ca technologies

CA Technologies Edition

Agile For Dummies®

Learn to:

- Improve developer productivity
- Put the most critical information at developers' fingertips
- Deliver great results with all your Agile projects

Mario E. Moreira
Michael Lester
Steve Holzner

CA Technologies provides your organization with the most complete and proven project and portfolio management solutions in the industry today. **CA Agile Vision** can help you get better visibility into your deliverables, costs, and resources for all your sprints and projects. CA Agile Vision runs on Force.com, one of the leading cloud management platforms. Delivered 100% over the Web, the solution is available anywhere, fully secure, rapidly scalable, and is supported by world-class facilities and professionals.

Agile FOR DUMMIES®

CA TECHNOLOGIES EDITION

by Mario E. Moreira,
Michael Lester, and
Steve Holzner

WILEY

Wiley Publishing, Inc.

Agile For Dummies®, CA Technologies Edition

Published by
Wiley Publishing, Inc.
111 River Street
Hoboken, NJ 07030-5774
www.wiley.com

Published by Wiley Publishing, Inc., Indianapolis, Indiana

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Business Development Department in the U.S. at 317-572-3205. For details on how to create a custom *For Dummies* book for your business or organization, contact bizdev@wiley.com. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN: 978-0-470-87693-0

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

WILEY

Table of Contents

Publisher's Acknowledgments

We're proud of this book and of the people who worked on it. For details on how to create a custom *For Dummies* book for your business or organization, contact `bizdev@wiley.com`. For information about licensing the *For Dummies* brand for products or services, contact `BrandedRights&Licenses@Wiley.com`.

Some of the people who helped bring this book to market include the following:

Acquisitions, Editorial, and Media Development

Project Editor: Carrie A. Burchfield

Editorial Manager: Rev Mengle

Custom Publishing Project Specialist: Michael Sullivan

Composition Services

Senior Project Coordinator: Kristie Rees

Layout and Graphics: Kelly Kijovsky

Proofreader: Melissa Cossell

Publishing and Editorial for Technology Dummies

Richard Swadley, Vice President and Executive Group Publisher

Andy Cummings, Vice President and Publisher

Mary Bednarek, Executive Director, Acquisitions

Mary C. Corder, Editorial Director

Publishing and Editorial for Consumer Dummies

Diane Graves Steele, Vice President and Publisher, Consumer Dummies

Composition Services

Debbie Stailey, Director of Composition Services

Business Development

Lisa Coleman, Director, New Market and Brand Development

Introduction

Agile is a relatively new strategy that's proving to be very effective in project development of all kinds. Agile development involves an incremental, iterative approach to projects. It stands in distinction to methods that plan extensively from the start, instead letting the project dictate its own parameters as much as possible.

Although Agile emphasizes leaner governance of development teams than organizations may be used to, that hardly means Agile is less disciplined than other development approaches. Instead of relying on discipline coming from above, in Agile, the team self organizes and internalizes its own discipline and commits to the project.

This eliminates the two-step aspect of many project development models, where management directs the team in a horse-and-cart fashion. In Agile development, governance and development aspects become a seamless whole, eliminating the two-step nature of other development strategies. And when a development team becomes a single highly-motivated self-governing unit, development can become much more streamlined and effective.

About This Book

In this book, you're going to get a guided tour of Agile development with particular emphasis on the most popular of Agile methodologies — Scrum development.

While Agile is a development philosophy, Scrum offers concrete plans for your project. It contains many well-tested components that have been tuned over the years, and we bring Agile to life using Scrum practices.

Scrum development involves many roles, such as the ScrumMaster, the Scrum Team, the Product Owner; new concrete concepts, such as sprint (product development iterations) product and sprint backlogs; and much more. You see all those terms in this book.

In addition, you get an inside look at a powerful Agile product, CA Agile Vision. It implements the Scrum methodology and lets teams communicate and plan online. CA Agile Vision provides you with the tools you need to implement Scrum quickly and effectively. Throughout this book, you get information on Agile in general, but then we include the CA Agile Vision methods of making your life easier with this software.

How This Book Is Organized

This book contains seven chapters to help you grasp the concepts of Agile and Scrum and CA Agile Vision.

Chapter 1: Welcome to Agile

This chapter serves as an introduction to Agile theory. We go through what makes the theory of Agile development tick and how it's different from traditional methods.

Chapter 2: Introducing Scrum

This chapter covers Scrum methodology in particular with an emphasis on Scrum for a concrete implementation of Agile. You also get your first inside look at CA Agile Vision, a powerful software tool for implementing Scrum for your organization.

Chapter 3: Understanding the Roles in Scrum

Scrum Teams are small — ideally 7 to 12 people on each team. In this chapter, you see what it takes to create a Scrum Team and make it work, including internalizing the needed discipline. You meet the key players of the Scrum Team and find out how to use CA Agile vision to create your team.

Chapter 4: Meeting Your Sprint Planning Goals

In Chapter 4, you discover how to start breaking up projects into manageable iterations, called *sprints* in Agile development. This is the chapter where you get the info on managing backlogs — lists of requirements the developers must meet, as you transfer items from the product backlog (the overall project requirements) to a sprint backlog. A primer in CA Agile Vision gives you some background on how to move a user story from a backlog to a sprint and how to plan a sprint.

Chapter 5: Keeping Everyone Connected

Scrum development relies on communication between team members and others involved in a project. In this chapter, you see how Scrum helps everyone stay in touch, with special attention to the tools offered by CA Agile Vision for this purpose.

Chapter 6: Ten Ways to Decide if Agile is Right for You

Agile and Scrum aren't for every organization. This chapter offers a quick ten-point list of questions to ask to help decide if Agile is right for you.

Chapter 7: Ten Ways CA Agile Vision Can Help You

CA Technologies specializes in bringing software solutions to make your life easier. CA Agile Vision is one of those products available for anyone to purchase and use. In this chapter, you discover the strengths of the software and how it can help your business.

Icons Used in This Book

In the margins of this book, you see several helpful little icons that call out various nuggets of information. They include the following:

REMEMBER

This icon clues you in to information that you should pay attention to.

TIP

A tip points to some practical information that can help you save time, effort, or money (and maybe even all three).

WARNING!

This icon points out pitfalls to avoid.

TECHNICAL STUFF

Information in this icon can represent statistics or industry analytics and information. You can choose to skip it if you want, but you techies out there just may love it!

Chapter 1

Welcome to Agile

In This Chapter

- Exploring Agile's beginnings
- Understanding the parts of Agile
- Figuring out if Agile is right for you

Agile is a set of exciting new practices and techniques that make product development more cyclical (or *iterative,* as we like to call it in Agile–speak) and incremental and brings you closer to what the customer wants by involving the customer at every step. It relies on *lean governance* (management) as opposed to more traditional techniques that rely on *heavyweight governance.*

Agile is all about empowering the team and getting closer to what the customer wants. In place of rigorous upfront planning and the phase-based process, it offers a dynamic, iterative build-and-test cycle, where change is handled well. One of Agile's hallmark features is that it drives the decision-making process lower in an organization, making that organization more responsive and adaptive.

Agile may give you the basic orientation, but Scrum lets you bring Agile to your organization and make it work (for more information on Scrum, see Chapter 2). While Agile suggests a need for good communication, Scrum sets up rules for stand-up meetings and how they should be conducted. Where Agile discusses the need for iterative and incremental development, Scrum brings those terms into focus with well-defined sprints that represent a specific iteration. Where Agile suggests you track tasks to be performed with lists, Scrum has specific techniques in place for doing exactly that.

Why Agile Was Born

Agile was originally conceived as a set of software development values and principles. It was a reaction against the rigid and “big upfront planning” approach that many software developers and business analysts thought stifled the development process and contact with the customer, emphasizing the plan rather than the product and customer need.

As software development corporations became larger, they often increased the emphasis on governance. Top-down control became a significant part of the process, sometimes making developers spend the majority of their time dealing with such control structures, as every step of the development process became rigid and stratified. This development environment was at odds with the smaller, more adaptive software development companies that brought more customer value to the process by creating products that were closer to what the customer wanted, taking less time to create and costing less to develop.

Agile wasn’t a simple software developer’s revolt. While Agile became popular with software developers because it freed time for creativity and drove the decision-making process lower, the principles also made sense from a business perspective.

By driving development costs and times down dramatically and by creating products that are much closer to what the customer wants than most traditional methods, Agile has proven itself as a business strategy. As opposed to the standard requirements-gathering, plan-driven project development, the customer is continually involved in Agile. That continual involvement has the following impact:

- The developed product is far closer to what the customer wants, even as his requirements change or become more elaborate.
- Your customer is happier!
- The costs and development times may be lower.
- The risk of project deviation from what the customer wants is minimized.

And all of that makes good business sense.

The principles of Agile were in the air for a long time when they were finally put to paper in the Manifesto for Agile Software Development (aka, *Agile Manifesto*) in 2001 by its 17 cosigners at The Lodge at the Snowbird ski resort in the Wasatch Range of mountains in Utah. They valued

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

The Manifesto spells out what have since become the foundation principles of Agile:

- The highest priority is to satisfy the customer through early and continuous delivery of product.
- Changes are welcome, even late in development.
- Deliver working product frequently, typically on the order of weeks.
- Build projects around motivated individuals.
- Emphasize face-to-face conversation.
- Working product is the primary measure of progress.
- Continuous attention to technical excellence is a must.
- Simplicity is a great virtue.
- The best designs emerge from self-organizing teams.

The Manifesto cosigners weren't a group of cowboy programmers bent on maintaining their wild and wooly ways. Much of Agile has to do with internalizing the control patterns that were formerly externally governed and internalizing discipline in the team itself. Agile seeks to develop self-motivated teams whose priority is the project.

REMEMBER

When it works, this synthesis of control is invaluable. Rather than having to enforce control from plans, schedules, and rules, the team itself takes over and makes external governance largely obsolete. That, in a nutshell, is the inspiration and the power of Agile.

Having the Agile Mindset

What changes in thinking do you need to make to begin thinking more Agile? Hopefully, with the information in this section, you can figure that out.

Mario E. Moreira, coauthor of this book, wrote *Adapting Configuration Management for Agile Teams.* The Agile Mindset is part of the concept from that book.

Think self-empowered

In Agile, decisions are pushed down to the lowest effective level, and that empowers self-motivated team members. Think in terms of self-empowered and self-motivated teams, not in terms of external command structures holding a whip over reluctant developers.

