SEPTEMBER 2018

The Agile Team's Playbook to Doing Agile

A how-to guide for agile practitioners

Agile is an umbrella term for a variety of work-management approaches that share common principles, among them cross-functional collaboration, a focus on customer value, iterative and incremental delivery (early and often), pulling quality forward, limiting work in progress (WiP), and continuous improvement. Once considered a fad, agile has matured into a popular and respected set of development methods. In fact, agile has expanded outside of software development and IT into sectors like banking, management consulting, automotive manufacturing and healthcare.

Purpose-driven companies are moving to agile methods because the global marketplace demands better outcomes and better products that address customers' needs faster. Agile approaches offer faster delivery, higher quality and an engaged development team that can deliver on its commitments.

To enable agile fully, organizations must ensure their agile teams are empowered to deliver value with increased speed and flexibility. And to stay competitive, organizations must learn how to leverage the work of these teams effectively. This requires technology, process and reporting uniformity to facilitate increased velocity, communication, collaboration and value.

This playbook has been designed to help organizations of all sizes—whether they have one agile team or many—harness the power of agile. If your team(s) have been struggling with adopting agile, this guide is for you.

> "The benefits of agile are multidimensional. But the most important change is that it focuses the entire organization on meaningful delivery to the customer. "
>
> -Vice president of infrastructure management, BMC Software

1: What Does an Agile Team Look Like?

2: How to Write a Great User Story

3: How to Prioritize Work

4: How to Use Agile Data to Measure Value

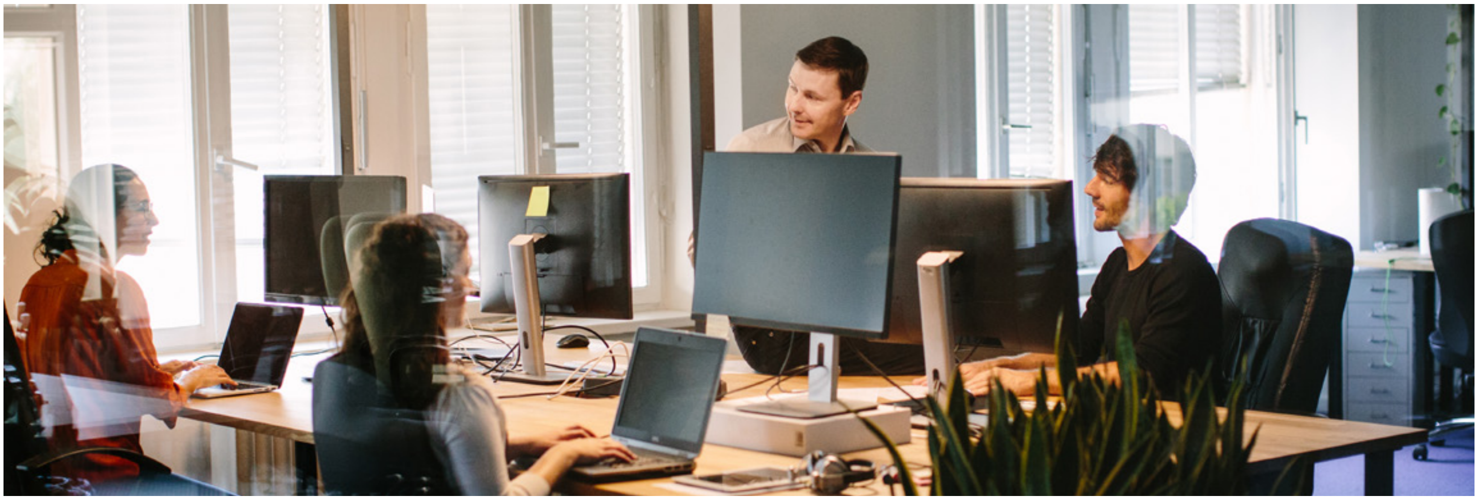
5: What Is Release Planning?

