# 第三部分

# 执行：将Scrumban投入实践

# 第五章 动员：启用Scrumban

## 本章内容

* 框架的选择如何影响结果
* 起始分步指导
* 在改进之前使用 Scrumban 稳定团队

前面已经概述了使 Scrumban 成功的基本原则和机制，现在是讨论团队和组织如何开始使用 Scrumban 的时候了。启用 Scrumban 并不需要很大的努力。事实上, 本章概述的方法可以在一天内完成。

## 5.1 初始条件

对于启用Scrumban，基本上有三种可能的情况：

* 团队或组织还没有用过Scrum，也没有用过Scrumban；
* Scrum 团队或组织已经存在，打算使用 Scrumban 来改进其对 Scrum 框架下的职责和仪式的掌握；
* Scrum 团队或组织已经存在，打算使用 Scrumban 监视绩效，诊断问题，并调整现有的做法到最适合其上下文环境。

如果所有团队能参加一个正式的启动仪式，拥有相同的初始信息，那是最好的，但是如果没有，那并不是不可以。

### 5.1.1 如果你还不知道Scrum

尽管Scrum 是一种相对简单的框架，但是Scrum 却难以掌握。认识到这个现实，我们选择Scrumban作为框架来为团队和组织初始导入 Scrum。

因为Scrumban追求把由于强加新职责定义和责任给员工所带来的破坏降到最低，因此在这个框架下导入Scrum 与传统的方法截然不同。我们强调发现现有的系统和流程, 而不是直接开始于 Scrum 特有的引导和培训，然后使用此框架来逐步引入上下文环境所需要的Scrum 元素。

这个渐进的导入既可基于角色，也可以基于流程。例如，"每日Scrum" 是一个流程方面的变化，在 Scrumban 框架环境下团队可以一开始就使用，而对于新 Scrum master，可以每次只引入一个元素到他们的职责中，这样比较容易[[[1]](#footnote-1)]。

The Kickstart Process

Coaching Tip!

Prefer to bypass the background detail provided in this section? Check out our

quick Kickstart reference in the Appendix.

The kickstart process covered here is particularly effective when you’re faced with

limited time or resources—in other words, when teams or organizations need to better

manage workflow, but don’t have sufficient resources to develop those better ways.

Naturally, this is not the only way to introduce Scrumban to teams. 2 Always use your

understanding of Scrumban principles to modify the process to fit your context.

2. For example, the approach taken by Siemens Health Services (see the Appendix for the full case study) would

not have called for individual teams to engage in a process of systems discovery and definition.

Incidentally, this process is modeled after a Kanban kickstart process developed by

Christophe Achouiantz, a Lean/Agile coach, and Johan Nordin, development support

manager at Sandvik IT. Sandvik needed a way to empower its company’s teams to begin

a process of continuous improvement that wouldn’t involve a significant initial

investment of time or effort. Achouiantz and Nordin have documented this process and

made it available to the public as a reference guide. While the particulars may not apply

directly to your needs, it’s an outstanding summary of the key points and objectives

relevant to any process.

Preparation

As systems thinkers, it’s critical to first understand the context of the environment before

attempting to kickstart anything. At a minimum, preliminary preparation should include

the following steps:

Identifying the current conditions and organizational priorities (e.g., increased

quality, productivity, predictability)

Developing working relationships with key managers and team leads

Ascertaining areas of desired change

Introducing Kanban/Scrumban as a framework for evolutionary change

Starting to educate staff on Scrumban’s core principles and practices

Identifying any risks and potential impediments to introducing Scrumban to

targeted teams

Regarding that last item, it’s important to recognize that some teams or contexts may

not benefit from introducing Scrumban until certain conditions or impediments are

addressed. These can include teams in an active state of reorganization, teams with

serious internal conflicts, and so forth. Your context will determine how you address

these situations. For example, Sandvik’s needs dictated that such teams be bypassed

until circumstances changed.