Think small

Agile relies on a divide-and-conquer strategy. You should be able to break any task up into manageable chunks, each of which is more or less modular in scope. You can refactor problems into smaller ones and make global changes in a single, well-defined place.

Think business value

The highest priority of the Agile process is all about providing value to the customer. Agile encourages team members to internalize the discipline of providing the greatest business value. Team members think in terms of delivery of value during each of the iteration cycles.

Think continuous

In the real world, change happens. When you're working on a project, the customer may change the requirements at any time. The iterative approach allows for change to be introduced at the planning session for the next iteration.

Although the Agile Manifesto says that Agile developers welcome change, that may be a little too much to hope for in an actual development environment. Agile is far better equipped to handle change, especially continuous change, than traditional development methods.

Because the customer is always involved, changes are likely to be of smaller scope than they are in a traditional development process. Fifty small changes can often be handled easier than one massive change. In addition, Agile thinks in terms of short iterations rather than monolithic (single-block) projects, making change easier to handle.

So while it may be too much to say that Agile teams welcome many substantial changes during the course of a project's development, they're better equipped to handle them.

Think collaboration

Agile puts a huge emphasis on collaboration, particularly face to face. This importance is a marked difference from traditional product development where communication often has to go through a third party. Instead, Agile collaboration encourages working together across standard roles to build solutions.

Think discipline

The main reason many organizations are reluctant to adopt Agile is the mistaken belief that it involves *ad hoc* or *cowboy development.* Organizations with heavy governance believe that loosely unsupervised teams result in chaos, not a finished product.

In Agile development, teams are empowered and internalize the discipline needed to complete each cycle. So while many organizations fear the seemingly unsupervised nature of Agile development, the fact is that when it works as designed, it's far more effective and efficient.

Will Agile Work in Your Environment?

Is Agile right for you? That's a question that takes some consideration. Start by taking a look at the methodologies commonly in use today:

- **Waterfall:** A phase-based methodology where you need to complete a phase before moving to the next
- **Hybrid Waterfall:** A phase-based methodology that provides for phase overlap so you can begin the next phase before completing the current phase
- **Incremental:** A methodology that provides for customer delivery in short increments — that is, shorter release cycles
- **Iterative:** A methodology that works with multiple short cycles of progress where at the end of the cycle customer validation is needed
- **Agile:** A collection of methodologies that's derived from both Iterative and Incremental approaches to development to provide customer value

With waterfall development, you have to move from phase to phase (that is, waterfall to waterfall) in accordance with a specified plan. The customer is involved only during requirements gathering at the beginning and user acceptance testing at the end.

Agile, on the other hand, has no set phases, but proceeds iteratively instead with short cycles. Requirements are planned for, implemented, tested, and evaluated repeatedly in a matter of weeks, not months. The customer is always involved.

So how do you decide if Agile is right for you? Agile techniques, hastily and incompletely applied, can result in significant, time-consuming problems. So put on your thinking cap and check out the breakdown of parts of Agile in this section.

Small teams

Agile flourishes with small, face-to-face teams of 12 or fewer members. If you can't break your task up into pieces that can be handled by such small teams, Agile may not be right for you.

Collocation

Collocation means that things are placed together, and in the sense of Agile, we use this term to indicate a preference that everyone is in the same location. While technically not an absolute requirement, this togetherness goes a long way in making the Agile — and in particular, Scrum — process work.

If your teams are highly distributed, Agile may not work as well as it should for your needs, or your teams will need to make an extra effort to communicate. However, some Agile practices can help make this more feasible.

Motivated, seasoned developers

Agile development does require motivated, seasoned developers who can supervise themselves. Seasoned developers tend to be self-guiding, while novice developers may require a great deal of supervision. Agile methods are dependant on developers internalizing their own discipline.

If you only have novice developers, Agile methods may not work for you.

Lean governance

Agile teams do best when they're left alone to do their own thing as much as possible. A properly constructed Agile team is self-disciplined and needs relatively little external governance.

Some organizations can't get away from a heavy governance model — if that describes your organization, Agile may not be for you.

Customer involvement

Traditionally, the customer submits a set of requirements at the beginning of the project and is next contacted when the near-finished product goes to validation. That's not the way Agile works. Here, the customers are involved along the way and particularly in the End-of-Sprint Review (or demo) where continuous feedback is collected to ensure that you're delivering something the customer actually wants. If the customer can't be present, the role of the Product Owner acts as the voice of the customer. (See Chapter 2 for information on the role of the Product Owner.)

If that's too much customer involvement for you, Agile may not be for you.

Chapter 2

Introducing Scrum

In This Chapter

- Discovering Scrum as a part of Agile
- Breaking down the parts of Scrum
- Taking a look at CA Agile Vision

Scrum is a particular variant of Agile (covered in Chapter 1) and the most popular variant today. While Agile is more a set of principles, Scrum puts principles into effect with well-designed practices and techniques. Roles and concepts that are theoretical in Agile become concrete when it comes to Scrum. In other words, Scrum lets you make your project actually happen.

So Scrum is a team-based iterative and incremental Agile methodology for tackling projects. It has its own components, such as the Scrum Team, backlogs, and sprints that we introduce to you in this chapter.

Agile itself is more a set of guiding principles than a strategy ready to be put into practice. In time, Agile evolved into many specific methodologies, such as XP (Extreme Programming), Scrum, Kanban, DSDM, AUP, and more. In this chapter, you focus on the methodology that's become by far the most popular Agile approach — Scrum.

Welcome to Scrum (Yes, as in Rugby)

Welcome to Scrum, Agile's best-known methodology. Before we go any further, we want to let you know a little bit about the name, which comes from the game of rugby. Why rugby?

Looking at Scrum as a methodology

Scrum has a brief history as a methodology. Take a look:

- **1993:** Jeff Sutherland creates Scrum at Easel Corp.
- **1996:** Jeff Sutherland and Ken Schwaber introduce Scrum at OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications) conference.
- **2001:** Ken Schwaber and Mike Beedle write the foundation paper "Agile Software Development with Scrum."
- **2002:** Ken Schwaber and Mike Cohn co-founded Scrum Alliance in 2002, initially within the Agile Alliance.

In rugby, as opposed to American football, the ball is routinely thrown from team member to team member as the whole team progresses up the field. Everyone shares in the task of running, passing, kicking, tackling, defending, and scoring goals.

Scrum is a mass meeting of the two opposing teams where the ball gets put into play. Both teams face each other and put their arms around their teammates' waists; the ball is then rolled between the two opposing teams, which try to control the ball by kicking or throwing it to teammates.

In other words, a scrum is composed of total confusion, culminating in coherence as a team gets control of the ball and then starts to work as a coordinated whole. That's the rugby term, scrum.

So what's Scrum for Agile?

For Agile, Scrum is an iterative, incremental process for developing any product or managing any work. Scrum produces a potentially shippable product at the end of every iteration. In Scrum, you break a task into a series of iterations, called *sprints,* which get you ever closer to the goal. And like rugby, the team acts in a self-empowered and coordinated manner to ensure investment of a common goal and commitment to that goal.

Who are the people in your Scrum?

Scrum revolves around Scrum Teams. Scrum Teams make it all happen; they're at once the source of energy and commitment in Scrum. The Scrum Team is expected to have daily stand-ups where team members discuss what they've done in the previous day, what they expect to do next, and any roadblocks.

The Scrum Team consists of three components: the ScrumMaster, the Scrum Team itself, and the Product Owner (the customer's representative). Each is covered in greater detail in Chapter 3.

The ScrumMaster

The ScrumMaster is the point person for the Scrum Team. They're responsible for implementing Scrum methods, values, and practices. The ScrumMaster is the person who keeps the project on track and moving forward.

The Scrum Team

The Scrum Team consists of the actual task implementers. A team typically consists of 7 to 12 people, and all members typically share the same location.

The Product Owner

The Product Owner is the voice of the customer in the Scrum Team. The Product Owner is typically a product manager or business analyst in the IT firm, and he knows what the customer wants the product under development to do.

Seeing the Greater Scrum Picture

For most of you, you want to know the nitty-gritty details of what's involved in Scrum. This section gives you what's involved.

Telling it like it is with user stories

The customer tells the Scrum Team what's needed through *requirements.* In Scrum, requirements are called *stories.*

A story is a high-level definition of a requirement, containing just enough information so the developers can produce a reasonable estimate of the effort to implement it.

For example, a story may be "Clients need to be able to buy a daily subway pass from an automat."

During Sprint Planning, the Scrum Team takes the supplied stories and breaks them down into specific tasks, each of which is given a time estimate, either in story points (recommended) or in days/hours. These tasks are then kept track of in the sprint *backlog*.

All about backlogs

Backlogs are lists of stories and tasks that enable a Scrum Team to keep track of what needs to be done. The two primary backlogs are the product backlog and the sprint backlog.

- The *product backlog* represents the larger picture and lists epics and stories for a product.
- A *sprint backlog* is concerned with the stories and tasks that are to be undertaken within a sprint. The items in sprint backlogs come from the product backlog during sprint planning.

In sprint planning, the sprint backlog is populated with items. The backlog forms the backbone of the sprint — all the tasks the sprint is to accomplish are recorded there. An example sprint backlog appears in Table 2-1.

Table 2-1 A Sprint Backlog

Backlog Item	*Estimate*
As a customer, I want to be able to insert coins in the vending machine.	12 points
As a customer, I want to be able to select my candy bar.	4 points
As a customer, I want to get my money back if I have not made a selection.	10 points

Over and over with sprints

A *sprint* is a development iteration. In Scrum, you execute projects in successive iterations known as sprints. The sprint is essential to a Scrum Team, because sprints provide the project focus. In a sprint planning session, items are taken from the product backlog and moved to the sprint backlog, based on the priority set by the Product Owner, and given an estimate. The goal of the sprint is to accomplish the items in the sprint backlog.

Sprints are intended to be 2-to-4-week work iterations. At the end of the sprint, a potentially shippable product is delivered to the Product Owner for review.

For the Product Owner's part, no changes should be made to the requirements during a sprint (changes can be introduced at the next sprint planning session).

Sprints typically start with a planning session during which the sprint backlog is created from a prioritized product backlog, and end with an end-of-sprint review, as well as delivering a product to the product owner.