6: What Is Iteration Planning?

7: How to Stay on Track With Daily Standups

1: What Does an Agile Team Look Like?

Agile teams come in many shapes and sizes; what works for some organizations may not work for others. But there are some clear recommendations and best practices for keeping your agile teams manageable and efficient. The magic team size range typically used is between five and nine people. Once a team grows, meetings last longer, and the amount of not-so-relevant information and people increases. There is a danger that individuals will start to question the value of everyone attending the daily stand-ups and planning meetings. Use retrospective meetings to generate ideas for how to proceed with the larger team and to brainstorm options on how to reorganize teams that have become unruly.



What Are the Different Roles in an Agile Team?

The **Scrum Master** is a facilitative servant leader who helps the team members stay true to their practices so they can follow through with their commitments. Responsibilities include:

- Enabling close cooperation across all roles and functions
- Removing blocks and shielding the team from disturbances
- Working with the organization to track progress and re-factor priorities and processes within the organization
- Ensuring that agile processes and ceremonies are leveraged, including daily standups, planning meetings, demo and review, and retrospectives
- Facilitating team meetings and decision-making sessions

The **product owner** drives the product from the business perspective. Responsibilities include:

- Defining the requirements and prioritizing their value
- Determining the release date and content, or features included
- Taking an active role in iteration and release planning meetings
- Ensuring the team is always working on the most valuable, high-priority stories and requirements
- Representing the voice of the customer
- Accepting stories that meet the team's definition of done and defined acceptance criteria

The **team members** are the doers. These are the people who are doing and creating the work. Responsibilities include:

- Representing anyone or any team with a vested interest in the success of the initiative or project
- Taking an active role in iteration and release planning meetings
- Doing and completing the work outlined within the user stories

2: How to Write a Great User Story

A user story represents a small piece of business value that a team can deliver in an iteration. While traditional requirements (like use cases) try to be as detailed as possible, a user story is defined incrementally, in three stages:

- A brief description of the need
- The conversations that happen during backlog grooming and iteration planning to solidify the details
- The tests that confirm the story's satisfactory completion

To write well-formed stories, you need to INVEST in these six criteria:

Independent We want to be able to develop in any sequence.

Negotiable Avoid too much detail; keep things flexible so the team can adjust how much of the story to implement.

Valuable Users or customers get some value from the story.

Estimable The team must be able to use the story for planning.

Small Large stories are harder to estimate and plan. By the time of iteration planning, the story should be designed, coded and tested within the iteration.

Testable Ensure the team understands acceptance criteria and the definition of "done," which includes completion testing procedures.

Who uses user stories?

A lot of people. Developers, testers and technical writers use user stories to be able to know what to implement and when things are done. Product owners use them to track overall progress based on their status. And management tends to track user stories rolled up to features, and ultimately releases and release schedules.

Why use user stories?

- Continuous way to express business value
- Avoid introducing details too early that would prevent design options and inappropriately lock developers into one solution
- Avoid the appearance of false completeness and clarity
- Provide small chunks of work that invite negotiation and movement in the backlog

3: How to Prioritize Work

Agile teams rely on a prioritized backlog to guide their work. The most important work is at the top of the list, available for the team to start as soon as they are ready. Less important work is ranked lower and is elaborated on only when it moves closer to the top of the list. As new work comes in, the product owner adds it to the backlog in the appropriate position so that the team always knows what to work on next.

When to prioritize:

- **Constantly.** Rank the backlog continuously.
- **If strategy or focus changes.** Keep in mind, once you have committed to an iteration plan, you should not change the rank of those committed items unless strategy or focus calls for realignment.
- **When feedback dictates the need for change.** Backlog items and work items should be re-ranked whenever you learn new information, usually in the form of feedback or information gathered from the team, stakeholders or end users.

