small Scope and Timeboxes *(continued)*

development projects miss their planned rollout dates. Giving the team a firm deadline and asking them to get as much work as possible done in that time period makes for a more reliable delivery schedule and helps develop a team rhythm. It also improves stakeholder satisfaction.

The reasons for this philosophy are very simple. Stakeholders remember time delays more than they remember scope changes, and business needs change frequently as time passes. Think about projects on which you have worked. When they are delivered late, users remember. But users don't always remember every minor feature that was scoped out to get the work done. Therefore, it is best to complete the most critical items first. This may be a difficult change for traditional teams who have worked on large, comprehensive projects. The team will need to be creative, imagining smaller, implementable product features (sometimes referred to as minimally marketable features, or MMF). Quick delivery of value is an important goal of everyone in the organization and agile accomplishes this by challenging the traditional focus on a firm product scope.

Another benefit of this approach is that at any time, after any release, the business may decide to stop product development. Once the most important, highest value features have been delivered, users may see no need to build all of the other functions originally desired. The cost-benefit analysis for those lower priority features may no longer justify the expense, in which case, they would rather have their product development team move onto another product.

To learn more:

1. McConnell, Steve. *Rapid Development: Taming Wild Software Schedules.* Microsoft Press, 1996.

2. Martin, James. *Rapid Application Development.* McMillan Publishing Company, 1991.

3. Carkenord Barbara, MBA, CBAP. *Seven Steps to Mastering Business Analysis.* USA: J. Ross Publishing, 2009.

6. Prioritizing Requirements

Prioritizing requirements is not really a new technique. Project managers and business analysts have always tried to get business stakeholders to identify the most important features. Every requirement or feature is not equal. Some are more important, more valuable to the business, than others. The key is getting the business stakeholders, and most importantly, the product owner, to acknowledge this and to decide which features should be built first.

Business people hate prioritization. They never like being forced to choose between two requirements and agile doesn't change this reality.

Business people hate prioritization. They never like being forced to choose between two requirements, and agile doesn't change this reality. Experienced business analysts spend much of their time helping business stakeholders understand tradeoffs and make tough choices. (Again, this is one of the reasons an analyst, or someone with business analysis skills, should be a required member of every team.) Convincing business people to choose one feature over another requires trust in the process and trust in the team. This is difficult because over the last thirty years in software development projects, business people have seen project teams miss deadline after deadline. The business people have reacted by trying to get everything into the current phase, fearing the next phase might be years away. Successful teams will have to prove themselves able to deliver quickly on their commitments, and only then will business people be comfortable selecting the features which should come first. This will take time, constant communication and transparency about the team's process. (Technique #9 Visualize Results.)

Agile teams focus prioritization at a more detailed level than some traditional approaches. Stakeholders are asked to prioritize product features, rather than higher level business needs or processes. This is because the agile team assumes the business sponsor has already analyzed business models and requested a product to address a higher level need. Rather than asking business people "Would you rather be able to respond to customer requests in two days or report to your management weekly?", an agilist asks "Would you rather the system send your customers an e-mail message or a text message?" These smaller prioritization decisions are easier to make and quicker to implement. But they are only easy to make if the stakeholders have an overriding vision of the business needs and the product architecture. When the business stakeholders and the team have clear direction and understanding of the business needs, small prioritization decisions are easier and lead to big value. This has always been a focus of business analysis, understanding the true business need before designing a solution. Agile uses a product backlog to track requested features and capture prioritization decisions. This simple list is maintained for the life of the product as enhancements are identified and inserted at the appropriate priority level.

To learn more:

1. Carkenord Barbara, MBA, CBAP. *Seven Steps to Mastering Business Analysis.* USA: J. Ross Publishing, 2009.

2. Gottesdiener,Ellen. *Requirements by Collaboration: Workshops for Defining Needs.* Addison-Wesley, 2002.

7. Fast Failure

Fast failure? Why would I want to fail fast? It takes a moment to grasp how these two words go together, but once you think about it, they make perfect sense. If we are going to fail in the building of a new product, it is best to find out before a lot of resources have been invested. Problems with design and technology are more easily corrected in the early stages of development. The agile approach supports fast failure by recommending that the team do the hardest or most risky tasks first. This sounds great, but is a very difficult discipline for a team to implement. Human nature drives us to success. We want to perform activities we know we are good at. We want to show progress and get something done. This is true during requirements elicitation, process modeling, software development, testing, and most other types of work. Human nature makes fast failure one of the most challenging agile principles to follow.