Staying coordinated with daily stand-ups

A primary feature of Scrum development is Daily Stand-ups, called the *daily scrum.* This is a mandatory meeting of the Scrum Team that takes about 15 minutes at the start of the day.

The main attendees are the ScrumMaster and the Scrum Team, although the Product Owner may also attend. The meeting is usually held with all people standing up to emphasize the brevity of the meeting, which gives the meeting another commonly used name — the Daily Stand-up.

At the meeting, each participant is expected to answer three questions:

- What did you do since the last Scrum?
- What will you do until the next Scrum?
- Do you have any roadblocks preventing you from doing your work?

The first two questions are about progress, and the last question is about removing any obstacles to progress.

Inside Look at CA Agile Vision: Introducing CA Agile Vision

So while Agile is the guiding light (see Chapter 1), Scrum is what brings the discipline that makes it work. And that's where CA Agile Vision comes in. CA Agile Vision is a powerful online solution for implementing Scrum for your organization, putting the planning and tracking parts in place for you. CA Agile Vision takes Scrum out of the textbook and lets you put it to work immediately, planning sprints, tracking progress, and creating iterations in the real world. CA Agile Vision provides a complete Scrum implementation, bringing Agile home to you.

Coordination can be critical when it comes to distributed teams, especially those in significantly different geographic locations. By hosting all planning and scheduling documents online, everyone has equal access to a single copy of such documents — the duplication problems that come with multiple locations are avoided. By using an online Scrum system, such as CA Agile Vision, you automatically coordinate these resources for Scrum development:

- Backlog management tools
- Sprint management tools
- Sprint reporting tools
- Time tracking tools

Chapter 3

Understanding the Roles in Scrum

In This Chapter

- Getting a commitment
- Identifying the key players
- Deciding if single or multiple scrum teams work for you
- Using CA Agile Vision to create your team

The focus of any Scrum project is the Scrum team. It's the Scrum team that makes the project happen — the team is the engine behind what needs to be done.

The conditions for building a Scrum team are rarely ideal, and that's where the structure of Scrum development comes in. The whole structure of Scrum development helps pull team members together and motivate them. Team members come to know that they're not operating in a vacuum; they're responsible to the entire team and the customer.

This chapter lets you become familiar with the Scrum team and its delegates. You explore small teams to large teams and how to handle the challenges and victories of each.

The Need for Commitment

When it comes to building Scrum teams, the need for commitment is vital. Scrum teams take a lot of commitment on the part of team members. The team needs to internalize the discipline and drive needed to see the project through to completion. If team members can't commit, the Scrum process is in danger.

Are you the ham or the chicken?

There are two types of people involved with Scrum teams: Those that are committed team members and those that are merely involved (called on for occasional expertise).

There's a joke told by Scrum people to illustrate the difference between being committed vs. being involved: A pig and a chicken were walking down the road one day, and the chicken said, "Hey, let's start a restaurant."

"Sure," said the pig. "What shall we call it?

"How about 'Ham and Eggs'?"

"Don't think so," said the pig. "I'd be committed, but you'd only be involved."

REMEMBER

The ability to commit is a direct result of how self-empowered a team feels.

This, of course, assumes that the team members have something to commit to. Although commitment can be encouraged by the ScrumMaster (see "Getting to Know the Key Personnel" for details), the actual commitments made by each team member are up to that member. The ability to commit and own the project along with the rest of the team is part of what makes a good team member (people who need extensive guidance and watch the clock may not be good candidates for Scrum teams).

Welcome to the Scrum Team

The Scrum team is the engine behind project development. When a team operates well, its efficiency and drive are unsurpassed. When a team doesn't operate well, it can present a significant challenge, which can include replacing team members.

REMEMBER

When you develop any team, remember that teams should have several general qualities. These are ideal requirements — of course any organization has to work with what it has. It's one of the ScrumMaster's responsibilities to teach Scrum methods to those who aren't up to speed yet. For more information on the ScrumMaster see "Getting to Know the Key Personnel" later in this chapter.

Team attributes include being

- Experienced
- Motivated
- Committed (full-time)
- Competent
- Proud of their work
- Able to work well with others
- Responsible
- Willing to work in teams
- Autonomous

Being autonomous is one of the biggest parts of being on the team. Part of creating a well-functioning team involves giving members the autonomy needed for the project. Scrum pushes the level of decision making downward, and it's necessary that team members feel they have the autonomy to become truly invested in the project.

Autonomy stands in contradiction to the environment that prevails in many heavy-governance organizations. Autonomy is often the one thing that developers lack and what prevents them from fully investing themselves in a project. In some organizations, programmers must get approval for every modification to the code, no matter how small, from a manager. This kind of control can often have a stultifying effect on a team.

A good measure of real autonomy is necessary for a Scrum team to work. To some extent, team members have to believe they "own" the project before they can take pride in it, commit themselves to it fully, and internalize the goals of the project. If you can't grant your Scrum team sufficient autonomy to make the process work, you will face challenges in transitioning to Scrum. Taking responsibility for a project means owning it, and that's the way Scrum teams work.

Getting to Know the Key Personnel

Scrum revolves around Scrum teams. Scrum teams make it all happen; they're at once the source of energy and commitment in Scrum. The Scrum team is expected to have daily, face-to-face meetings where team members discuss what they've done in the previous day and what they expect to do next.

The Scrum team consists of three components: the ScrumMaster, the Scrum team itself, and the Product Owner (the customer's representative).

The Team Leader: The ScrumMaster

The ScrumMaster is the team leader of the Scrum team. It's up to the ScrumMaster to make sure that the team meets the sprint goals defined by the Product Owner. The ScrumMaster makes the project go and keeps it on track. Here are some of the attributes a ScrumMaster needs:

- Responsible for implementing Scrum methods, values, and practices
- Responsible for teaching Scrum to the Product Owner and Team
- Acts as leader of the team (aka, Project Manager or Team Leader)
- Facilitates daily Scrum meeting in which team members report to each other face-to-face
- Removes roadblocks and obstacles from the team's path
- Deals with team member problems/issues
- Be ultimately responsible for the success of the project

Ideally, experienced teams come together and the chemistry works to get the project off the ground. But the ScrumMaster is the team leader, and it's the ScrumMaster's job to make sure all team members are motivated. As you may expect, working with experienced ScrumMasters is easier than working with first-time ones, just as it's easier to work with experienced team members than novices.

In organizations new to Scrum, the ScrumMaster may spend a significant amount of time teaching Scrum methods, goals, and values to the team (and the Product Owner).

The customer's representative: The Product Owner

In Agile and Scrum, the customer is always present through a representative called the *Product Owner.* The Product Owner is responsible for providing the Scrum team with the requirements for the project.

The Product Owner should be readily available, ideally collocated with the Scrum team. Typically the Product Owner

- Is a Product Manager himself and talks in terms of requirements and gathers requirements effectively
- Knows all the requirements of the project — shouldn't have to refer to third parties continually
- Is the voice of the Customer
- Knows what the customer wants the Product to do and is able to supply that information immediately
- Always is accessible to the team
- Manages Product Backlog
- Gathers requirements/stories from customers
- Provides acceptance criteria for requirements/stories
- Prioritizes the requirements/stories (Backlog Grooming)
- Decides what will actually be delivered
- Validates the functionality/product delivered in reviews (End-of-Sprint Reviews)

WARNING!

The Product Owner can be a challenging role, because it involves being committed to both the Scrum team and the customer. It's also a tough role, because it's the Product Owner's responsibility to get the project requirements right; if those requirements are wrong, and the team goes in the wrong direction as a consequence, accountability lies with the Product Owner.

It's important that the team can treat the Product Owner as the final source of project requirements — the go-to person when there's a question. Too often, lack of knowledge is a bottleneck in standard projects, because the customer representative is unsure of the answer and needs to refer to someone else, which usually involves phone messages, conferences, and inevitably, delays.

Although not mandatory, the Product Owner can also attend the daily Scrums. This can greatly help maintain effective communication between the customer and the Scrum team.

The team itself

The people who make up the team are the driving force on the Scrum team. They should be seasoned self-driven professionals and be able to work well with others but also given sufficient autonomy to take responsibility for the project and own it.

The team needs to embody all that's needed to successfully complete the project. A team's attributes usually involve the following:

- Typically about 7 to 12 people
- Cross-functional to all the necessary tasks to be handled (for example, including programmers, architect, designers, QA/testers, tech writer, and CM/build engineer)
- Full-time (committed)
- Self-organizing and self directed
- Autonomous
- Motivated
- Owns the project
- Ideally collocated (while distributed teams are possible, they can complicate the process considerably)

The team takes items from the product backlog and moves them to sprint backlogs based on the priority set by the Product Owner. They estimate how much time each item in a sprint will take. They track progress in burndown charts. They get the work done and take ownership for the project.

Working Together or Apart?

Scrum is intended to be very face-to-face. Team members are supposed to be in constant communication with each other, and with the ScrumMaster. Continual work-related conversations are expected to take place in a Scrum team. But that situation may not always be possible.

Communication is crucial to any Scrum team — both internal and external communication. For more information on communication within the team, check out Chapter 5.

Collocated teams

Ideally, you want a collocated Scrum team. Face-to-face communication is best when it comes to Scrum. There should be no barrier to communication between team members and that means physical as well as professional barriers.

If a developer has a design question, he should be able to get an immediate answer from a designer, for example, without the delays introduced by phone tag, unreturned e-mails, and the like.

For this reason, Scrum experts often say that a Scrum team should inhabit not only the same building, but the same room inside that building. If someone needs an answer, they should be free to walk over to someone's location or cubicle and get that answer quickly.

Collocation also fosters a team spirit. For example, if you have a few members who are working late together, maybe they can pitch in on a pizza together. Taking a break from work and having nonwork related interaction is important too. The idea is to put all team members in a face-to-face mix for maximum productivity, keeping all team members engaged and in communication.

Dealing with a distributed Scrum team

Sometimes, collocating an entire Scrum team isn't possible (see the preceding section on collocating). Although not necessarily ideal, it's possible to work with distributed Scrum teams.

A *distributed team* occurs when within a large organization, team members may be anywhere around the world, and although that represents some challenges to the Scrum process, there are ways of dealing with such challenges as much as possible.