Though it’s possible for teams to initiate Scrumban adoption without organizational

buy-in, as previous chapters indicate, broader engagement is necessary to achieve the

full breadth of desired outcomes and to sustain them over time. As such, Scrumban is

not substantially different from a pure Scrum context.

PMO/Program Managers:

If you’re overseeing a pilot program or broad-based transformation effort, we

strongly encourage you to consider the recommendations outlined here. The

choices you make in the early stages will greatly influence your ultimate success.

In his book Succeeding with Agile: Software Development Using Scrum (Boston:

Addison-Wesley, 2009), Mike Cohn addresses a variety of considerations that apply

when you’re selecting the project context in which to roll out a new process (these

considerations are made with an eye toward building off demonstrated success). Though

less critical in a context where the Scrumban or Kanban framework is employed, they

nonetheless represent worthy considerations (see Figure 5.1 for a graphical illustration

of the convergence of these considerations):

Project size: For Scrum pilots, Mike suggests selecting a project that will not

grow to more than five teams, and limiting initial efforts to just one or two of

those teams. Coordinating work among more Scrum teams represents more work

than you can effectively tackle.

For Scrumban rollouts, there’s substantially more leeway. With fewer concepts

to learn and master (again, we start out respecting current roles and processes),

working with a slightly larger number of teams is feasible.

Project duration: Select a pilot project of average duration for your organization,

ideally around 3–4 months. This is plenty of time for teams to begin mastering the

framework and seeing benefits. It’s usually also sufficient for demonstrating that

your new approach will lead to similar success in other contexts.

Project importance: Mike suggests that with Scrum pilots, it’s critical to select

high-profile projects. Unimportant projects will not get the organizational

attention necessary to make Scrum successful, and team members may be

disinclined to do all the hard things Scrum requires.

There’s slightly more leeway with Scrumban. Building momentum and trust

through success is still an important objective, but the framework’s service-

oriented agenda and active management of psychological barriers to change

represent additional capabilities that can be leveraged to ultimately satisfy these

needs.

Business owner’s level of engagement: As previously mentioned, Scrum and

Scrumban ultimately require changes in the way both the business and the

technology sides of the house approach things. Having an engaged player on the

business side can be invaluable for overcoming challenges and evangelizing

success across the organization.

Executives:

Your efforts to educate your organization’s nontechnical employees about how

this undertaking is important to the business, and encouraging their active

engagement in the process, will go a long way toward creating long-term success.

Scrumban modifies the weight of this factor substantially. To function properly,

Scrum requires active engagement on the business side—its prioritization and

feedback functions depend on it. Scrumban won’t function well without business

engagement, but it does afford teams the ability to create positive changes by

allowing knowledge discovery to “stand in” for disengaged business players.

In a similar vein, Scrumban is ultimately designed to respond to business

demands. If the business isn’t demanding change, then Scrumban’s pragmatic

“response” is to reflect the absence of any need to change (although a culture of

continuous improvement will continue to spawn incremental changes to make what

you’re already doing well even better).

Source: Michael Cohn. Succeeding with Agile: Software Development Using Scrum (Boston: Addison-Wesley,

2009).

FIGURE 5.1 The convergence of key considerations in selecting a pilot project for

introducing Scrum. These same considerations apply to Scrumban.

Initial Considerations

Many topics can be addressed during a kickstart. What constitutes “information

overload” for a particular group will vary from context to context, and must always be

judged against the level of ongoing support available following the kickstart (for

example, you will likely cover more material if teams will have coaching support

following the kickstart than if they will have none at all).

With these considerations in mind, incorporating themes that reinforce the service-

oriented philosophy and the organization’s core values will go a long way toward

positioning teams to evolve more effectively. There are many ways to do this, and it’s

one area where an experienced practitioner or coach will really add value to the

process. 3

3. As previously noted, the Siemens Health Group case study (see the Appendix) represents a great example of

how expert assistance played a critical role in contributing to the depth of the company’s success.