Human nature makes fast failure one of the most difficult agile principles to follow.

Fast failure is even more difficult if team members don't know each other well, don't trust each other, or if team members don't trust their management. When a new member joins a team, he wants to show the other team members that he is smart, capable, productive, etc. The last thing he wants to do is try to build a feature he is not confident he can complete. He will choose tasks from the task board that are easy and well understood. If all of the team members behave in this very natural way, the team will complete lots of easy tasks and the project will look very successful. But lurking on the "not yet started" task list will be the most risky work. Eventually someone on the team will have to begin working on these tasks, and problems will begin to be exposed. If one of these problems highlights an architecture problem or technology limitation, much of the work already completed may need to be redone. This makes the late failure even more expensive and time consuming.

Effective teams and their leaders must be diligent about trying to find the fast failures, starting out with the hard and risky tasks first and openly discussing problems during daily standup meetings. Once the hard problems are solved, the easy tasks will move along more quickly and product development will move even faster than expected. Team members must trust that fast failures are not considered personal weaknesses, but rather team successes. The more problems that are discovered early, the more success will come later.

We've always understood the importance of risk assessment and management. Traditional project managers and business analysts develop a list of risks and monitor them throughout the life of the project. But an agile approach incorporates a more proactive way

7. Fast Failure *(continued)*

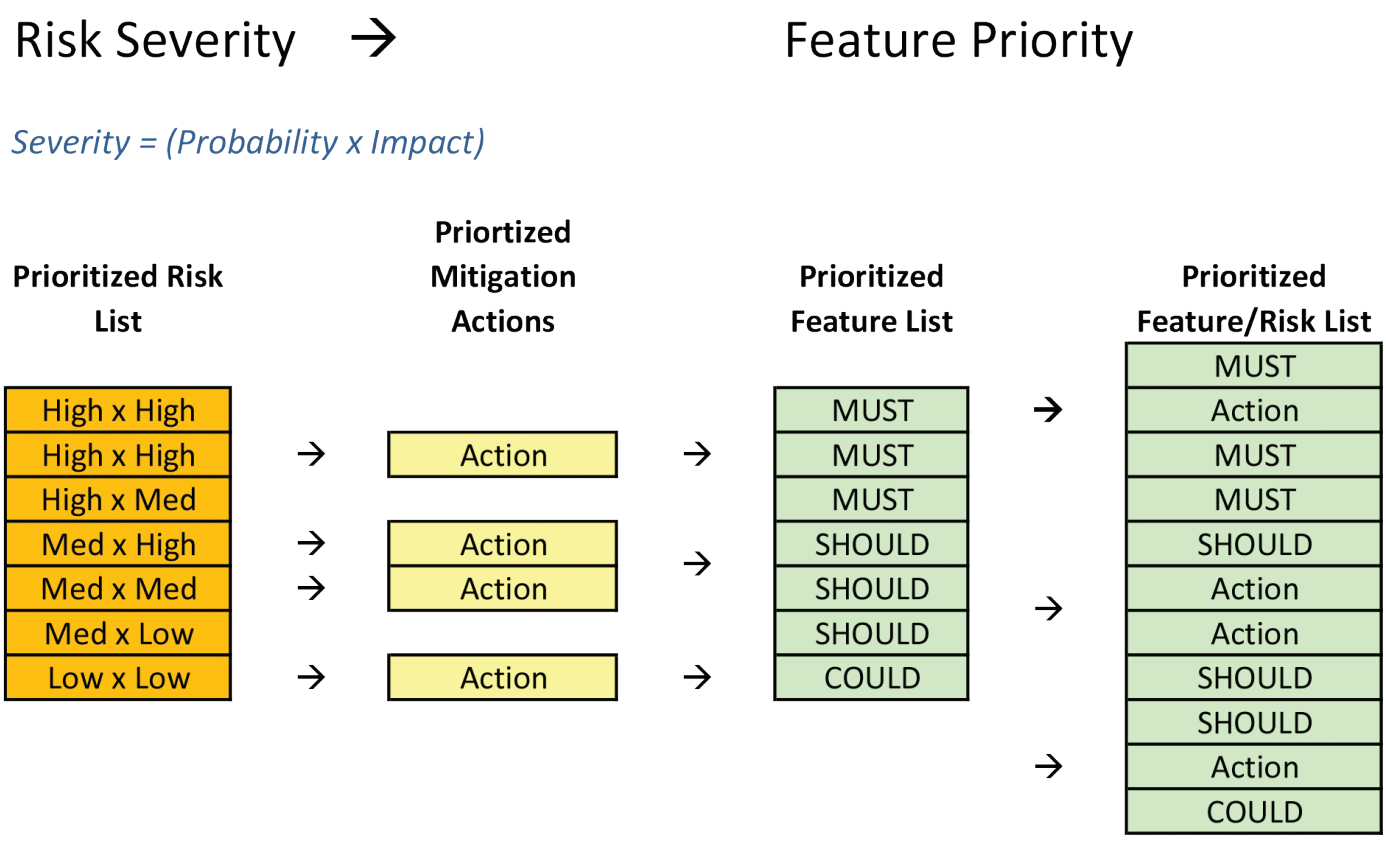
of managing risk. Risks deemed to be high-likelihood and/or high-impact are reframed as features or requirements and prioritized to be addressed early in the project. For example: If there is a risk that a new technology may not support a product feature adequately, trying that new technology becomes a high priority task. These risk spikes are assigned to early sprints so the team can find out as early as possible if the plan or design needs to change. Once a risk has been addressed, it can be removed from the risk list, which allows the product development to move faster. Project managers and business analysts don't have to monitor risks as separate components because they are built into the product backlog and are addressed in the order of importance, just like every other feature.