When you have a distributed team, you face several significant challenges:

- **Time zone differences:** Developers who are in time zones 11.5 hours apart (California and Hyderabad, for example) can't quickly and easily share thoughts and information or give instant feedback.
- **Language and regional differences:** Differences in language and regional holidays, transportation, and communication infrastructure also place limits on when and how meetings can be conducted.
- **Multiple copies of documents:** Scrum development depends on recording tasks and progress in product and sprint backlogs, burndown charts, task lists, and so on. When you maintain multiple locations for a team, multiple copies of those documents can easily become uncoordinated.
- **The impossibility of daily stand-up meetings:** If developers are located in significantly distributed geographic locations, the daily Scrum may not be possible.

For more detailed information about how to handle some of these challenges, flip to Chapter 5.

The backbone of working with a distributed Scrum team is document coordination. Documents set goals and track progress for Scrum teams, and making sure everyone has a current copy of all schedules and charts is vital to working in a distributed way.

Working With Multiple Scrum Teams: Scrum of Scrums

It can happen that, on larger projects, one Scrum team isn't enough. Say you're designing a whole software operating system, for example, with millions of lines of code. Such a project would take a 7-person Scrum team a long, long time to complete.

So you create a 100-person Scrum team . . . wait, wait, wait — that's contrary to the Scrum philosophy, which insists that Scrum teams stay small. So the solution?

Larger projects should be divided up between multiple Scrum teams to get the work done. To keep larger projects going smoothly, Scrum experts recommend the formation of three additional scrums, or "scrum of scrums:"

- The Project Progress Scrum of Scrums
- The Architecture Scrum of Scrums
- The Product Owner Scrum of Scrums

Each of these is a "scrum of scrums" in that they can be composed of team members from each of the Scrum teams working on the project. Each Scrum of Scrums serves a different purpose.

Introducing the Project Progress Scrum of Scrums

The primary Scrum of Scrums typically focus on project progress and dependencies across Scrum teams. This group makes sure that scrum teams are selecting the high priority stories for sprints and breaking them down effectively. It keeps the overall project on track — and that's especially important in larger projects.

This Scrum of Scrums is typically comprised of the ScrumMasters, Development and QA leads, and the Product Owners who have overall release responsibility. The Project Progress Scrum of Scrums is responsible for the following:

- Monitoring the progress of each Scrum team and the overall release
- Monitoring the velocity for each Scrum team and the overall project
- Identifying and working any blocking issues that are beyond the scope of an individual Scrum team

The Project Progress Scrum of Scrums is in charge of just that — overall project progress. They're supposed to find blockages and fix them across Scrum teams before they become problems. This Scrum of Scrums can also serve as a "steering committee" for the entire project, keeping the project on track.

Ideally, Project Progress Scrum of Scrums meet daily for 15 minutes with a representative from each Scrum team.

Introducing the Architecture Scrum of Scrums

The Architecture Scrum of Scrums has to do with maintaining the standards of a common architecture across the project.

This Scrum is typically comprised of technical leaders and lead architects from the overall project. The Architecture Scrum of Scrums' responsibilities include

- Decisions across Scrum teams that affect the overall direction of the architecture of the overall product
- Setting common strategies or standards by which all the Scrum teams operate
- Identifying areas in which to avoid duplicate efforts by Scrum teams
- Being a focus for general, cross-team communication

Architecture Scrums include these best practices:

- For a project with a Sprint duration of 2 to 4 weeks, meet weekly for one hour.
- Publish a meeting agenda 2 days before the meeting with the topics to be covered at that meeting.
- Collect issues and assign to smaller working teams.

Introducing the Product Owner Scrum

The Product Owner Scrum of Scrums (also called a Requirements Scrum) is a Scrum intended to keep the overall requirements for the project coordinated between Scrum teams.

This Scrum is typically comprised of the Product Owners from each of the Scrum teams in the project. Its responsibilities include

- Communicating stories that need to be added to the Product or Release backlog of various Scrum teams
- Explaining changes to the Product backlog as Scrum teams have been operating to the members of those Scrum teams
- Identifying overlapping dependencies of stories between Scrum teams and working to alleviate that overlap
- Moving stories from one Scrum team's Product backlog to another team's Product backlog as needed

The Product Owner Scrum of Scrums is all about making sure that the requirements the various Scrum teams are working on stay coordinated and mirror what the customer wants.

Ideally, the Product Owner Scrum of Scrums should meet for 2 to 4 hours the week before the Sprint planning sessions for all the Scrum teams with the goal that each team's Backlog is prioritized and ready for planning, while avoiding needless overlap.

Inside Look at CA Agile Vision: Creating a Team

Creating new Scrum teams and adding new members is easy in CA Agile Vision. With collocated or distributed teams, CA Agile Vision makes it simple to coordinate and communicate between team members.

To create a new scrum team, follow these steps:

1. **Click the Navigator menu and select Teams from the Resource Management menu.**

   The Teams page appears.

2. **Click New Team.**

   The New Scrum Team page appears.

3. **Complete the following fields:**

   - Scrum Team Name
   - Active (specifies whether the team is active)
   - Expected Velocity (defines the estimated total story points that a scrum team believes they can realistically complete during a sprint)
   - Story Point Scale (defines the story point scale your team uses. Enter a comma-separated list of numbers; the default is a Fibonacci sequence 1 through 21).
   - Project (the unique name of the project)
   - Scrum meeting time and location (specifies the time and location for daily scrum meetings)
   - Hours per day (defines the base number of hours per day that all team members spend actively working for the team)
   - Scrum team domain (specifies the URL for the team)

4. **Click Save.**

   The Sprint Assignment page appears.

5. **Click Skip This Step, which lets you assign the team to sprints later.**

   The Scrum Team Detail page appears. From this page you can add members to the team.

6. **Move to the Scrum Team Members section and click New Scrum Team Member.**

   The Scrum Team Member Edit page appears.

7. **Edit the following fields:**

- Member Name
- Role (specifies the member's role on the team)

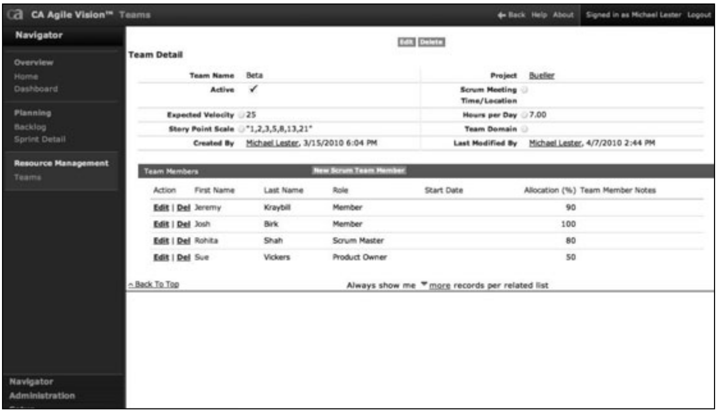
  Possible values include Member, ScrumMaster, and Product Owner.
- Start Date (specifies the date that the member starts on the team)
- Team Member Notes (specifies additional relevant information about the team member)
- Scrum Team (specifies the name of the scrum team to which the member is being added)

  The default is the current scrum team.
- Active (specifies whether the team member is an active part of the team)
- Allocation (%) (specifies the percentage of time the member is allocated to this team or project)
- End Date (specifies the date when team participation for the member ends)

8. **To add the person to the team, click Save.**

   Repeat the process of additional team members.

Check out Figure 3-1 for a successfully created team in CA Agile Vision. After setting up team members, editing or deleting them is easy — just click the matching link next to a name.



Team Detail

Team Name	Beta	Project	Buefler
Active	✓	Scrum Meeting Time/Location	
Expected Velocity	25	Hours per Day	7.00
Story Point Scale	"1,2,3,5,8,13,21"	Team Domain	
Created By	Michael Lester, 3/15/2010 6:04 PM	Last Modified By	Michael Lester, 4/7/2010 2:44 PM

Team Members

Action	First Name	Last Name	Role	Start Date	Allocation (%)	Team Member Notes
Edit | Del	Jeremy	Kraybill	Member		90	
Edit | Del	Josh	Birk	Member		100	
Edit | Del	Rohita	Shah	Scrum Master		80	
Edit | Del	Sue	Vickers	Product Owner		50	

Always show me more records per related list

Figure 3-1: A successfully created team.

Chapter 4

Meeting Your Sprint Planning Goals

In This Chapter

- Discovering the ins and outs of Sprint Planning
- Dividing Sprint Planning into segments
- Reviewing the overall process
- Using CA Agile Vision in real examples

Sprint Planning is crucially important to Scrum. Scrum attacks a project in a series of iterations called sprints, and it's during the sprints that the work gets done. That's why it's important to make sure that a sprint is planned correctly — that's where you set your sprint goals and select your stories. At the end of the sprint, the team is supposed to have a potentially deliverable product to present to the customer.

That's how projects get done in Scrum — you start Sprint Planning by peeling tasks from the Product backlog and move them to a Sprint backlog — then the team performs those tasks, followed by demonstrating the deliverables to the Customer and Product Owner during the End-of-Sprint Review and then this cycle starts again from sprint to sprint. In this iterative way, the product evolves.

This chapter is all about planning a sprint.

Setting Your Sights: Sprint Planning

Sprints are where the work gets done in Scrum development, and before you start a sprint, you spend a day planning that sprint. That's a rule in Scrum — each sprint starts with a Sprint Planning session.

The idea behind Sprint Planning is to determine exactly which stories the team will work on during that sprint. Sprint Planning accomplishes these things:

- It makes sure you know what you're focusing on in a sprint.
- Combined with the End-of-Sprint Review, Sprint Planning ensures that what the team delivers is more closely aligned with what the customer needs and wants.
- It allows timely decision making. Combining Sprint Planning with effective story writing practices means that the product can be delivered sooner in a project.

WARNING!

Without Sprint Planning, you run the risk of

- A significant reduction in the attention to the priority of the work, which can result in lower-priority work getting done in place of higher-priority work
- A reduction in the team understanding of the work and limiting the amount of clarification that can regularly occur
- An increase in the amount of clarification during the sprint reducing team velocity and delivery of real commitments
- A lack of understanding the team burndown velocity and estimation data

Sprint Planning is a great setting for communication between all people concerned with a project, from the Product Owner to the team members. Participants in this communication should follow a few rules:

- The Product Owner shouldn't force the team into committing to more story points than they feel comfortable with committing to. After a number of sprints, the team's velocity becomes clear.
- After the team commits to the sprint, the sprint doesn't change: No change in the makeup of the team and no change in the sprint requirements. Of course, clarifications can be provided.
- The team can take a task and break it into two or more tasks. The team may find that a given task isn't needed and can cancel this task. For each task added, edited, or deleted, the team discusses the reasons for the change during Sprint Review, especially if they aren't able to complete the sprint goal.

