The Addison Wesley Signature Series

A Mike Cohn Signature Book

Unlocking Agility

An Insider's Guide to Agile Enterprise Transformation

Jorgen Hesselberg

Foreword by

Bjarte Bogsnes,
Chairman, Beyond Budgeting Roundtable

Rich Sheridan,
CEO, Menlo Innovations

Praise for *Unlocking Agility*

"Agile has become a buzzword that has no meaning. That's because most organizations just append the word agile to what they are currently doing, shorten their 'pressure cycle,' and then expect miracles. To get serious benefits from Agile, your organization must change. And if you are serious about changing to a more agile way of working, you'll want Jorgen's help 'Unlocking Agility.'"

—**James Grenning,** Agile Manifesto Signatory and
Author of *Test-Driven Development for Embedded C*

"What a joy to read this book! From lean experimentation to agile collaboration, from business modeling to organization design, from anecdotes and stories to scaling models and transformational roadmaps, Jorgen craftily weaves it all together in this compelling book. If you were wondering where to start your agile transformation, you can start right here."

—**Jurgen Appelo,** author of *Management 3.0*
and *Managing for Happiness*

"A lot has been written about agility in enterprises. What makes this book special and extremely valuable is that it serves as an excellent and practical entry point for those who are getting started, as well as a compendium, reference, and idea-booster for those who have come far in their journey toward agility and want to take the next step.

"In each well-structured chapter, you feel that Jorgen is a very experienced practitioner, who not only knows what he is talking about, but who is also skilled in making practical use of that knowledge.

"You get a great overview of relevant models, their use, tons of great references for further self-study, and a strategic and hands-on approach to creating and driving a transformation toward agility."

—**Hendrik Esser,** *V*ice President, Enterprise
Transformation Program, Ericsson

"…we need more people looking at the big picture rather than promoting a particular practice, and this book is a valuable contribution to that work."

—**Dave Snowden,** author of *Cynefin,* and CEO, Cognitive Edge

The Iteration, or Sprint, is fixed. The PO, in collaboration with the team, selects a subset of Stories from the Product Backlog. This subset is to be completed in the Sprint, which is typically no longer than two weeks (although Scrum officially sets the limit at 30 days).

Although the Product Owner prioritizes the Stories, the team members estimate the effort required to do the work. The team then lets the PO know whether the scope of work is realistic. In other words, the people doing the work are the ones responsible for making estimates. This involvement helps create ownership and realistic commitments.

The team itself is a multidisciplinary, cross-functional group that encompasses all skills necessary to execute on the Backlog: developers, testers, User Experience (UX) designers, architects, business analysts—whichever skills, knowledge, and abilities are required to create value. The catch is that the team is not to be larger than 7 +/–2 people to keep collaboration and communication efficient. This size recommendation is based on the fact that lines of communication (and associated inefficiencies) increase dramatically as the team size increases. For instance, the lines of communication of a 6-person team is 15; the lines of communication of a 10-person team is 45, which is a threefold increase in complexity.

In addition to the PO and team, the third role in Scrum is the ScrumMaster. The ScrumMaster is the team "coach" who helps remove impediments to getting work done. The ScrumMaster stewards the process and helps the team members continuously improve the way they work.

Each day, the team meets for a 15-minute coordination session called a "Scrum" or "daily stand-up." (It's called a stand-up because people typically stand during the brief meeting to keep things moving along and keep the session focused.) In this session, team members help each other understand what they are doing to execute on the team's goals for the Sprint and whether anything is standing in the way of accomplishing what they committed to for the Sprint.

The questions asked are:

1. What did you do yesterday to help advance the goals of the Sprint?
2. What are you planning to do today to help advance the goals of the Sprint?
3. What is standing in your way of advancing the goals of the Sprint?

If impediments are uncovered, the ScrumMaster springs into action following the meeting and ensures work can continue with minimal disruption.

At the end of the Sprint, the team demonstrates the great work it has done in a "Sprint Review." This allows the Product Owner to formally sign off on what the

team accomplished. It's also an opportunity to share updates with the rest of the organization and other stakeholders. Working software is demonstrated in this session. It's important to note that this is not a mere presentation; it's about demonstrating working software to show progress toward the product goal. Sprint Reviews also lead to additional insights and learning.

The final ceremony in Scrum is called the "Retrospective." Whereas the Sprint Review focuses on the Product, the Retrospective focuses on the team members and how they work together. In the Retrospective, team members reflect on how they worked together and explore ways to do even better the next time. Although there are continuous improvement elements throughout Scrum, the Retrospective in Scrum is crucial because it focuses solely on how the team members work together and what they can do to improve how they create value as a team—not the product itself.

What Problems Does It Solve?

