

- » Seeing essential scrum principles
- » Identifying scrum values and structure

Chapter 1

The Basics of Scrum

Scrum is an exposure framework based on empiricism, meaning people who employ the scrum framework gain knowledge from real-life experience and make decisions based on that experience. It's a way of organizing your work — releasing a new smartphone, coordinating your daughter's fifth-grade birthday party, or exposing whether your approach is generating intended results. If you need to get something done, scrum provides a structure for increased efficiency and more effective results.

Within scrum, common sense reigns. You focus on what can be done today with an eye toward breaking future work into manageable pieces. You can immediately see how well your effort is working, and when you find inefficiencies in your approach, scrum enables you to act on them by making adjustments with clarity and speed.

Although empirical process controls go back to the beginning of time in the arts, its modern-day usage stems from computer modeling. For example, in sculpting, you chisel away, check the results, make any adaptations necessary, and chisel away some more. The empirical exposure model means observing or experiencing actual results rather than simulating them based on research or a mathematical formula and then making decisions based on these experiences. In scrum, you break your work into actionable chunks, observing your results every step of the way. This approach allows you to immediately make the necessary changes to stay on track.

The Bird's-Eye Basics

Scrum isn't a methodology; it's a new way of thinking. It isn't a paint-by-numbers approach in which you end up with a product or outcome; it's a simple framework for clearly defining accountabilities and organizing your actionable work so that you're more effective in prioritizing and more efficient in completing the work selected. Frameworks are less prescriptive than methodologies and provide appropriate flexibility for the processes, structures, and tools that complement them. When this approach is used, you can clearly observe and adopt complementary methods and practices and quickly determine whether you're making real, tangible progress. You create usable results within weeks, days, or (in some cases) hours.

Like the process of building a house brick by brick, scrum is an iterative, incremental approach. It gives you early empirical evidence of performance and quality. Roles are distinct and self-ruling, and individuals and teams are given the required autonomy and tools to get the job done. Lengthy progress reports, redundant meetings, and bloated management layers are nonexistent. Scrum is the approach to use if you just want to get the job done.



TECHNICAL
STUFF

Scrum is a term that comes from the rough-and-tumble game of rugby. Huddles, or scrums, are formed with the forwards from one side interlocking their arms with their heads down and pushing against the forwards from the opposing team, who are also interlocking arms with their heads down. The ball is then thrown into the midst of this tightly condensed group of athletes. Although each team member plays a unique position, all team members play both attacking and defending roles and work together to move the ball down the field of play. Like rugby, scrum relies on talented people with varying responsibilities and domains working closely together in teams toward a common goal.

We want to emphasize — and have written two-thirds of this book on — an overlooked concept of scrum: its amazing versatility. People who know about scrum commonly think that it's customized for software, information technology (IT), or tech use, but that's just the tip of the iceberg. Applications for using scrum can be found everywhere, including large, small, tech, artistic, social, and even personal use. In Chapters 8 through 18, we show you how. Be forewarned! Scrum is such an addictive framework that you'll be using it to coach your kid's soccer team, plan your neighborhood watch, and even ratchet up your exercise routine.