Sprint Planning may seem like it has many rules, so in the next section, we break the planning down to help delineate the process.

Planning in Two Different Segments

Sprint Planning takes place on the first day of a sprint — you plan before you start the sprint work. Each Sprint Planning session lasts typically a full work day, or eight hours.

Each Sprint Planning session consists of two segments. Each segment is designed to last about half the day, roughly, or about four hours. (That leaves time for a convenient lunch break between the two segments.)

Although Scrum guidelines say that Sprint Planning typically takes eight hours and is broken down into two four-hour segments, those times can vary. Many factors affect the actual times spent in planning a sprint:

- The complexity and depth of work to be undertaken in the sprint
- How well established the team is (and have worked together)

- The length of the sprint
- The size of the team
- How well prepared the stories in the backlog are

Segment #1: Moving items from the product backlog

The first segment of Sprint Planning is focused on selecting the high-priority backlog items from the Product backlog and moving them to the Sprint backlog and defining the Sprint goal(s). Limiting this segment of the Sprint Planning to four hours makes sure that these two purposes are pursued efficiently.

Who's coming to the meeting?

Who attends the first segment of Sprint Planning? The Product Owner, the ScrumMaster, the Scrum team, and involved others:

- **The Product Owner:** It's essential that the Product Owner be present for the first segment of a Sprint Planning session. The Product Owner is responsible for setting the sprint goal, with suggestions from the team. And the Product Owner is also responsible for presenting those items that are the highest priority in the Product backlog to the team.
- **The ScrumMaster:** Keeps the planning session on track and on schedule. The ScrumMaster facilitates the meeting between the Product Owner and the Scrum team, ensuring that anyone unfamiliar with Scrum practices is brought up to speed.
- **The Scrum Team:** The team reviews the Product backlog items as prioritized by the Product Owner and asks questions about items that are unclear. Ultimately, the team is meant to estimate the time needed for each story and commit to the Sprint backlog, so they have to know what they're doing. They must analyze each story and break it down into tasks that can be performed. The entire team has to commit to the work for this iteration, so although the Product Owner presents the items from the Product backlog, it's up to the team to appropriately estimate and commit to them.
- **Others:** Others can attend a Sprint Planning session if they have information to contribute. They can act in an advisory capacity only and can't assign work or direction.

Identifying high-priority Product backlog items

The first segment of a Sprint Planning session kicks off with the Product Owner taking center stage. The Product Owner must prepare and prioritize the stories in the Product backlog before the Sprint Planning session starts.

The Product Owner presents the high-priority Product backlog items to the team in story format (these stories may be broken down into tasks in the Product backlog, and usually will be broken down into tasks in the Sprint backlog). This fosters open communication both ways between the Product Owner and the Scrum team.

TECHNICAL STUFF

Typically, the Product Owner is expected to prepare and prioritize about 50 percent more stories than the Sprint Planning session is expected to use.

Each story should be accompanied by a set of acceptance criteria (the Product Owner should make clear what constitutes successful completion of each story as the story is presented to the team). The time estimated to implement each story hasn't entered the picture yet except in a rough way — refining those estimates is the responsibility of the Scrum team in segment two of a Sprint Planning session (see the section "Segment #2: Estimating backlog items").

The first segment is all about face-to-face communication between the Product Owner and the Scrum team, with the ScrumMaster acting as facilitator. This opportunity is the chance for the team to get to know what's expected by the Product Owner in this sprint.

REMEMBER

It's important that each side understand the other here — especially the expectations for fulfilling each story in the sprint. The team is expected to ask many questions and offer suggestions if appropriate.

So, for example, the Product Owner may present the story, "As a customer, I want to insert my card into the ATM." The acceptance criteria may specify that the card be read within a certain number of seconds, verified in a certain number of seconds (and may specify what to do if the customer can't verify the card), and the card is pushed back out to the customer in another number of seconds.

Defining the goals of the sprint

As much as possible, everything about the sprint must be laid out during the Sprint Planning session, including the goals of the sprint itself. Understanding the Sprint goal is crucial for the team if it wants to create what the Product Owner wants. Adhering to the goal of the sprint at all times is important for the team.

TIP

There may be more than one goal for a sprint, but the Product Owner is expected to limit the number of goals for any sprint to three — any more than that and the sprint may become unfocussed.

The goal of the sprint is reviewed at the End-of-Sprint Review to make sure that it has been met. So for example, the goal for a sprint may be, "Complete the card-verification process by using the customer's PIN with the ATM."

The testers on the team are expected to pay especially close attention to the goal of the sprint to make sure the iteration meets that goal and then can be adequately tested.

Segment #2: Estimating backlog items

The second segment of Sprint Planning is all about honing the Sprint Backlog and understanding the backlog items in more detail. This segment has three main purposes:

- Team hones the Stories to be delivered
- Team estimates the work
- Team commits to the work and the sprint

Bringing it all into focus: Team refines the stories to be delivered

At this point, the team takes under consideration all the Sprint goals and the stories that are to be transferred to the Sprint backlog from the Product backlog. A big issue here is whether the overall goals of the sprint can be met during the time allotted for the sprint.

Do I have to go to this meeting?

Who attends the second segment of a Sprint Planning session? Check out this list to see if you're on it:

- **Scrum Team:** The team refines the backlog items as it creates the Sprint backlog, and breaks each story into tasks. It also must estimate the work. The team also commits to the work in this segment. It seeks information from the Product Owner and other sources in this segment, but all direction is up to the team and the ScrumMaster at this point.
- **ScrumMaster:** The ScrumMaster makes sure that the process goes smoothly — that the Sprint backlog is created, that the team commits to the sprint, and so on. The ScrumMaster can also act as liaison to others if the team needs more information.
- **Product Owner:** Attendance by the Product Owner is important and based on the team's need for clarification.

The team looks through the prioritized backlog items, making sure they understand each story. Members usually break down each story into tasks within the Sprint backlog.

Each acceptance criteria for a story or a task is also transferred to the Sprint backlog. If the team feels it needs further clarification on any acceptance criteria, it may confer with the Product Owner. Acceptance criteria may also change as a story is broken down into tasks.

At the completion of this stage, the Sprint backlog has begun to come into focus, and the stories and tasks in that backlog are in place. Stories may be transferred whole from the Product backlog to the Sprint backlog if they're clear enough and small enough in scope — otherwise, they will be broken up into tasks.

Get out your timesheets: Team estimates the work

The Sprint backlog has been taking shape, and the team must know if it's realistic to complete the work in the time assigned to the sprint.

Estimating the time of the work is a tricky process, but a necessary one. Scrum practices depend on being efficient with times, and the closer the team can come to estimating the time all the tasks in a sprint will take, the better. Seasoned teams will be better at this than novice teams.

A preferred approach of time estimation is to use Story Points in Scrum teams. A *Story Point* is a unit of time-work that can be assigned to each task — typically, a Story Point is one day of work for one person but this is adaptable. Most organizations actually use relative sizing to decide story points.

In a process called *Planning Poker,* the team members estimate the Story Point for each story or task. That assigns a specific amount of work and time per task.

Before this stage of the planning session, team members should consider:

- How many Story Points did they deliver last sprint? This helps set a baseline of how much work can be done in this sprint.
- Are there any holidays planned for the coming sprint? This adjusts the total points a team can complete in a sprint.
- Are there any big milestones/events taking place that may affect the process? This may impact the amount of work that can be delivered.

The team then considers a sprint velocity target. The sprint velocity is the time it will take to complete the sprint, and gives an indication of the team's development capacity.

It's very helpful here to look at a team's historical sprint velocity in trying to plan the current sprint's velocity. The team should not try to be too ambitious by setting its predicted velocity too high, which invites disappointment on the part of the Product Owner (and the team).

Each task is estimated for time, typically in Story Points, until the potential Sprint Velocity is reached. Usually, the team will start with the highest-priority stories and tasks first, estimating the time they will take, and then continue with the lower-priority stories and tasks.

Some teams have each member responsible for a story or task estimate how many Story Points (or hours) that story or task will take, and then compare their independent estimates. If the estimates don't agree sufficiently, discussion is needed.

Let's do it: Team commits to the work and the sprint

This phase is the final step of the second segment of Sprint Planning. After the stories have been added to the Sprint backlog and the team has determined what fits into the sprint according to the velocity estimate, it's time for the team to commit to the tasks.

Committing to the work is done on a member-by-member basis. This part of the second segment is facilitated by the ScrumMaster, who reviews each task, presenting it to the team.

When tasks are presented, team members volunteer to commit to the tasks. If more than one team member commits to a particular task, the ScrumMaster should facilitate discussion (and the task may end up being shared).

The ScrumMaster typically starts with the highest-priority stories or tasks, asking the team members to commit to each in turn, proceeding with the lower-priority items as the higher-priority items are taken.

In this way, tasks aren't simply assigned to team members — the team members have to step forward to accept a task. This commitment process, as far as it can go, is important to Scrum. Team members should feel that they have made a choice to "own" a particular task, accepting the responsibility for it.

The commitment process aids in team autonomy, in making sure the team can take responsibility for the sprint's tasks. Autonomy is important to the Scrum team, and letting members choose what tasks to commit to assists in the process.

Some low-priority tasks may remain uncommitted-to near the end of this stage, and it's up to the ScrumMaster to get them committed to without simply assigning them to team members. If there are issues here, the team should discuss them; no team member should feel openly coerced to take more work than they think they can manage during the sprint.

The second segment of the Sprint Planning session ends when all stories and tasks in the Sprint backlog have been committed to.

To have additional accountability, each day consists of a stand-up meeting, or Scrum, in which team members relate what they've done, what they will do, and what obstacles they face. The Daily Scrum serves to keep the team coordinated and on track. More discussion on Daily Scrums can be found in Chapter 5.

The team also keeps track of its progress by plotting completed story points on *burndown charts.* These charts are designed to display how the team is doing on reaching their goals. Any deviation from expected velocity, as shown on the burndown charts, must be addressed by the ScrumMaster.

At the end of a Sprint Planning session, a team should be able to check off all the items in the checklist in Table 4-1.