This *risk adjusted back log* combines stakeholder requests with mitigating actions in the order they will be completed.

To learn more:

1. Highsmith, Jim. *Agile Project Management: Creating Innovative Products* (2nd Edition) Agile Software Development Series. Pearson Education. Kindle Edition, 2009.

2. Pixton, Pollyanna, Nickolaisen, Niel, Little, Todd and McDonald, Kent. *Stand Back and Deliver: Accelerating Business Agility.* Addison-Wesley, Pearson Education, 2009.



8. Barely Sufficient

The word "sufficient" means *enough; being as much as needed*; or *adequate for the purpose*. The word "agile" means *quickness, lightness, ease of movement, nimble.* Agile encourages the concept of *barely sufficient* deliverables. All outputs of the team, including requirements, documentation, coding and testing, should be sufficient, just enough but no more. Teams don't have time to do more work than is absolutely needed. To be nimble, you can't have any extra weight or carry any unnecessary supplies. If you imagine an agile person who can change direction at any time and move into various positions, you can see that he or she would be wearing clothing that is barely sufficient (think about acrobats). When hikers go camping in the mountains, they carefully pack their backpacks with only the necessities so they can easily climb in challenging terrain.

Teams don't have time to do more work than is absolutely needed.

The idea of barely sufficient documentation and requirements is a major change for many project managers and business analysts. Requirements engineers, systems analysts, lawyers writing contracts, and technical writers have been trained to carefully document every detail. Plans and schedules are created using sophisticated project management tools. Product design documents are drafted, refined, edited, reviewed and approved before the development effort is approved. Many of these documents are used to authorize funding. Many are used to purchase product components or consulting services. How can we be successful without these carefully crafted specifications?