Focus on highly visible priorities. One of the reasons Scrum is the most popular Agile framework by far is because it helps make the 4 values and 12 principles of the Agile Manifesto concrete. In fact, because the Manifesto and Scrum are often tied to one another, many erroneously equate Scrum with Agile; however, this would be a fallacy. Agile is a mind-set and a set of values and principles that guide how to work in a more agile manner; Scrum is one manifestation of an agile way of working.

Where Scrum is most effective is in helping teams execute on a defined set of prioritized work items. Due to its short iterations, frequent feedback loops, and high degree of customer representation, Scrum has enabled companies to learn more about the customers' needs and produce value faster. Thus, it has quickly grown to become the dominant way of building software today.

What You Need to Make It Useful

For Scrum to be effective, however, several criteria need to be fulfilled. Paraphrasing Mike Cottmeyer, CEO of Leading Agile and a top organizational change consultant, being successful requires three essential things:

1. A complete team, able to deliver value as defined by the Product Backlog
2. Fully tested and integrated software available at the end of each iteration
3. A well-defined, groomed Backlog maintained by an engaged Product Owner

My experience is that Scrum is often introduced in traditional companies without accounting for these three elements. Initially, organizations tend to simply translate

existing team structures and roles and put new labels on them to reflect a Scrum team. For instance, the Project Manager becomes the ScrumMaster or the Product Requirements Document (PRD) becomes the Product Backlog.

This is a common mistake that will effectively kill the benefits of Scrum. The PRD is a static document that, once committed and signed off on, will not evolve without change orders and approvals. The Product Backlog, in contrast, is a dynamic set of priorities that is represented in various levels of detail. By simply translating the PRD into a set of user stories, you've effectively restated the same static document, not accounting for any learning in the form of customer feedback along the way. Instead, a lightweight PRD—perhaps represented as a Canvas—can be an excellent starting point to set the stage and create a common understanding among the team members, whereas the Product Backlog itself continues to evolve as more learning is acquired.

Scrum is therefore not simply a different framework with different labels; it represents a dramatically different way of working, compared to plan-driven, predictive methodologies.

Hence, for Scrum to be effective, you have to **change the way you work.** This is part of the power of Scrum: it highlights and makes visible the many flaws you have in your current processes so you can reflect on them in the Retrospective and make appropriate changes to address them. Remember how Toyota viewed defects as "treasures"—information presented to us as a way to improve our system? This is exactly the same thinking.

To get the most out of Scrum in your organization, I recommend leveraging some external coaching help, at least as the methodology is introduced. Because it represents such a fundamentally different way of working from traditional project management techniques, having some external assistance will be critical unless you've already invested in this expertise inside your organization. (See Chapter 8, "Building Your Organization's Agile Working Group.")

Practical Application

Scrum was initially built for software product development, but you can apply it in any business setting that deals with the Obvious, Complicated, and to some degree the Complex domain. Its sweet spot is probably the "Complicated" domain, because having an understanding of the "what" is essential to a Product Backlog and to executing according to commitments.

Scrum can be used in many contexts outside of software, however. My family uses Scrum to plan our week together. Each Sunday, my wife, our two kids, and I define our "Backlog" for the week and plan how we're going to get our work accomplished together. Once we've discussed the Backlog and estimated the relative effort required

to get it done, we place the work on a highly visible Scrum board visible in our living room so we can always be aware of current status and quickly help each other if any of us gets stuck with something. Although we don't have a formal ScrumMaster or Product Owner (my wife is probably the closest we have to the PO role), we self-organize around this agreed-upon Backlog and help each other get it done. It definitely beats the honey-do list!

Kanban: Taming Chaotic Environments Through Visualization

Kanban, translated from the Japanese, means "visual signal" or "card." Originally a visual technique in which an instruction card was sent along the production line to help visualize flow of work and enable "just-in-time" inventory management in manufacturing, Kanban was later adopted for knowledge work in the mid-2000s by a group of thought leaders including David Anderson and Jim Benson, both authors and management consultants.[14]

At its core, Kanban is about visualizing the work being done, limiting the amount of work done simultaneously, and continuously improving the processes and amplifying positive outcomes.

Although Scrum uses timeboxes to help constrain the system and create an implicit "rhythm" of work, Kanban removes the timeboxes and instead focuses on managing the flow of work itself through Work in Progress (WIP) limits. That is, by ensuring teams do not work on more than they can handle at once, we can reduce bottlenecks, increase speed, and improve the overall quality of the work.

Scrum is the more popular technique used in environments with more traditional software product development, but Kanban tends to be more popular in environments with less predictability and more interruptions, such as what is typically found in software maintenance or support work.

How It Works

Kanban enables teams to quickly see how work flows within their processes. It also allows teams to communicate their status and to provide a deeper understanding of the context of the work being done. As one of the leading Kanban consultancies states, "Kanban takes information that typically would be communicated via words and turns it into brain candy."[15]

A Kanban board is a simple, visual that's easily grasped within seconds, as illustrated in Figure 3.8.