Table 4-1 A Sprint Planning Checklist

Checklist Items	*Complete? Y/N*
Clear sprint goal(s)	
Backlog items (aka, stories) in a Sprint Backlog	
Acceptance criteria for each Story/task	
Estimated Stories/Tasks	
Commitment by the team members to the Stories/Tasks	

All Things Considered: The End-of-Sprint Review

When the Sprint is complete, there's an End-of-Sprint Review meeting that takes places. The ScrumMaster, the Scrum team, and the Product Owner all attend. The Product Owner also lines up several customers who can attend.

The process proceeds as follows:

1. **The goal of the sprint is reviewed at the End-of-Sprint meeting to see if the sprint has been met.**

   Each story and/or task is enumerated, along with its acceptance criteria, to see if those criteria have been met.

2. **The results of the Sprint are presented to the Product Owner (and available customers).**

   Ideally, this is a potentially shippable deliverable of the desired product. The Product Owner determines if the results from the End-of-Sprint Review are acceptable.

3. **If everything is acceptable, the Sprint velocity is calculated as a guideline for future sprints.**

Inside Look at CA Agile Vision

Sprint Planning with CA Agile Vision is truly simple. This section gives you the inside look at CA Agile Vision and how to drag a story from the backlog to the sprint and how to track Sprint progress.

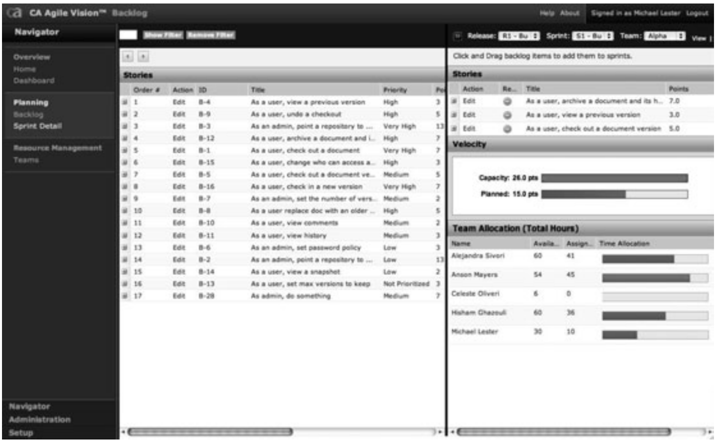
How to move stories to the Sprint backlog

You can create Sprint backlogs from the Product backlog easily. Just follow these steps:

1. **In the Backlog page, display the backlog for the project you want to work with.**
2. **Click the Show Sprints link to display the Sprint backlog, and filter the view to display the backlog for the sprint you want to work with.**
3. **Select the release, sprint, and team you're planning for.**
4. **Click and drag a user story from the project backlog and drop it in the sprint backlog.**

The user story is added to the sprint backlog and assigned to the selected team. The team's velocity reflects the assignment of the new story to the team.

You can see an example of the Sprint backlog creation in CA Agile Vision in Figure 4-1.



Release: R1 - Bu Sprint: S1 - Bu Team: Alpha

Stories

Order #	Action	ID	Title	Priority	Po
1	Edit	B-4	As a user, view a previous version	High	3
2	Edit	B-9	As a user, undo a checkout	High	5
3	Edit	B-3	As an admin, point a repository to ...	Very High	13
4	Edit	B-12	As a user, archive a document and i...	High	7
5	Edit	B-1	As a user, check out a document	Very High	7
6	Edit	B-15	As a user, change who can access a...	High	3
7	Edit	B-5	As a user, check out a document ve...	Medium	5
8	Edit	B-16	As a user, check in a new version	Very High	7
9	Edit	B-7	As an admin, set the number of vers...	Medium	2
10	Edit	B-8	As a user replace doc with an older ...	High	5
11	Edit	B-10	As a user, view comments	Medium	2
12	Edit	B-11	As a user, view history	Medium	3
13	Edit	B-6	As an admin, set password policy	Low	3
14	Edit	B-2	As an admin, point a repository to ...	Low	13
15	Edit	B-14	As a user, view a snapshot	Low	2
16	Edit	B-13	As a user, set max versions to keep	Not Prioritized	3
17	Edit	B-28	As admin, do something	Medium	7

Click and Drag backlog items to add them to sprints.

Stories

Action	Re...	Title	Points
Edit	⊖	As a user, archive a document and its h...	7.0
Edit	⊖	As a user, view a previous version	3.0
Edit	⊖	As a user, check out a document version	5.0

Velocity

Capacity: 26.0 pts

Planned: 15.0 pts

Team Allocation (Total Hours)

Name	Availa...	Assign...	Time Allocation
Alejandra Sivori	60	41	
Anson Mayers	54	45	
Celeste Oliveri	6	0	
Hisham Ghazouli	60	36	
Michael Lester	30	10	

Figure 4-1: A Sprint backlog creation with CA Agile Vision.

Tracking Sprint progress

CA Agile Vision gives you many ways to track Sprint progress and to share that information among all members of the team. Team members, product owners, and management can monitor sprint tasks and track team member progress by doing the following:

- Viewing Sprint progress charts and reports in the CA Agile Vision Dashboard page and User Stories & Charts in the Sprint Detail page
- Viewing and updating comments and notes in the Sprint Information and User Story Detail pages.
- Monitoring progress on the project Virtual Wall (see Chapter 5)

Stories and Charts on the Sprint Detail Page display charts to provide a comprehensive report of sprint progress. The view can be filtered by project, sprint, and team.

For example, a Sprint burndown chart compares the actual hours the team(s) burned on user stories against the expected burndown for the sprint.

The x axis shows the days in the sprint. All days, including weekends, are considered valid workdays. The y axis shows the task hours in the sprint. Actual hours remaining displays as a green line. The expected burndown, or guideline, displays in red. Each point on the lines is a data point representing a day in the sprint; you can see an example in Figure 4-2.

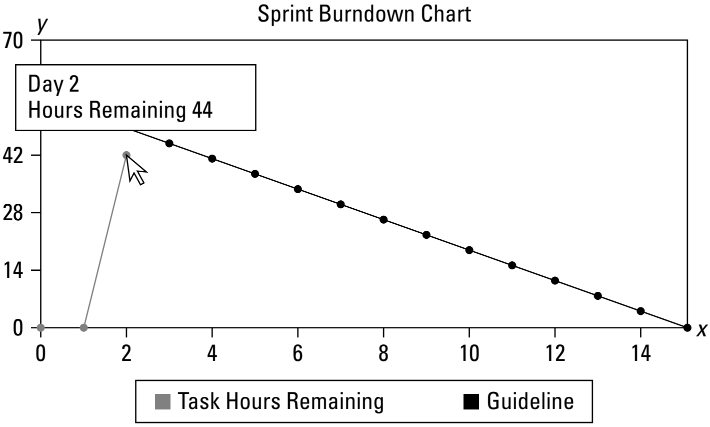


Figure 4-2: A Sprint burndown chart in CA Agile Vision.

Chapter 5

Keeping Everyone Connected

In This Chapter

- Checking the best practice of the daily Scrum
- Communication with distributed teams
- Handling large projects
- Using the virtual wall in CA Agile Vision

Communication is at the heart of the Scrum development process. Scrum practices are about meetings — the daily Scrum meeting, how to handle meetings with distributed teams, and Scrum of Scrums for larger projects in detail (for more info on Scrum of Scrums, see Chapter 3). This chapter aids in keeping everyone in touch.

Toward the end of this chapter, we show you some tools that CA Agile Vision provides to keep everyone in touch — in particular, the virtual wall and reports.

Communicating As a Team

Communication is crucial to any Scrum team. Team members should be in constant communication with each other, and with the ScrumMaster. Continual work-related conversations should take place in a Scrum team.

Both internal and external communication take place in a Scrum team. This section shows you the vitality of both.

Internal communication

Internal communication between team members is a central part of the Scrum process. Free-and-easy communication must exist within the team, because often times the team is composed of cross-discipline members who aren't used to communicating. It's the ScrumMaster's job to make sure that team members get along and communicate well, but the ScrumMaster can only do so much — communication is really up to the team members. Ultimately, team members who don't communicate well may need to be replaced.

The structure of the Scrum process facilitates communication. The daily Scrum is a stand-up meeting, with all attendees literally standing up to emphasize the brief nature of the meeting (around 15 minutes) and is all about communication. Each team member is meant to report on what they've done in the previous day, what they'll be working on today, and what roadblocks they face.

The job of the ScrumMaster here is to make sure that team members engage fully in the meeting, avoiding brief, two-or-three word answers to those questions. Ideally, each team member must understand that their answers must be complete, as though they were targeted to someone who needs to know, but may not be up-to-speed on their particular specialty, such as the Product Owner.

The ScrumMaster is also supposed to note the roadblocks mentioned by each team member and work to remove those roadblocks; some commentators on Scrum say that's the most important of all the ScrumMaster's jobs, and if the team is functioning well together, it very well might be.

External communication

External communication means communication between the team and the customer, which happens through the Product Owner. As the customer's representative, the Product Owner has to know all that's required to complete the project and be accessible at all times — perhaps even collocated with the Scrum team.

Additional external communication takes place between the Scrum team and the larger organization that team is part of. This communication concerns itself largely with project resources and boundaries — access to needed resources, such as database servers; boundaries include budget concerns and time constraints.

This type of external communication almost invariably takes place, when needed, through the ScrumMaster. In other words, the ScrumMaster is usually the liaison between the Scrum team and the larger organization that the team is part of. As much as possible, the team must be allowed to do its work without being bothered by extraneous details.

The Daily Scrum: Best Practices

The daily Scrum, or daily stand-up meeting, is the centerpiece of Scrum development. This meeting is all about communication. All attendees are requested to stand up if possible to emphasize that the meeting should be brief. Keeping the meetings brief does two things:

- Maintains a fresh, focused meeting that doesn't drag on
- Frees team members from unduly delays when their time could be better spent working

REMEMBER

Keeping the daily Scrum fresh is particularly important — you've probably been to meetings that droned on forever, wasting time and emphasizing to everyone that the meeting was being held mostly for form's sake, not the sake of function. The daily Scrum is intended to be highly functional — the name is well taken from rugby, where the team gathers briefly before launching into action. Another important aspect is that all team members are to be on time, to ensure no one is wasting other people's time.

The whole point of the daily Scrum is communication — it lets each team member know where everyone is in their work, what they're intending to do next, and what obstacles they face. It keeps at least a minimum amount of face-to-face interaction going for the team, whose members otherwise may simply disappear into cubicles.