Techniques for prioritization:

- The product owner must rank the backlog, deciding which story is first, second, third and so on.
- Use a bucket scheme to prioritize backlog items, such as MoSCoW (must have, should have, could have, won't have), to simplify the prioritization efforts and maintain consistent objectivity.
- Try individually ranking each item in the backlog so teams can always start with the most prioritized stories.
- Break down larger stories. If a story contains components that can be prioritized differently, consider breaking it into multiple child stories.

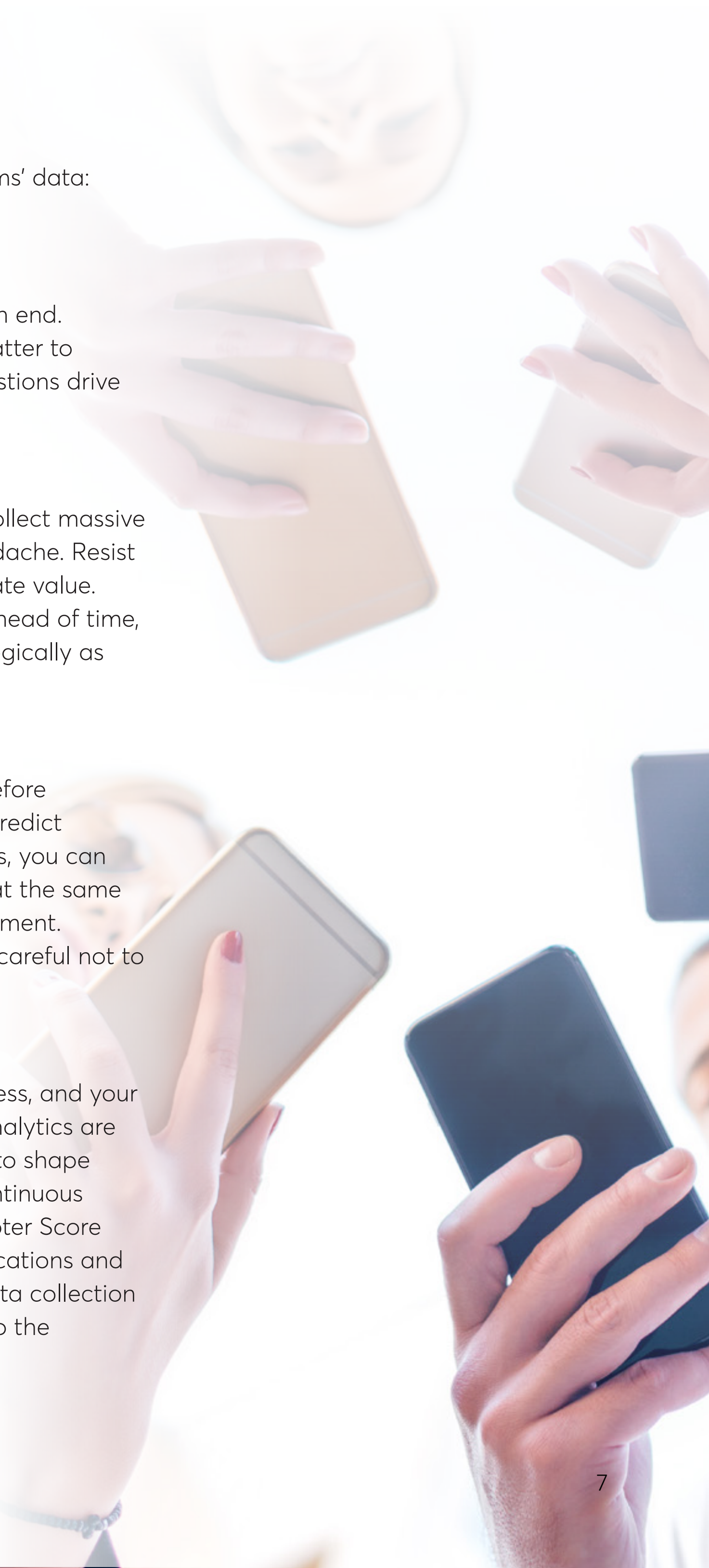
What to consider when prioritizing:

- The value of the feature for the masses
- The value of the feature to a few elite customers
- The cost of the story to build
- The estimated time to implement
- The impact on other stories
- The risk and opportunity costs associated with the story and subsequent feature

4: How to Use Agile Data to Measure Value

In an effort to get agile teams working immediately, many organizations fail to understand a critical component of their new agile organization—data. To ensure continuous improvement of the customer experience, metrics, data and analysis must be considered. By following a few simple steps up front, agile teams can be poised to answer some of the most critical business questions, such as:

- "Why is a feature not being used?"
- "Why are some customers churning?"
- "Why is a minor improvement driving a spike in adoption?"



Follow these four steps to get the value you need out of your agile teams' data:

Step 1: Focus on the questions themselves.

The questions you are working to answer must become your true north. Instrumentation, data collection and analytics are simply a means to an end. Technology is a powerful tool, but it can't tell you the questions that matter to your customers and your business. Make sure the answers to those questions drive specific, meaningful action and, above all, customer value.

Step 2: Measure for insights.

It will be tempting to over-instrument code and infrastructure and to collect massive amounts of data that will end up being just another management headache. Resist the urge and start small, concentrating on a specific area to demonstrate value. Keep in mind that you won't always know what data will be valuable ahead of time, so you will need to be speculative. Don't be afraid to over-collect strategically as long as doing so does not become an operational burden.

Step 3: Run experiments.

Analytics allow you to validate the results of proposed optimizations before adopting them at scale. Use them to guide ongoing experiments and predict changes that will improve the customer experience. Armed with insights, you can then make changes to your service—or provide two different versions at the same time—and measure customer response to inform your product development. Customer insights are the best way to improve customer value; just be careful not to experiment at the expense of your customers' experience.

Step 4: Optimize for the business.

The insights gained from targeted measurement can inform your business, and your business needs should inform the insights you set out to obtain. Your analytics are central to your ability to understand your customers, who, in turn, help to shape the evolution of your products and business. The goal is to create a continuous feedback loop. Integrate customer-facing measures such as Net Promoter Score (NPS) to track your progress and help guide the evolution of your applications and services. Don't get distracted by the data itself. Instrumentation and data collection are important, but only if they help you answer questions that matter to the customer and to the business.

5: What Is Release Planning?

Release planning is a commitment to a plan for delivering an increment of product value. It is a collaborative meeting involving Scrum Masters, product owners, delivery teams and stakeholders.

Who is involved, and what is the goal?

As mentioned above, release planning involves the entire agile team. The goal of the planning meeting is to decide what items or stories from the product backlog will make it into the scheduled release.

Here is who is involved and a description of their typical role during planning session:

- **Scrum Master:** Facilitates the meeting
- **Product owner:** Represents a general view of the product backlog
- **Delivery team or agile team:** Provides insights into technical feasibility and dependencies
- **Stakeholders:** Act as trusted advisors as decisions are made around the release plan

What is needed before the planning meeting?

Before getting started with release planning, you will need the following:

- A ranked product backlog that is actively managed by the product owner
- Input from the team about overall capabilities, known velocity and technical impacts
- The high-level vision, market and business objectives
- An acknowledgment of whether new product backlog items are needed
- Actual supplies to host the meeting: projectors, flip-charts, markers, planning data (described below) and a posted version of the meeting purpose

What do you mean by "planning data"?

Before arriving at your release planning meeting, certain information needs to be compiled, assessed and made accessible.

This data usually includes the following:

- Results of previous iterations and releases
- Feedback from stakeholders on the product, market situation and deadlines
- Action plans and goals from prior releases and retrospectives
- Items and defects to consider
- Development and architecture information
- Velocity from previous iterations or estimated velocity of the team(s)
- Input from other teams and subject matter experts to manage dependencies

What is the desired output of the planning meeting?