Roadmap to value

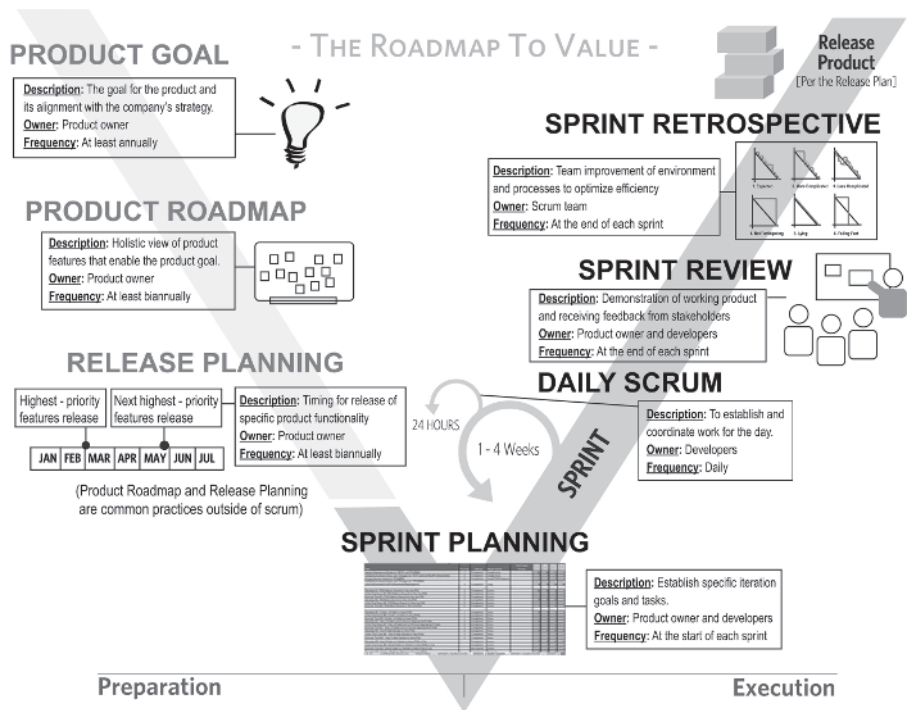
Throughout this book, we discuss techniques some expert scrum practitioners apply as common practice extensions to scrum. These techniques complement,

not replace, the scrum framework. We point out the differences when they occur. All the common practices we include and recommend are tried and tested — always with a clear understanding these practices are outside the basic scrum framework and are suggested for consideration in your own situations.

We call this aggregation of scrum and vetted common practices the “roadmap to value.” This roadmap consists of seven elements that walk you through the goal of your product to the task level and back again in a continual, iterative, and incremental process of inspection and adaptation. In other words, the roadmap to value helps you see what you want to achieve and progressively break that goal into pieces through an iterative cycle that leads to real results every day, week, and month.

You know that billion-dollar idea that’s been lurking in the back of your head for years? Follow the roadmap to value. It will show you the feasibility or fallacy of your idea and where to make your improvements — step by step, piece by piece.

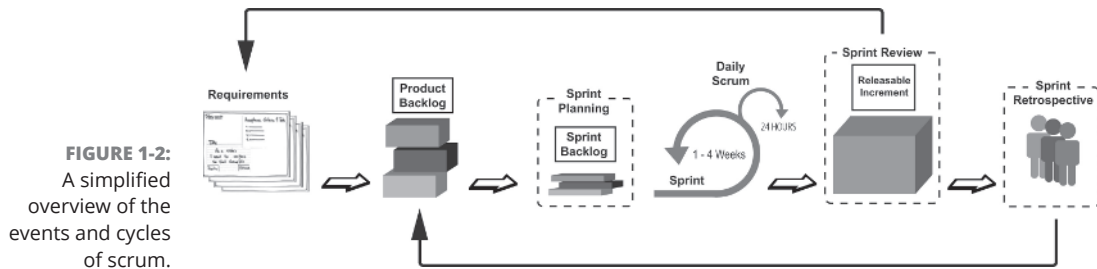
Figure 1-1 shows a holistic view of the roadmap to value. This figure shows that you begin with the product goal; work through planning; and then enter the cyclical world of sprints, reviews, and retrospectives.



Scrum overview

The scrum process is simple and circular, with constant and transparent elements of inspection and adaptation. First, a deliberately ordered to-do list — called a *product backlog* — is created and maintained. Then top-priority items are selected for a fixed, regular period — called a *sprint* — during which the scrum team strives for a predetermined and mutually agreed upon sprint goal.

Figure 1-2 shows a scrum overview.



The scrum process allows you to adapt quickly to changing market forces, technological constraints, regulations, new innovations, family preferences, and almost anything else you can think of. The key is the ongoing process of working on the highest-priority items to completion. Each of the highest-priority items gets fully developed and tested through the following steps:

1. Requirement elaboration
2. Design
3. Development
4. Comprehensive testing
5. Integration
6. Documentation
7. Approval



REMEMBER

The seven steps to fully build the scope of each requirement are performed for every item. Every requirement taken on during a sprint, no matter how small or large, is fully built, tested, and approved or rejected. When a backlog item is approved and deemed “releasable,” you know it works. Hope and guesswork are taken out of the equation and replaced by reality. You showcase these tangible results to stakeholders for feedback. This feedback generates new items that are placed in the product backlog and prioritized against existing known work.