Success comes from thinking about what barely sufficient means for each team and for each deliverable. The answer will be different for different products and only an experienced, knowledgeable team can make these judgments. There are many factors to consider in determining the formality of deliverables needed. Is the product in a regulated industry? If so, more formality will be needed. Is the product a mission critical product for the success of the organization? If so, more formal planning and business case analysis is needed. How many customers or sales are expected as a result of the new product? Is it a sure thing, or a pilot? How well do the business stakeholders understand the customer's needs?

Project managers may find it helpful to relate this topic to the concept of gold plating. Project teams are cautioned to produce only the deliverables specified in the product requirements. Adding additional features or functionality is gold plating, and considered a waste of time and resources. In agile, the focus is on avoiding gold plating in project and requirements documentation.

To learn more:

1. http://alistair.cockburn.us/Balancing+lightness+with+sufficiency

2. Highsmith, Jim. *Agile Project Management: Creating Innovative Products* (2nd Edition) Agile Software Development Series. Pearson Education. Kindle Edition, 2009.

3. Cockburn, Alistair. *Agile Software Development: The Cooperative Game* (2nd edition). Addison-Wesley, Pearson Education, 2007.

4. Gottesdiener, Ellen and Gorman, Mary. *Discover To Deliver: Agile Product Planning and Analysis*. EBG Consulting, 2012.

9. Visualize Results

Another key principle of agile is to be very transparent about progress. Rather than a traditional project manager creating weekly or monthly management status reports, agile teams report their progress daily, in a public place, for anyone to see. Since the team's progress is updated by the team and is always available, this approach saves time for the project manager. It also gives business stakeholders and managers a constant view into the progress of the product, eliminating the risk of an unpleasant surprise just before a deadline. This open, transparent strategy fosters better communication and reduces team members' anxiety about what management will do when they find out the project is behind schedule. Unfortunately, visualizing results is a significant and difficult change for many teams.

Since agile team assignments are broken into small tasks, reporting daily progress allows the team to monitor their work, learn their own velocity, and get better at estimating future tasks.

Organizations that have inadvertently developed a culture of blame and retribution will really struggle with this aspect of agile. When team members and team leaders fear truth telling and have learned to hide bad news for as long as possible, they will be very reluctant to try this technique. But, as much of the information in our world becomes more available and uncensored, project progress reporting must also change. Since agile team assignments are broken into small tasks, reporting daily progress allows the team to monitor their work, learn their own velocity, and get better at estimating future tasks. Small tasks can only fall behind in small ways, allowing the team to immediately respond. Visualizing progress gives management constant access and awareness of progress.

This technique is very useful on traditional teams as well. Often, a simple list of today's priorities posted on a wall helps managers and stakeholders stay aware of the team workload. Most teams have more work to do than time to do it, and must constantly make decisions about where to spend their time. If those decisions are posted and made public, managers will be made aware of which tasks are getting done and which are delayed due to resource limitations. If a manager or business stakeholder brings an additional request for work to the team, the team can easily point to the current assignments and ask for an updated prioritization. Example: Would you like me to stop working on a new product feature and instead solve the current system problem? Like business stakeholders, managers don't like to make prioritization decisions, so teams must tactfully present options to their managers. Visualization is a strong technique for doing so.