It might go without saying, but the output is your plan. The entire purpose of this meeting is to agree on what the agile teams will be working on over the course of the next release cycle. Here are a few things that should come out of the planning meeting:

- A solid release plan and commitment
- Issues, concerns, dependencies and assumptions that will be monitored
- Any new items for the release backlog
- Suggestions to improve future planning meetings

Release planning may be the single most important meeting held within your organization. This is your chance to drive alignment across all stakeholders and ensure that everyone leaves the meeting with a clear vision of how to achieve their goals.

6: What Is Iteration Planning?

The purpose of the iteration planning meeting is for the team to commit to the completion of a set of the highest-ranked product backlog items. These commitments define the iteration backlog and are based on the team's velocity or capacity and the length of the iteration timebox. If you recently went through release planning, you should have a pretty good idea of the work that needs to be done over the release cycle. The iteration meeting aims to assign the highest priority items to an increment of time, or an iteration—also called a sprint.

Who is involved, and what is the goal?

Planning the contents of an iteration has two stages: determining how many user stories can fit into the iteration, and then breaking those stories down into tasks and assigning owners. Iteration planning is a collaborative effort involving these roles:

- **Scrum Master:** Facilitates the meeting
- **Product owner:** Represents the detail of the product backlog items and their acceptance criteria
- **Delivery team:** Defines the tasks and effort necessary to fulfill the commitment

What is needed before the planning meeting?

- The team has sized items in the product backlog and assigned relative story point values
- The product backlog stack-ranked to reflect the priorities of the product owner
- There is a general understanding of the acceptance criteria for the ranked backlog items

How to determine team capacity

The capacity for the team is derived from three measures for each team member:

- Number of ideal hours in the workday
- Days in the iteration that each person will be available
- Percentage of time each person will dedicate to this team

What are the planning steps?

- The product owner describes the backlog item
- The team determines completion tasks for that item
- Team members volunteer to own the tasks
- Task owners estimate the ideal hours they need to finish their task
- Planning continues until the team reaches capacity; if an individual exceeds their capacity during planning, the team collaborates to better distribute the load

7: How to Stay on Track With Daily Standups

The daily standup meeting is for and about the team and its commitments. In this meeting, the team checks in on how the work is progressing in the iteration, adjusts plans and gets assistance with removing impediments.

Who

Delivery team, product owner and Scrum Master. Stakeholders can observe but are not allowed to speak. This is not a status meeting for the product owner.

Timing

Every day, same time, same place, same people. This puts a regular rhythm and cadence on everyone's calendar. The meeting lasts no more than 15 minutes.

Preparation

Individuals collect their achievements and impediments from the previous day. Remember to keep necessary progress charts available for reference.