What's more important: efficiency or effectiveness? Hands down, it's effectiveness — working on the right thing at the right time. Don't worry about efficiency until you figure out how to be effective. A very efficient team working on the wrong things is a waste of time. A super-effective team, however, can easily learn efficiency. Always work on the *right* things first. As economist and management author Peter F. Drucker said, "There is nothing so useless as doing efficiently that which should not be done at all."

The scrum cycle is run again and again. The constant flow of feedback and emphasis on developing only the highest-priority items helps you reflect what your customers are looking for, deliver it to them faster, and deliver it with higher quality.

Scrum teams

No matter what the scope of your product is, your scrum team will have similar characteristics. The sizes of teams vary somewhat, but the roles or accountabilities remain the same. We discuss the specific accountabilities in detail throughout this book. Figure 1-3 depicts a scrum team.

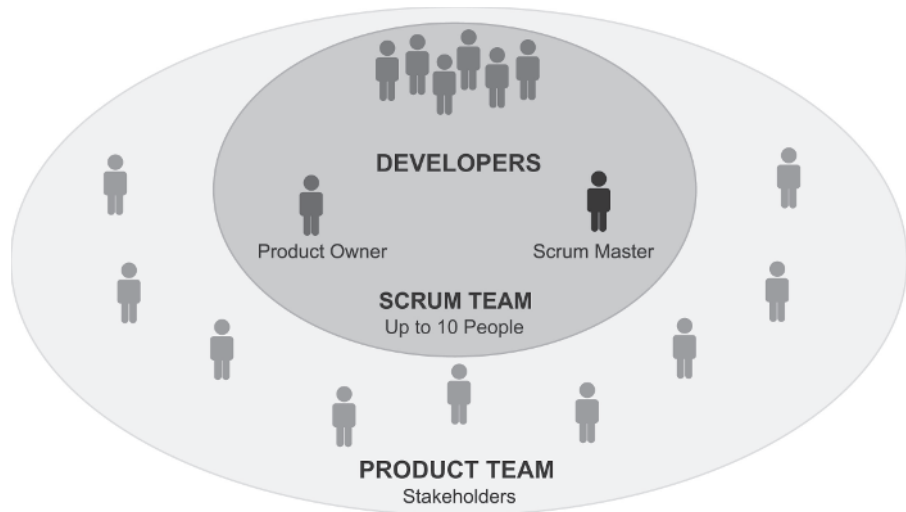


FIGURE 1-3:
A scrum team has the developers at its core.

The developers are the heart of a scrum team — the folks who work together to create the product, service, or solution itself. They work directly with a product owner and scrum master, who align business and development priorities for the organization and eliminate distractions so that the developers can focus on “developing” a quality result.



REMEMBER

Don't get hung up on the term "developer," thinking it refers to software development. Developers on a scrum team are simply the people with the skills needed to take an idea and "develop" it into something of value for the customer. Someone who writes software code is just one example of a scrum developer. There are many other skills a developer on a scrum team may possess, such as testing, writing, configuring, molding, waxing, teaching, designing, and so on. The word "developer" is used throughout this book to simplify, not exclude.

Stakeholders aren't scrum roles, but we include them in Figure 1-3 because they affect your product, service, or solution. Stakeholders can be internal or external. Marketing, legal, compliance team members, in-laws, and especially users and customers are examples of stakeholders.

The scrum team itself has ultimate accountability. Team members self-manage to figure out how to achieve their objectives within the environment in which they find themselves.

Governance

Scrum has three accountabilities that are equal in status yet separate and independent in function:

- » **Product owner:** The *why*, *what*, and *when* (not *how much*)
- » **Developers:** The *how* and *how much*
- » **Scrum master:** The *improvement* and *environment for success*

Each role has a defined purpose directly designed to enhance the team's productivity.

The creators of scrum didn't happen to devise these accountabilities by chance but through years of experience in working with all kinds of teams. They saw good, bad, and ugly combinations and found that the best results came from these three accountabilities.



TIP

We prefer that each person in a scrum team be a full-time participant dedicated solely to the scrum team's objectives. Don't thrash your team members across several priorities or use part-time players. How many professional football teams have part-time players or those who play for several teams? None.