The benefits of the daily scrum include

- Promoting commitment
- Communicating progress to the team
- Identifying obstacles so the team can remove them
- Maintaining focus
- Promoting teamwork

All Scrum projects are expected to hold daily Scrums — it's where the very name of the methodology comes from.

How the daily scrum actually works

Daily scrums are daily meetings that take place during a sprint, from the second day on to the conclusion of the sprint (the first day of a Sprint is used for Sprint planning). Daily Scrums are quick, focused, and very collaborative. They're facilitated by the ScrumMaster, but each team member must participate and speak. Each team member and the ScrumMaster (as well as the Product Owner if applicable) has the responsibility to attend every daily Scrum and to be on time.

Attendance by each team member and the ScrumMaster is particularly important, because the daily Scrum also serves as a way of identifying problems. A particular challenge faced by one team member may be the specialty of another, and this issue can initiate the right connection between team members to then talk offline after the daily standup. The daily Scrum also keeps every team member informed as to what every other team member is doing and what obstacles they face.

REMEMBER

Daily Scrums are limited to 15 minutes, although they can be as short as 5 minutes if everything is running smoothly. If discussions threaten to last more than a few minutes, they should be postponed until the Scrum is over so just those interested parties can continue the discussion in a separate meeting. Sticking to the short time frame of these meetings is important to the Scrum process to keep these meetings fresh.

The daily Scrum seeks to accomplish many goals:

- **Promotes commitment:** During the daily Scrum, team members commit to what they're doing each day. Making daily commitments to each other as a team is one of the most important goals of daily Scrums.
- **Communicates progress:** It's important to realize that the team members should feel responsible to the team. The daily Scrum promotes that responsibility — instead of reporting progress to a manager, each team member reports to the team.
- **Identifies obstacles:** When obstacles arise, it often makes sense that the team member facing the problem shouldn't have to struggle alone. If other team members can offer advice or a solution, progress isn't delayed. This kind of collaboration is what Scrum practice is all about.
- **Maintains focus:** During the daily Scrum, the ScrumMaster calls attention to the Sprint backlog. The Scrum is used to continually remind the team what that direction is so the team can focus on those tasks.
- **Promotes teamwork:** Effective teams are built by regularly communicating between, working with, and helping each other. The more contact the team members have, the more they can come to rely on each other.

So we've told you about the meetings and how important they are, but you may be wondering, "What goes on in those meetings?" Funny you should ask. Here are the down and dirty details of the daily Scrum:

- The ScrumMaster sends out a recurring meeting notice for the same time and place each day.
- The Team members show up on time.
- Each team member answers the following questions:
    - What did I accomplish yesterday?
    - What will I do today?
    - What obstacles are impeding my progress?
- The ScrumMaster notes progress and updates the sprint burndown charts. Any obstacles are added to the list to ensure they're mitigated.
- Any longer items can be followed-up with those team members interested in discussing them.

During the daily Scrum, the team members may look at the Sprint backlog, the burndown charts, the virtual wall, and the obstacle list to gain insight into progress.

Playing by the rules

When you were younger, you may have learned that life is dictated by a bunch of rules. Well, working in a highly successful business as an adult is no different. The rules of the daily Scrum include the following:

- The daily Scrum should be held at the same time each day to provide a sense of stability and ownership. Interested parties should be able to observe a daily Scrum to avoid having to schedule other meetings.
- Team members aren't allowed to force a delay or change the location of the daily Scrum.

  REMEMBER

  In general, no one person should be able to change the schedule or location. This should be a team discussion during Sprint Planning or the retrospective.
- The schedule may be modified only if the whole team agrees.
- Everyone should attend. The daily Scrum is for the team's benefit, and all Scrum members are required to attend.
- The meeting isn't postponed if someone is late (Scrum methodology is very clear on not "rewarding" latecomers).
- Each team member is expected to speak to the whole team. During the meeting, team members take turns speaking, sometimes passing along a token to indicate the current person intended to speak.
- If a team member can't attend a Scrum, he should provide his information to a proxy (such as another Scrum team member).
- Being committed. All those speaking in the daily Scrum must be Scrum team members assigned to that project and that Sprint (sometimes known as "pigs" from the story illustrating commitment in Chapter 3) or the Product Owner. Others may attend but aren't allowed to speak (these people are known as "chickens").
- Keeping it brief. The goal of a daily stand-up is to be brief. Make sure it achieves that goal.

Tying up the loose ends

So much occurs in the daily Scrum meeting that we could write a whole book just on the meeting. Do you have some lingering questions? We hope to answer some of those in this section by giving you some final points to remember.

What happens if the team falls behind?

If the team notices that it's falling behind, as shown by the burndown charts, it should bring this fact to the ScrumMaster's attention rather than simply hoping it will catch up without any corrective action.

WARNING!

Some amount of overtime may correct the issue, but prolonged and extreme overtime isn't recommended — at least in part because it would disturb the true measure of the team's velocity (and future projects may expect the same accelerated velocity). The progress and velocity issues can be discussed in the retrospective and adjusted in the next Sprint Planning session for the next sprint.

If some amount of overtime isn't enough to correct the problem, the ScrumMaster should investigate the possibilities of adding temporary team members. The additional temporary team members should be as allied with the project as possible, and the project goals should fall within their areas of expertise.

What if the team identifies an additional task?

You're team may identify additional necessary tasks that didn't come up in Sprint planning. If you have this question in your head, the answer depends on whether the new task delays the project or impacts the sprint. If the new task won't introduce a significant delay, a team member should commit to it and execute the task.

If the new task introduces a delay in project completion or impacts the sprint, it must be entered into the Sprint backlog by the ScrumMaster and treated like any other task. Confirmation of the new task may be requested of the Product Owner.

REMEMBER

The daily Scrum isn't considered a problem-solving session. The daily Scrum is a communication session — if there are issues that need more attention, interested parties should meet outside the daily Scrum.

Keeping Distributed Teams in Touch

Classically, a Scrum team is collocated in the same spot. They share the same physical working environment, including the same charts and boards listing Sprint backlog tasks. Such items can be placed in a shared team space and everyone has equal access to them.

Achieving that unity is much more difficult with a distributed team. In this case, you don't have a shared physical team space — no collocation, no physical charts, and no boards that list Sprint tasks. Instead, you'd set up an online collaboration space and use Agile planning tools like CA Agile Vision to share information and documents.

In particular, distributed Scrum teams face two significant challenges:

- **Lack of efficient meetings:** With some or most of the team elsewhere, holding efficient, coordinated meetings is a challenge.
- **Duplication of documents:** If each team site has its own copies of the project documents, such as burndown charts, backlogs, and so on, there's obviously a large risk that those documents will become uncoordinated quickly.

This section takes a closer look at solving these two problems.

Creating efficient distributed meetings

If one half of your daily Scrum takes place in San Diego, California, and the other half in Mumbai, India, you're going to have some communication problems to face.

Simply dividing the daily Scrum into two independent meetings is the least attractive option. That effectively cuts communication between the team in half, which is obviously undesirable.

Fortunately, there are a wealth of web tools that can help, as well as some other points:

- **Use short, focused meetings.** Short meetings stay on track better than longer meetings. Meetings can be run to include only critical information. Supporting documentation for the meeting should be made available to all team members at least 24 hours prior to the meeting.
- **Use Web conferencing tools.** Tools such as Microsoft's LiveMeeting are great to maintain a presence for far-flung team members. Distributed teams can approach the experience of collocated daily Scrums with the right mix of Web-based conferencing software.
- **Minimize time and regional differences.** Do the best you can to select meeting times that work the best for everyone. Teams should avoid scheduling meetings based on a single location and be aware of region differences in schedules, such as local holidays.
- **Document everything:** Web-based tools, such as CA Agile Vision, should be used for supporting information and details.

Coordinating documents virtually

Much of Scrum development depends on sharing documents — the Product backlog, the Sprint backlog, burndown charts, release planning backlogs — with the team.

WARNING!

Sharing such documents becomes much harder when there are multiple copies of documents. A burndown chart in Chicago, Illinois, runs the risk of getting out of sync with a similar burndown chart in Beijing, China. If they're two separate charts that are supposed to represent the same progress, then they'll slowly get out of sync with each other with (probably) neither being the truth. At that point, you have to reconcile at sprint end.

The solution involves hosting shared documents, charts, and schedules on the Web. These solutions include the following features:

- **Backlog management:** If all backlog item descriptions, priorities, acceptance criteria, and statuses are maintained in one place, there's less chance of miscommunication about the scope and scale of work to be done.
- **Sprint management:** This tool uses a common task board so each team member is aware of what every other team member is working on.
- **Sprint reporting:** Burndown charts and other sprint reporting metrics can provide the entire team with a quick view into the progress of the team and of individual team members.
- **Time tracking:** To make the metrics in the charts and reports meaningful, team members need to track their time.
- **Virtual walls for team communication:** Team members can post on the walls at any time, and when the wall is shared immediately throughout the world, that communication keeps the team coordinated.

TIP

All this immediate communication between members of distributed Scrum teams can be achieved instantly with a Web-based tool like CA Agile Vision. CA Agile Vision gives you the tools to host all the shared documents a Scrum team requires online to keep everyone in the loop.

Inside Look at CA Agile Vision: Adding Tasks to a Virtual Wall

When teams first start with Scrum, they use note cards to capture the user stories and tasks. They then put them on a wall and manage them there. With CA Agile Vision, the task wall becomes virtual and available to teams everywhere — and the cards can't get accidentally brushed off the wall by the cleaning staff.

CA Agile Vision specializes in keeping team members and others in touch. The *virtual wall* acts as a central focus for keeping people connected on how the sprint is going. The virtual wall enables team members to manage tasks graphically.

The Virtual Wall follows a metaphor that many teams working manually will follow. These teams create user stories and/or tasks on note cards and track their priority and progress by attaching them to a wall divided by status.

Team members can view all user stories and tasks committed for the sprint and can edit tasks and update their status right in the page. In this way, everyone on the team — and other interested parties, such as the Product Owner — can see how the sprint is going at a glance.