One special consideration for existing Scrum teams may be to introduce the concept

of Sprint 0 (if this tactic is employed within the involved team or organization). There’s

much debate about the “appropriateness” of Sprint 0 in Scrum circles. These special

sprints are often described as necessary vehicles for managing what needs to be done

before a Scrum project can start. For example, the team may need to be assembled,

hardware acquired and set up, and initial product backlogs developed. Purists’ feathers

may be ruffled by the notion of differentiating sprints by type and, more importantly, by

the idea that any sprint would be structured in a way that fails to deliver value.

The Scrumban framework obviates the need for debate on these issues. The tasks

associated with ramping up a new project can be easily accommodated and managed

within the Scrumban framework as a special work type. If it’s important for your

environment to ensure Scrum ceremonies hold true to their Agile objectives, then

Scrumban provides a ready solution.

It also makes sense to assess existing Scrum practices with an eye toward

understanding their impact on performance. Larry Maccherone and his former

colleagues at Rally Software have analyzed data from thousands of Scrum teams in an

effort to assess how differences in the way Scrum is practiced affect various elements

of performance. This data mining exposed some interesting realities about how our

choices involving Scrum practices can influence various facets of performance.

Incidentally, Maccherone’s analysis is consistent with data mined from hundreds of

teams using my own Scrum project management platform (ScrumDo.com).

Maccherone elected to adopt a “Software Development Performance Index” as a

mode of measuring the impact of specific practices. The index is composed of

measurements for the following aspects:

Productivity: Average throughput—the number of stories completed in a given

period of time

Predictability: The stability of throughput over time—how much throughput

values varied from the average over time for a given team

Responsiveness: The average amount of time work on each user story is “in

process”

Quality: Measured as defect density

How You Choose to Organize Makes a Difference

Especially For:

Managers & Executives

As noted, Larry Maccherone’s analysis reflects a definite correlation between certain

choices and software development performance. Although correlation does not

necessarily mean causation, we can use these relationships as a guide for tailoring a

kickstart process to address specific factors relevant to a particular implementation.

Teams can be guided to position their visualizations and practices to better understand

and discover which choices regarding Scrum practices are likely to work best for their

desired outcomes. 4

4. In sharing this data, Maccherone has been diligent in emphasizing that there is no such thing as “best practices”

relevant to all contexts. Each team and organization must discover what works best at any given point in time.

Determining Team Size

Humans are the slowest-moving parts in any complex organization. Teams can help us

counteract this reality by making us smarter and faster. However, this outcome is

possible only if we get teams right. Team dynamics is Scrum’s strong suit—and an

arena Kanban addresses only tangentially, if at all. In fact, providing a framework that

helps us improve team dynamics is a great example of how Scrum boosts Kanban’s

capabilities.

To this end, size stands out as the most significant predictor of team success. There’s

a right size for every team, and like so many other aspects of managing systems, that size

should be dictated by overall context. Even so, having guidelines doesn’t hurt.

The military is a great starting point for gaining perspective on ideal team size. The

basic unit in the U.S. Army’s Special Forces is the 12-person team. The army arrived at

this number after recognizing there are certain dynamics that arise only in small teams.

For example, when a team is made up of 12 or fewer people, its members are more

likely to care about one another. They’re more likely to share information, and far more

likely to come to each other’s assistance. If the mission is important enough, they’ll even

sacrifice themselves for the good of the team. This happens in business, too.

The founders of Scrum understood these dynamics. Ask anyone in the Scrum

community about ideal team size, and you’ll hear 7 members plus or minus 2.

The reasons for maintaining small team sizes are valid, but not every 12-person

military unit is right for a particular assignment. Likewise, not every 9-person Scrum

team is right for a given environment. Your system ultimately will dictate your needs,

and it may very well require expanding your teams to incorporate a larger number of

specialists or address some other need.

Scrumban supports this flexibility through its enhanced information sharing and

visualization capabilities. Even the cadence of daily stand-ups is different: We’ve seen

daily stand-ups in a Scrumban environment involving almost 100 individuals that are

both effective in addressing organizational needs and capable of being completed within

15 minutes. This would be impossible in a Scrum context.