REMEMBER

In scrum, no single person is above another. Everyone is a peer; no one is a boss or underling to anyone else on the team. *We* is the operative word rather than *I*. Scrum teams thrive on psychological safety, vulnerability, and trust — key attributes of peer-level team members.

Scrum framework

Scrum is a framework that provides clarity of responsibilities through roles, visibility through artifacts, and opportunities for inspection and adaptation through events. Within this structure, scrum is a container for other practices, processes, and tools that are appropriate for meeting the specific needs of a team, organization, product, or service.



REMEMBER

Scrum has:

- » Five values
- » Three accountabilities
- » Three artifacts
- » Five events

Each framework element fits within the scrum process, which is iterative and incremental. You incrementally create and improve your product, and you incrementally improve your process and environment with this simple framework, as follows:

» Values

- Commitment
- Focus
- Openness
- Respect
- Courage

» Accountabilities

- Product owner
- Developers
- Scrum master

» Artifacts

- Product backlog
- Sprint backlog
- Increment

» Events

- Sprint
- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective



TECHNICAL
STUFF

In the scrum world, *artifacts* are lists of work to be done or work products that have been done and are deemed valuable and usable. Unlike archaeological artifacts, scrum artifacts aren't set in stone. Scrum teams continually inspect and assess artifacts to make sure that they're digging in the right direction.

Successful implementation of the scrum framework depends on people's ability to embrace the five scrum values in their culture and thinking. The new way of working needs to change the day-to-day habits of teams and individuals.

Each role, artifact, and event in scrum has a set purpose. You place your work in the scrum framework, moving through the roadmap to value (discussed earlier in this chapter), but the actual tools and techniques for accomplishing your goals are your own. Scrum doesn't tell you how to achieve your goal; it merely provides a framework within which you can clearly see what you're doing.



REMEMBER

Scrum is an empirical process control framework. It is actually process-agnostic; it does not tell you the processes to use to create your outcomes.

Scrum is simple, but it's not easy. Scrum is much like getting into shape physically. In concept, you need to exercise more and take in fewer calories; in practice, the process can be complex.

Following are some common practices that complement scrum and have produced incredible successes. (Extra elements are shown in *italic*.) Here, we've expanded the scrum framework with recommended practices:

» Accountabilities

- Product owner
- Developers
- Scrum master

- *Stakeholders*
- *Scrum mentor*

» Artifacts

- Product goal
- *Product roadmap*
- Product backlog
- *Release plan*
- Sprint backlog
- Product increment

» Events

- *Product planning*
- *Release planning*
- Sprint
- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective

The framework is still simple but with additional roles, artifacts, and events designed to smooth the process. Throughout the book, we discuss these roles, artifacts, and events.

SCRUM'S ORIGINS

Although there have been almost 100 years of building toward the agile frameworks we use today, the first scrum team was created by Jeff Sutherland in 1993 after he applied the concepts outlined in a 1986 *Harvard Business Review* article titled, "The New New Product Development Game." With scrum co-creator Ken Schwaber, Jeff formalized the scrum framework at OOPSLA '95 (International Conference on Object-Oriented Programming, Systems, Languages, and Applications).

The Feedback Feast

One clear advantage of scrum over other frameworks is the feedback loop, which tells you early and continuously what's working, what's not working, what's missing, and what's extraordinary.

Feedback is generated regularly from scrum team members, stakeholders, and customers. The process goes something like this:

1. Developers receive daily feedback as they build their product or service.
2. Direct daily interaction occurs between the product owner and the developers for on-the-spot answers and feedback.
3. The product owner provides direct feedback as they accept or reject every completed backlog item.
4. At the end of each sprint, internal stakeholders provide feedback.
5. At the end of every release, feedback is provided by the external marketplace.

You get more from the scrum model than traditional project management models because scrum emphasizes results rather than artifact development, delivering tangible, validated products rather than tomes of reports on what's possible. You receive regular feedback along the way, enabling you to incrementally get your product or service to market as fast as possible.

In the end, you're not left wondering whether you produced what your customers want; you've been communicating with and receiving feedback from them all along the way. The inspection and adaptation process has been at work on your behalf, and you're delivering what your customers actually asked.

Agile Roots

To understand scrum, it helps to dip into the broader world of agile techniques because scrum is one of many approaches with common characteristics of agility.