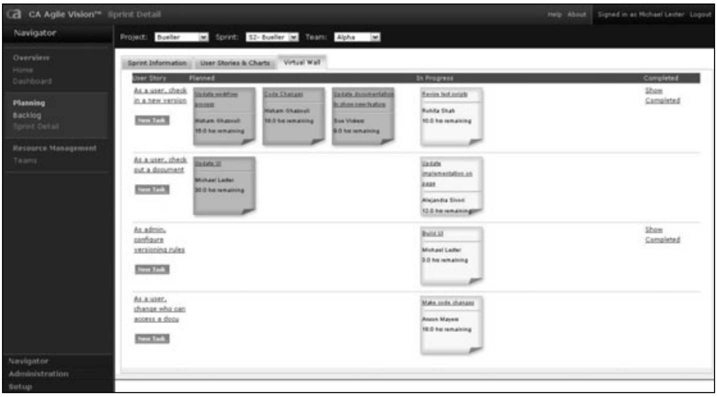
You can quickly add tasks to the Virtual Wall in CA Agile Vision; just follow these steps:

1. **Click the Navigation menu and select Sprint Detail from the Planning menu.**
2. **Display the details for the sprint that the user story is a part of, and go to Virtual Wall.**
3. **Click New Task for the user story you want to add a task.**

   A new task window is added to the user story.
4. **Double-click the task window below the name.**

   The task window redisplays with fields that you can edit.
5. **Complete the fields, which include the following:**
   - Clear the top field and enter a task title.
   - Enter the name of the team member who will be assigned to the task in the second field.
   - Enter the number of hours estimated to complete the task in the third field.
   - If the task is already started, enter the number of hours worked.
   - If the task is already started, click the right arrow button to change the task status from Planned to In Progress.
6. **Click the check button to save your settings.**

   You can see example tasks on the virtual wall in Figure 5-1.



Project: Bueller Sprint: S2- Bueller Team: Alpha

User Story | Planned | In Progress | Completed

As a user, check in a new version — Update workflow process, Hisham Ghazouli, 16.0 hrs remaining; Code Changes, Hisham Ghazouli, 16.0 hrs remaining; Update documentation to show new feature, Sue Videen, 8.0 hrs remaining — Revise test scripts, Rohita Shah, 10.0 hrs remaining

As a user, check out a document — Update UI, Michael Lester, 30.0 hrs remaining — Update implementation on page, Alejandra Sivori, 12.0 hrs remaining

As admin, configure versioning rules — Build UI, Michael Lester, 3.0 hrs remaining

As a user, change who can access a docu — Make code changes, Anson Mayers, 18.0 hrs remaining

Figure 5-1: Tasks on the virtual wall in CA Agile Vision.

Chapter 6

Ten Ways to Decide If Agile Is Right for You

In This Chapter

- Fitting Agile into your work environment
- Discovering how your team can use Agile
- Knowing if you meet all the requirements

Agile techniques aren't right for every organization. The questions in this chapter should help you to decide if Agile fits in with your project development or not.

Is Your Team Collocated?

Agile and Scrum techniques thrive when the team is located in the same place — distributed teams can introduce problems because not being collocated requires the team to work harder at communicating and collaborating. But if you can get your team together in one place or utilize an online solution to manage the Agile artifacts, that's a point in favor of Agile development.

Can You Tolerate an Empowered Team?

Agile teams must be as autonomous as possible for the Agile methodology to work. Individuals internalize their own discipline. If you can live without the need for heavy external management, that's in Agile's favor.

TIP

In organizations with a strong management style, solutions like CA Clarity PPM coupled with CA Agile Vision can help reach the goals of both Management and the Agile team.

Is the Project Large?

Large projects may not fit the Agile and Scrum methodology, which are targeted to smaller teams. Although you can break up a larger project into smaller teams, you have to be prepared for the coordination issue (such as the creation of Scrums of Scrums) that will arise. The creation of Scrum of Scrums helps. For more information on Scrum of Scrums, see Chapter 3.

REMEMBER

Large projects can be successfully implemented using Agile. In fact, many organizations are quite successful running portions of a project using Agile and other portions using traditional project management methods. With larger projects and their associated increase in challenges, your teams will require extra communication.

Is an Iterative Approach Okay?

Some organizations have a need to plan all aspects of a project from beginning to end before the project even starts. Agile works in an iterative fashion, in successive cycles, without an overall and upfront, must-do plan.

Do You Have Seasoned Developers?

Seasoned developers know what's involved in the development process and don't require as much outside guidance. They already know the ropes and are familiar enough with project development so they know what's expected of them — something novice developers may not know.

Is Your Team Motivated and Committed?

Agile and Scrum teams need to internalize the motivation needed to carry a project through so they can do their work with minimal external management. If your team is motivated and can commit to the project, that's another point in favor of considering Agile development.

Do You Have Effective Team Leadership?

Scrum teams depend on the ScrumMaster, the team leader, to see them through a project, deal with obstacles, and run the daily Scrum.

TIP

Ideally, the person you have in mind for ScrumMaster should be knowledgeable in Scrum and an effective leader without the command and control style.

Can You Tolerate a Continuous Customer Presence?

Scrum methodology calls for continuous customer involvement — ideally collocating a customer representative, the Product Owner — with the Scrum team. If you can't tolerate such a continuous customer presence, Agile and Scrum may not be for you. However, continuous customer involvement is how you know you're building the right thing for the customer.

Does the Team Have All They Need?

Scrum teams should have all they need to carry a project through. They shouldn't need significant coordination with other people and components of your organization to do their work.

Can the Customer Representative Present All Requirements?

In Scrum development, the Product Owner should ideally be the team's go-to person for the project's requirements. The Product Owner is the person ultimately responsible for the product direction and requirements.

Chapter 7

Ten Ways CA Agile Vision Can Help You

In This Chapter

- Bringing your techniques together
- Empowering your team and its learning
- Making your business life easier

CA Agile Vision is one of the most powerful online Agile development environments available today. The software specializes in bringing Scrum techniques to your organization seamlessly. This chapter is dedicated to seeing if CA Agile Vision software can help your business.

Improves Communication

When you have a distributed Scrum team, it can be nearly impossible to maintain only one copy of important documents and schedules — unless you're operating online. With CA Agile Vision, you take all planning, scheduling, and backlog tasks online for immediate coordination between distributed team members.

Saves Time

Does your Agile team spend a lot of time providing reports to Management on their progress? CA Agile Vision helps your team create reports automatically, with very few clicks, saving a great deal of time. Oftentimes, the reporting requirements drive traditional project management reports, which are designed for pre-planned projects.

CA Agile Vision, with integration to CA Clarity on Demand can provide the translation from Agile methods and vocabulary into the terms and reports that management is accustomed to.

Empowers Your Teams

Traditional project management requires the team to plan an entire project and gain approval for the changes that are encountered along the way. These changes, and the original plan, often require the team to work through existing governance processes, taking away the ability to function as Agile recommends.

CA Agile Vision and CA Clarity On Demand can provide comfort to those used to seeing a project defined as a traditional work breakdown structure (WBS) and provide visibility and feedback when many traditional project managers are unsure they can get that information from an Agile project.

Incorporates the Product Owner

CA Agile Vision specializes in getting the Product Owner involved in your project — which is essential to any Scrum project. The Product Owner has a variety of online tools for immediate access to everything about the project.

With the ability to create and manage the Product Backlog, as well as track the progress of the product through the sprints from the Dashboard, the Product Owner can feel more like a part of the team with many other solutions designed specifically for the development team in Agile projects.

Evolves Your Learning

CA Agile Vision offers many Agile-enabled tools at a variety of levels. As you get deeper into Agile development, CA Agile Vision is there with you. CA Agile Vision is built for teams at all levels of Agile maturity so the solution is approachable for everyone.

Automates Your Processes

Does your Agile team struggle with using manual processes for backlog and Sprint Planning? Everything is automated and easy to use in CA Agile Vision. The interface is intuitive and natural and makes working with backlogs and Sprint Planning a breeze.

Speaks to Your Team in Different Languages

CA Agile Vision supports users that are non-English speakers by providing a translated and localized version for those workers who speak German, Italian, Japanese, Spanish, French, and Brazilian Portuguese.

Reduces Cost

Do you want zero administration and operating costs with high performance, security, and reliability? CA Agile Vision is available as a subscription service that doesn't require your organization to purchase and maintain expensive hardware, tying up critical administrative resources.

Educates Your Beginners

Do different levels of Agile maturity exist in your organization? Not every developer is proficient in Agile methodology and techniques. To bring novices up to speed, a pre-built Agile framework with all the tools needed can be invaluable. CA Agile Vision was designed with this issue in mind and is approachable for novice Agile team members and fully featured to support more mature teams, as well.

Integrates Solutions

Do you have team members working on Agile and traditional projects? When team members have to transition between tools and methods, as is common in many large organizations, having the ability to track and manage your work, your projects, and ultimately your time in one integrated solution can be a time-saver and stress-reliever.

TIP

The integrated solution of CA Clarity on Demand and CA Agile Vision makes integration possible for the team member, as well as possible for resource managers and project management office (PMO) staff to understand what's happening within one solution.

Manage your projects more efficiently

Agile is an exciting set of principles that brings value to the customer. It's exemplified by specific practices and techniques that make product development more iterative and incremental and bring you closer to what the customer wants by involving the customer during key steps. Using the iterative approach allows customers to continuously provide feedback to get to the product they really want, and then they may eventually buy and build company revenue. Agile is also supported by products and tools. See how CA Agile Vision can help you manage a project implementing Agile effectively.

- ***Understand the Agile mindset* — walk through the Agile Manifesto and the change in thinking when you are in an Agile culture**
- ***Determine how Agile can benefit you* — how Agile can help bridge business and technology, bringing more revenue into your company**
- ***Understand the key parts of Agile* — learn Scrum and the details of Sprint Planning, Daily Scrum, Sprint Review, Retrospectives, and more**
- ***Learn how CA Agile Vision helps* — implement Agile Scrum practices**

Open the book and find:

- **Understand the elements of Agile**
- **Figure out if Agile is right for you**
- **Learn about the key Agile roles**
- **Understand how Scrum works**
- **Learn how CA Agile Vision supports Agile**

Go to Dummies.com®
for videos, step-by-step photos, how-to articles, or to shop!

Mario E. Moreira is the author of *Adapting Configuration Management for Agile Teams.* **Michael Lester** is the CA Agile Vision Product Manager at CA Technologies. **Steve Holzner** is the author of *Patterns For Dummies* and *Ajax For Dummies.*

For Dummies®
A Branded Imprint of
WILEY

ISBN: 978-0-470-87693-0
Not for resale