Using the board, teams enumerate the work they are doing, track the progress they are making and continuously aim to improve the manner in which work gets done.

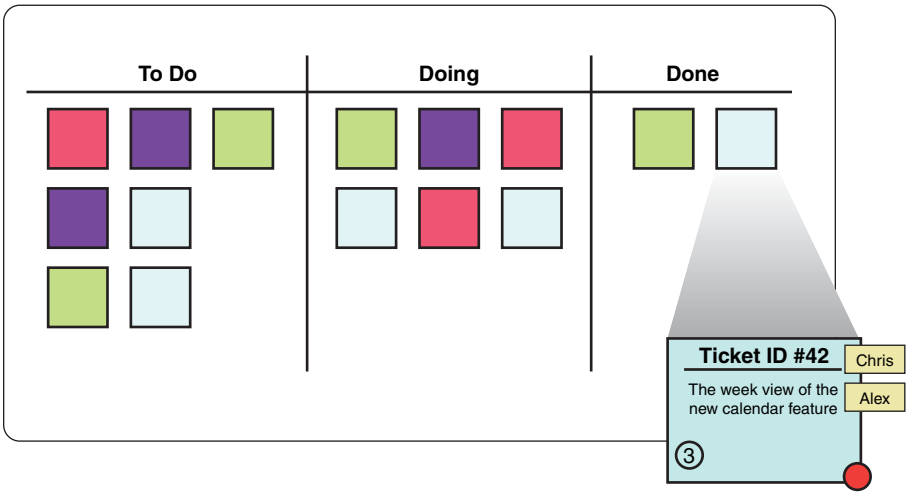

Figure 3.8 *Example of a Kanban Board*[16]

What You Need to Make It Useful

At the heart of Kanban are four principles:

1. **Visualize work.** Rather than introducing a new way of working (such as Scrum), Kanban recommends starting by visualizing the way you work right now. Having an understanding of your current state is in itself a critical component of improvement. Being able to visualize the processes, blockers, components, and dependencies involved with creating value deepens the team's understanding of what needs to improve.
2. **Limit Work in Progress (WIP).** We discussed some of the benefits of not doing too much at once in Chapter 2. Kanban recognizes this explicitly by encouraging teams to reduce the amount of unfinished work in progress. This helps increase the speed at which work is completed. It also reduces variance and helps increase quality by reducing task switching.
3. **Focus on Flow.** By setting WIP limits to manage the flow of work, teams are encouraged to continuously monitor, analyze, and improve the manner in which they work.
4. **Continuously Improve.** Once teams start changing the way they work, the resulting cultural shift spreads into the organization at large and becomes the basis for an enterprise that is focused on three goals: flow of value, managing WIP, and reducing the time to market.

Kanban can be an excellent way to introduce agile ways of working without rocking the boat by deploying new process, roles, and ceremonies. By following the Kanban principles, teams gradually gain a greater understanding of how they work and continuously improve as they track progress along the way. However, a potential drawback of this approach is that it does not require any changes to be made. If the teams do not track progress and actually manage their WIP over time, Kanban can be reduced to simply visualizing the way work is being done—and left at that. There is benefit in this from a communication and understanding perspective, but without actively managing WIP, the benefits of Kanban can be compromised.

Practical Application

Kanban is most useful when it is difficult to predict incoming work, such as support organizations that deal with a seemingly random number of support tickets arriving at various rates.

When I was coaching one of the support teams at McAfee, we quickly recognized that managing work with Scrum was not going to be very meaningful. We tried to plan our work through regular Sprint Planning every Monday, but after a few hours higher-priority tickets began arriving, and we had to replan all over again. Once we saw this become a pattern, we decided to stop trying to plan our work in two-week increments and instead make prioritization decisions on the go and focus on not working on more than we could handle at any one point in time.

Customers loved it: by collecting data on average cycle time per feature, we could predict with a fair amount of accuracy when support tickets would be completed, depending on their priority. A high-priority bug is coming in? No problem, knowing we have three high priority bugs ahead of you and that we'll spend about 3-5 days getting them done, we can confidently say you can expect a resolution sometime next week.

Kanban, without time boxes or regular iterations, is well suited for work in Snowden's "Complex" and "Chaotic" domains, where there is a fair amount of uncertainty and volatility in the nature of the work.

Building at the Right Speed: Optimizing for Flow

Identifying what your customers want and executing on the unique value proposition is critical. But to remain competitive and continuously improve the way you work, it's equally important to develop products in a way that removes waste in the process and reduces the time it takes to get from "Concept to Cash," as Mary Poppendieck, a long-time Lean thought leader, puts it.[17] In my experience working with