Agile is a descriptor of approaches that align with the values of the Agile Manifesto and the 12 Agile Principles, which we outline in this section. Scrum is one agile approach.



TIP

For a thorough look at agile techniques, see *Agile Project Management For Dummies*, Third Edition, by Mark Layton, Steve Ostermiller, and Dean Kynaston (John Wiley & Sons, Inc.).

Three pillars of improvement

The empirical process control model sits securely on three pillars, which are common to agile techniques, including scrum:

- » Transparency
- » Inspection
- » Adaptation

Transparency

One distinguishing feature of agile techniques, in general, and scrum, in particular, is transparency. When channels of communication are clear and accessible, information is radiated broadly. Everyone involved knows what's been done, what's being worked on, what's left to work on, and any impediments blocking the way. Right from the start, you produce real results that are validated and approved or sent back for adjustments. Now, the lag time between the start date and usable results is days rather than months.

Transparency isn't just about seeing results quickly. Everyone needs to look through the same lens. A framework (such as scrum) is shared, along with an agreed-on definition of *done*. Both observers and participants can see what's being accomplished and interpret the results in a common language.

Inspection

As you discover in the following chapters, goals and outcomes are broken into the smallest actionable chunks possible (product backlog items commonly captured as user stories; see Chapter 3). Goals are set within fixed-length iterations called sprints. As each item is accomplished, it's inspected to make sure that it works and does what the customer wants.

These inspections are done by people closest to the job — those who do the work and those who represent the customer. This process eliminates the time lag required for an outside person to complete this task, and it also means that any adjustments can be made quickly because the required knowledge is at hand.

Adaptation

If the inspection shows inaccuracies and/or inefficiencies — that is, if the feature or desired outcomes don't work right — an adaptation needs to be made. The adaptation should be made as soon as possible and before moving to the next

actionable item on the to-do list. In other words, when you move on, you know everything behind you is functioning properly.

Scrum allows inspections and adaptations to be accomplished immediately at the team and product or service levels through sprint reviews, sprint retrospectives, and the daily scrum (see Chapters 6 and 7).

One Agile Manifesto

Scrum is a framework, not a by-the-numbers methodology. You still need to think and make choices. Part of the scrum framework's benefit is that it is intentionally incomplete. It allows you to make the best decisions based on the reality in which you find yourself.

In 2001, 17 software and project experts agreed on the four values their experiments and approaches had in common. These values are known as the Manifesto for Agile Software Development (commonly referred to as the *Agile Manifesto*):

We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.

Agile Manifesto © 2001: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Even though the Agile Manifesto and principles were written by and for software experts, the values remain valid for whatever scrum project you embark upon. For example, the Global Positioning System (GPS) was designed by and for the military, but that doesn't mean that you can't benefit from it when you drive to a new part of town.



TIP

For more information on the Agile Manifesto and its authors, visit <http://agilemanifesto.org>.

Twelve Agile principles

The Agile Manifesto’s authors also agreed on 12 Agile principles. You can use these principles to make sure your approach is true to agile values:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for a shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. Face-to-face conversation is the most efficient and effective method of conveying information to and within a development team.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity — the art of maximizing the amount of work not done — is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, and then tunes and adjusts its behavior accordingly.



REMEMBER

The principles don’t change, but the tools and techniques to achieve them can. While some principles explicitly reference software development, their application is much broader. Try replacing the word “software” with other words relevant to your context, such as “product,” “solution,” “functionality,” and so on.

Some of the principles are easier to implement than others. Consider principle 2. Maybe some parts of your company (or group or family) are open to change and new ideas. For them, scrum is natural, and they’re ready to get started. But other parts may be more resistant to change.

Or consider principle 6. Working face-to-face may not be possible. With the Internet, the pandemic, and the globalization of workforces, you may have team

members from Mumbai to Moscow to Miami. You could consider several video conferencing solutions, but none of these solutions meets the intention of principle 6, and none is as good as face-to-face communication. Regardless, you may actually find that only scrum allows you to maximize your results, even with a distributed team.

Your work is bound to have unique challenges. Don't let a hiccup or a less-than-perfect scenario stop you cold or cause your work to limp along. Part of the fun of using scrum is working through issues to get results. Stick with the 12 principles, and your team will get quality results quickly.