Iteration Length

Many organizations seek to establish uniform iteration lengths because they mistakenly

believe this is a good approach for aligning different systems to the same cadence. Yet

for many years, the general consensus among Scrum practitioners has been to

recommend two-week iterations. What does the data tell us?

Rally Software’s data shows that almost 60% of the software development teams

using its tool adopt two-week sprints, and I’ve seen similar patterns among users of

ScrumDo. The data also suggests a team’s overall performance tends to be greatest at

this cadence (Figure 5.2).

Source: © 2014 Rally Software Development Corp. All rights reserved. Used with permission.

FIGURE 5.2 A view into the relationship between iteration length and team

performance.

Dig more deeply, however, and differences begin to show up among the various

components of the index. Throughput, for example, tends to be greatest among teams

adopting one-week iterations (Figure 5.3).

Source: © 2014 Rally Software Development Corp. All rights reserved. Used with permission.

FIGURE 5.3 A closer look at productivity.

Quality, in contrast, trends highest among teams adopting four-week cadences (Figure

5.4).

Source: © 2014 Rally Software Development Corp. All rights reserved. Used with permission.

FIGURE 5.4 A closer look at quality.

Every organization possesses characteristics that drive different outcomes, which is

where Scrumban’s visualization and added metrics can help teams discover what’s best

for their unique circumstances. The higher throughput typically associated with two-

week cadences won’t magically materialize if the historical lead time for most of your

completed work items is three weeks. Similarly, delivering completed work every two

weeks won’t magically produce better outcomes if the customers can’t accept it at that

rate.

Also, be wary of inferences drawn from data associated with teams that have adopted

longer iterations. For example, Frank Vega recently made an interesting observation on

this topic, recounting instances where teams chose longer iteration periods as a means

of opposing (consciously or subconsciously) Agile transformation efforts because

agreeing to a shorter iteration cycle would counteract their position that nothing of any

real value could be delivered in such a short time period.

Estimating Story Points

Scrum prescribes the Sprint Planning ceremony as an event that determines which

specific stories can be delivered in the upcoming sprint and how that work will be

achieved.

Although Scrum itself doesn’t prescribe a particular approach or method that teams

should use to assist with this decision making (other than expecting it to be based on

experience), most Scrum teams use story points and team estimation techniques as

components of this process.

Another process that teams use to aid their estimation efforts is Planning Poker—a

consensus-based technique developed by Mike Cohn. With this approach, which is used

primarily to forecast effort or relative size, team members offer estimates by placing

numbered cards face-down on the table (rather than speaking them aloud). Once

everyone has “played,” the cards are revealed and estimates discussed. This technique

helps a group avoid the cognitive bias associated with anchoring, where the first

suggestion sets a precedent for subsequent estimates.

Scrumban enables teams to begin measuring the correlation between story point

estimates and the actual time spent completing development, measured from the time

work on a story begins until the story is completed. Some teams reflect a strong

correlation between their estimates and actual work time. Others are more sporadic,

especially as the estimated “size” of stories grows. Teams using exponentially based

story size schemes tend to be more precise with their estimates. Comparing the two

charts in Figure 5.5 and Figure 5.6, it’s fairly evident that exponential estimation

reflects a tighter band of variability in lead time—especially among smaller stories.

The spectrum of variability is much wider across the Fibonacci series.

FIGURE 5.5 There tends to be only a loose correlation (if any at all) between

estimated story points and actual lead time.

FIGURE 5.6 The correlation is slightly better in an exponential points scheme.

As with the data on sprint length, the way teams estimate the size and effort of work

tends to correlate with overall performance (Figure 5.7).

Source: © 2014 Rally Software Development Corp. All rights reserved. Used with permission.

FIGURE 5.7 The relationship between team performance and estimation schemes.

It is not uncommon to find a lack of correlation between a team’s velocity, the

average number of story points completed during a sprint, and actual throughput.