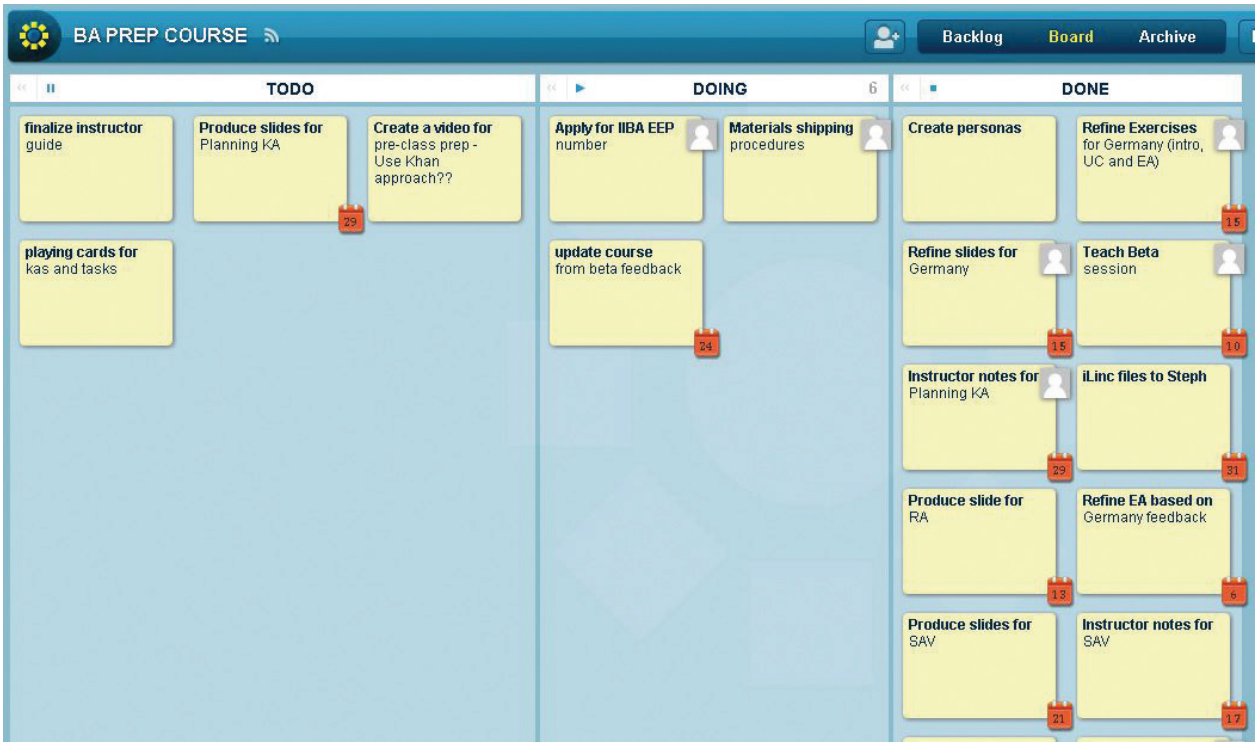
9. Visualize Results *(continued)*

A Kanban board is one visualization tool which is simple to use on any project. It is a great way to introduce an agile practice into a traditional team. Celebrate success as tasks are completed, and use the team's creative knowledge to help when tasks are stalled. On the left side are tasks that have not yet been started, in the middle are the tasks in process, and on the right are completed tasks. Team members update this board during their daily standups.

To try out a virtual Kanban Board visit www.leankit.com

To learn more:

1. Saddington, Peter. *The Agile Pocket Guide: A Quick Start to Making Your Business Agile Using Scrum and Beyond.* John Wiley & Sons, 2013.



10. Sustainable Pace

The last technique is one everyone likes: sustainable pace. Agile recognizes that traditional processes often result in delays and overtime near the end of a project. Team members have been expected to work as many hours as needed to finish on time, often resulting in exhausted team members. In his book, *Death March*, Edward Yourdon refers to this as "heroic efforts." The natural reaction is to rest once the project implementation is over. Workers often feel the company "owes them" for all of the stressful overtime, and work with less intensity at the beginning of the next project. The cycle continues as this sluggish start puts the new project behind schedule, inevitably leading to the panic phase months later, at its end.

Rather than this unproductive pattern of slack followed by overtime, agile practices strive to set a sustainable pace (or consistent velocity) for the team by setting shorter timeframes (two or three week sprints) where tasks are finished and productivity is assessed throughout the project. Rather than pushing out implementation dates, uncompleted work is reprioritized based on new information and scheduled for a future sprint. If a task took longer than expected because it was more complex than planned, the product owner has the opportunity to reconsider its cost-benefit and adjust immediately. Sustainable pace is reinforced through daily standups and the Kanban board or visual progress chart. Team members experience a less chaotic, more consistent work schedule which reduces stress and increases employee satisfaction.