THE MARSHMALLOW CHALLENGE

In 2010, author and speaker Tom Wujec gave a remarkable TED talk called *The Marshmallow Challenge* that discussed a design exercise devised by designer Peter Skillman. In this exercise, small groups of participants are given 18 minutes to build a free-standing structure as tall as possible using strange, minimal tools: 20 sticks of spaghetti, one yard of tape, one yard of string, and one marshmallow. Wujec began giving this test and studying the results. Most groups struggled to create anything high or reliable. Group members discussed options, planned a final design, and assembled it, only to find that the structure wouldn't stand because they'd left out a crucial aspect.

The groups who performed the worst were recent graduates of business schools. The groups who performed the best were kindergarteners, who consistently produced higher and more creative structures.

Wujec said that when business students work on an idea, they believe that only one "correct" solution exists, so they spend much of their time contemplating and planning for that approach. The children, however, started by playing with the tools. They learned what didn't work and changed it; then, they figured out what did work and kept it. In other words, they built prototypes all along the way.

A takeaway in the context of scrum is that humans' natural state is inspection and adaptation. We want to do these things, but somewhere along the way, we get trained out of them. We're taught that planning and coming up with one solution is the correct way to do things. But kindergarteners remind us that this way of thinking may be wrong.

Three Platinum Principles

At Platinum Edge, our consulting and training firm, we've worked with agile organizations and scrum teams for more than two decades and consulted with dozens of companies, businesses, and not-for-profit organizations. We know how well the following "Platinum Principles" work because we've seen their value as we assisted in their implementation.

Here are three Platinum Principles that have consistently improved the performance of the teams we've helped:

- » Resist formality
- » Think and act as a team
- » Visualize rather than write

These principles can be applied to any work, not just software development. Part of the beauty of agile techniques is that you can use them for anything.

Resist formality

Have you ever seen a knockout presentation and wondered how much time someone spent putting it together? Don't even think about doing that with scrum. You can scribble it on a flip chart, stick it on a wall where people will look at it, and then get back to creating value. If discussion is required, walk over to the concerned parties (or at least spin up a quick video chat) and have the discussion. Each iteration of the design process takes very little time to visualize. Focus your valuable time and effort on accomplishing the goals instead of stylized presentations.



Pageantry is too often mistaken for professionalism and progress. With scrum, you're encouraged to communicate immediately, directly, and informally whenever you have a question and to work closely with your team members to increase efficiency and save time.

Avoid these unproductive traps:

- » Fancy, time-consuming presentations
- » Long and/or unfocused meetings
- » Tomes of documentation
- » Excessive effort justifying progress

Emphasize these productivity builders:

- » Be barely sufficient. In all things, the work should be barely sufficient to accomplish the goal. (Don't mistake sufficiency for mediocrity. Sufficient is sufficient; more is wasteful, and problems often arise in that bloat. See Agile Principle 10 earlier in this chapter.)
- » Communicate frequently with all parties to reduce the need for extensive updates.
- » Communicate simply and directly. If you can speak to someone face to face, do so.

Figure out the simplest way to get what you need, always with the goal of delivering the highest quality.



TIP

Before long, your concerted effort will evolve an agile culture. As people become educated in the process and see the improved results, their buy-in for being barely sufficient will increase. So, bear through any initial pushback with education, patience, and consistency.

Think and act as a team

The heart of scrum is working as a team, which can at first be challenging for traditional corporate cultures where individual contributions are valued most. In scrum, the results survive or die at the team level. Using each individual's talent on a team makes the work from average to hyperproductive. As Aristotle said, "The whole is greater than the sum of its parts."

How do you create this team culture? The scrum framework itself emphasizes teamwork. Physical space, common goals, and collective ownership all scream "team." Add the following practices to your scrum framework:

- » Eliminate work titles. No one owns areas of development; team members add value through their skills and collaboration. We want situational and informal leadership driven by expertise.
- » Pair team members to enhance cross-functionality and front-load quality assurance.
- » Always report team metrics, not individual or pairing metrics.

Visualize rather than write

On the whole, people are visual; they think pictorially and remember pictorially. Most kids like pictures — visual illustrations of text. Adults are no different. We're

likely to start reading a long article by flipping or scrolling through its images and charts and then reading more about what piqued our interest.