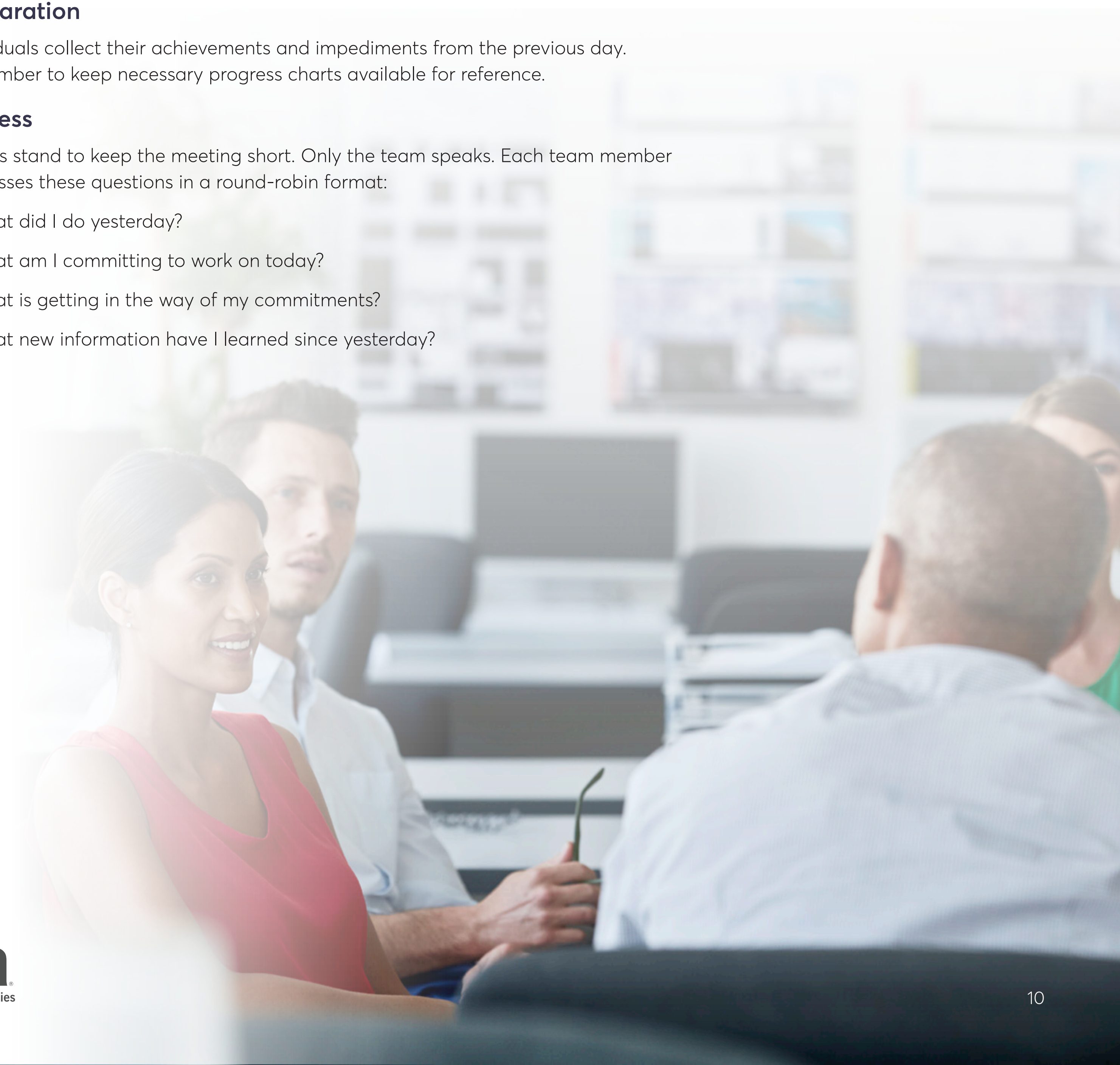
Process

Always stand to keep the meeting short. Only the team speaks. Each team member addresses these questions in a round-robin format:

- What did I do yesterday?
- What am I committing to work on today?
- What is getting in the way of my commitments?
- What new information have I learned since yesterday?

Results

As a result of the daily standup, individuals adjust their daily plans and get help removing impediments. No problem-solving occurs in the daily standup. If problem-solving must occur, the team ends the standup and starts a different meeting for this work to begin.



Summary

Agile is a great tool for any organization seeking to improve quality, time to market, productivity and customer satisfaction. Rest assured, it's not uncommon for organizations to experience uncertainty when changing from longer-term strategic planning to an iterative agile planning model. Often, organizations expect instantaneous results from their newly founded agile teams without allowing these teams the opportunity to learn how to "do" agile. The good news is that by implementing best practices and actively practicing agility, teams will get better at doing and becoming agile. And with the proper vision, context and objectives established up front, your agile teams can and will deliver your desired business outcomes.

If your organization needs additional information on how to become more agile or is looking for agile software that can support multiple agile teams, take the **CA Agile Central Product Tour** now.

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

Connect with CA Technologies