Agile practices strive to set a sustainable pace (or consistent velocity) for the team by setting shorter timeframes (two or three week sprints) where tasks are finished and productivity is assessed throughout the project.

To learn more:

1. Yourdon, Edward. *Death March: The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects*. Prentice Hall PTR, 1999.

2. Tate, Kevin. *Sustainable Software Development.* Addison-Wesley, 2006.

3. Highsmith, Jim. *Agile Project Management: Creating Innovative Products* (2nd Edition) Agile Software Development Series. Pearson Education. Kindle Edition, 2009.

Conclusion

Having productive, well-functioning teams is the key to successful new product development. Agile principles value the team and collaborative work, quick delivery of small, valuable product features, and transparent and open communications. Agile advocates believe that collaborative, stable teams will be the most productive and bring the most business value to an organization.

Most new products are too complex for any one person to design and build, creating the need for a highly skilled team, made up of individuals with expertise in the areas of needs analysis, product design, architecture design, product development and quality testing. These highly skilled team members do not need traditional, directive management, but rather are most effective in a collaborative environment where they learn about customer needs, participate in decisions about their work, and make suggestions about the product features. Successful leaders on these teams act as coaches or mentors, helping the team members to resolve problems and improve their unique team velocity, rather than dictating standard practices.

As projects and organizations increase in complexity, the skills and knowledge of project team members must keep pace, and the ability of team members to work well together becomes critical. Many of the principles and techniques of the agile approaches build on the experience of successful teams and innovative product development. Human beings are naturally collaborative and work best when they are supported in an environment where their opinions are valued and their relationships nurtured. Younger people have less interest in working for directive leaders and more interest in participating in choices about their assignments. They are comfortable working in teams, utilizing new and lightweight technology, and are interested in understanding stakeholder needs so they can contribute ideas to the product design and development. Younger workers are also committed to work-life balance and reject the heavy overtime requirements of traditional projects when they fall behind schedule. All of these preferences are supported by an agile approach.

In addition, the pace of change in organizations points to the advantage of delivering smaller product components or pieces of functionality which bring immediate value, rather than waiting for a large product to be completely developed. Agile planning techniques focus on short-term detail and long-term vision. The product owner is expected to work with other stakeholders to define clear priorities for delivery. Agile teams attack the hard problems early to eliminate risks and ensure success. The smaller increments of value are easier to create and allow for

The pace of change in organizations points to the advantage of delivering smaller product components or pieces of functionality which bring immediate value, rather than waiting for a large product to be completely developed.

Conclusion *(continued)*

adjustments as customer needs change. These smaller products can be delivered in shorter increments of time, allowing the team to celebrate success and reflect on process improvements on a regular basis. To deliver quick business value, agile teams use lightweight documentation and processes, replacing complex methodologies with collaborative, frequent communications.

Finally, transparency about team assignments, progress, and priorities improves communication, prioritization decisions, and stakeholder expectations. When everyone in the organization understands the scheduled timeframes for delivery and the progress of the team, change management is easier because customers and users face fewer surprises. Team leaders are not the only people who communicate project status. They spend their time removing barriers from the team and facilitating team behaviors that increase productivity. People who develop the products along with the people who use the products are more satisfied with their work, contributing to a positive work environment for everyone.

Related Resources

Agile Fundamentals
Instructor-Led Classroom

Agile Fundamentals
Instructor-Led Virtual Course

Agile Fundamentals
e-Learning Course

Connect to RMC's Blog
Converging 360

ABOUT RMC Learning Solutions™

RMC Learning Solutions develops and trains project managers, business analysts and agile practitioners by helping them learn the skills necessary to succeed in their careers. We deliver a wide range of training in multiple learning formats across the globe.
Founded in 1991 by Rita Mulcahy, the company continues to develop and provide innovative, real-world tools and instruction, delivered by professionals with extensive experience and a working knowledge of industry best practices.

For more information visit **shop.rmcls.com**

RMC Learning™ Solutions

10953 Bren Road East • Minnetonka, Minnesota 55343 USA
Main 952.846.4484 • Fax 952.846.4844 • E-mail info@rmcls.com