Pictures, diagrams, and graphs relay information instantly. Written reports require reader buy-in, which drops as the reports grow.



TECHNICAL
STUFF

Twitter was interested in studying the effectiveness of tweets with photos versus those that were text only. It conducted a study using SHIFT Media Manager and came up with some interesting results. Users engaged five times more frequently when tweets included photos as opposed to text-only tweets. And the rate of retweets and replies with photos was doubled. However, the cost per engagement of photo tweets was half that of text-only tweets (SHIFT Newsroom, January 17, 2014).

When possible, encourage your team to present information visually, even if that means sketching a diagram on a physical or virtual whiteboard. If someone doesn't understand the diagram, they can ask about it. The discussion coming from questions is usually valuable, as well. Changes can be made right there and then — simple graphs, charts, and models are at your fingertips.

The Five Scrum Values

Scrum is founded on five values that each member of the team uses to guide their decision-making:

- » Commitment
- » Focus
- » Openness
- » Respect
- » Courage

These values aren't rocket science. Instead, they fall into that familiar category of common sense. Yet, they're critical to the successful implementation of scrum, so they deserve discussion here.

In the following sections, we look at each of these values more closely to show how vital they are within scrum.

Commitment

Scrum team members must be committed to success and to each other. They must be willing to create realistic goals and stick to them. Everyone must participate. Scrum is an “all-in” situation in which everyone is part of a team and works together to meet the team’s commitments. Fortunately, the scrum model ensures you have the authority and freedom to do just that.

At the core of scrum is a sprint event, which we cover in Chapter 5. A sprint requires clear goals set within fixed timeboxes. In this model, you break those goals into the smallest chunks of work possible, so you know what you’re getting into. You know what’s realistic, so you can set appropriate goals and meet your commitments.

Focus

Part of the magic of scrum is that it’s built around the concept of focus. Focus on a few things at a time and do them well. You will have a clear role and clear goals within that role. Your job is to use your role to contribute to achieving the goal. Every day, team members know what to focus on for that day to be successful, which is liberating.

You made your goals and commitments earlier. Focus on those goals and nothing else.

Openness

Everything with your work, and everyone else’s work, is transparent and available for inspection and improvement. The goals, challenges, and progress of anyone involved — you, your boss, your employees, your in-laws — are open and visible. Gone are the days of six-months-down-the-road surprises.

Fortunately, the basis of scrum is the agile pillars of empiricism: transparency, inspection, and adaptation. Information radiators (big, visible charts) and real-time intelligence allow unfettered action. Most people aren’t used to this level of exposure. But after it catches on in your organization, they won’t have things any other way.

Respect

Each team member is selected for their strengths. As a result, scrum teams are diverse — diverse personalities, backgrounds, skill sets, and perspectives. Each

one is vital to success. Along with these strengths come weaknesses and opportunities to learn and grow. The golden rule within scrum is that each participant must respect everyone else. They respect each other to be capable, independent people and are respected as such by the people with whom they work.

Harmony is created by synchronizing roles and skills, which creates a development rhythm as the work progresses. If one person is out of tune for a bit, it's in your best interest to help that person because all team members are held accountable as a team.

People want to do good work; it's in our wiring. If you seek the positive, you'll find the positive. Likewise, if you seek the negative, you'll find the negative. Respect is the burning ember of positivity.

Courage

Scrum is about taking risks. It's about stepping into the unknown darkness. Every idea you have will get challenged in a scrum model. No procedure is justified because you've always done things a certain way. Say goodbye to procedures you've done by habit and say hello to a process that's built on what the team finds to be successful. Scrum team members have the courage to do the right thing, challenge the status quo, work on tough problems, and innovate ways to solve them. Philosopher Jacob Bronowski could have been speaking about the scrum model in his book *The Ascent of Man* when he said

It is important that students bring a certain ragamuffin, barefoot irreverence to their studies; they are not here to worship what is known but to question it.

Fiefdoms will be challenged. Rules will be tested. Routines will be broken. Improvements will happen. Change can be hard. Change takes courage. It takes courage to commit, focus, be open, and show and be worthy of respect — to embrace all the scrum values.

Scrum takes courage.