Nonetheless, there are many instances when the estimating process works well for

individual teams. As mentioned previously, it’s more challenging to work with this

metric on a portfolio level involving multiple teams. Calculating relative correlation is

one way that Scrumban helps teams gain a better sense of whether the ceremonies they

engage in, such as estimation events, are providing sufficient value to justify the

investment of time and effort.

Scrumban Stories: Objective Solutions

Objective Solutions’ teams showed a lack of correlation between their story

point estimates and actual lead time, but this discrepancy wasn’t noticed until

after the company’s implementation of the Scrumban framework. As the

organization opted to continue using team estimates for forecasting purposes, its

solution for improving predictability was to apply a “velocity factor” derived

from the difference between these values. To avoid the influence of Parkinson’s

law (the tendency for the time needed to complete work to expand to the time

allowed for it), these velocity factors were not communicated to the team

members, and only used by the product owners and Scrum masters engaged in

planning.

Read the full story of Objective Solutions in the Appendix.

Contextual Responsibilities

As teams begin to employ Scrumban, members will assume a range of new

“responsibilities.”

First and foremost, team members must be groomed to challenge and question their

team’s policies, facilitating team interest in and ownership of how they work.

Fortunately, this basic concept should not be foreign to people already used to a Scrum

context. A great way to promote and support this mindset is through the establishment of

“working agreements.” 5

5. See, for example, http://tiny.cc/AgileTeamAgreements.

Not surprisingly, Scrum’s existing ceremonies and artifacts implicitly support the

notion of working agreements. Scrumban’s framework goes one step further, calling for

the team to make such agreements explicit and to visualize them on their boards as

appropriate. In many respects, they are akin to establishing an internal service level

agreement between team members.

Ultimately, we want to develop leaders who help all team members acquire the

ability to see and understand concepts like waste, blockers, and bottlenecks. Good

leaders also ensure teams don’t get stuck in a comfort zone, by prodding team members

to always seek out ways to improve.

As previously mentioned, although teams will experience work improvements simply

from employing Scrumban independently, an active management layer is essential to

realizing benefits at scale. Systemic improvements are less likely to occur without a

manager who maintains contact, helps resolve issues outside the team’s domain, and

oversees the extent to which teams devote time and effort on pursuing discovered

opportunities for improvement. Setting expectations for this need at the kickstart helps

ensure a more sustainable implementation.

Scrumban Stories: Objective Solutions

Like Siemens Health Services, Objective Solutions had successfully adopted

Scrum/XP practices in its development operations but was struggling with a

number of challenges at scale. The company adopted elements of Scrumban to

address these challenges. Not surprisingly, this journey allowed the organization

to improve many of its Scrum and pair programming practices that had been

negatively influencing throughput and quality.

For example, Objective Solutions was experiencing challenges with managing

the pair programming process: Keyboard work was not balanced, the time

required to complete work unnecessarily expanded, and mentoring benefits were

not being fully realized. Scrumban allowed this company to recognize specific

problems, and to implement processes for managing its pair programming

practices in the course of managing its regular workflow.

Read the full story of Objective Solutions in the Appendix.

The Kickstart Event

Coaching Tip!

This overview was prepared for initiative leaders, and covers broad, general

guidelines. Adjust to your context as warranted.

As with any initiative, adoption of Scrumban should start with some type of “formal”

event. Though simpler to launch than a direct transformation to Scrum, some elements of

“education” and setting expectations are still necessary to ensure a smooth start. All

team members should be physically present for any kick-off meeting. This is obviously

more of a challenge with distributed teams, so some consideration must be given to

communications and visualization. The topics that the agenda for a good kick-off event

should, at a minimum, address are covered in the following sections.

Introductory Remarks

The kick-off event is an opportunity to set the stage for what follows. Remarks should

reinforce at least two key concepts.

First, you want to underscore that this effort is solely about introducing and

employing Scrumban in the context of your current ways of working, and represents

little to no change in existing work roles or responsibilities. It’s important to address

this point up front to minimize the potential barriers to effective learning caused by fear

or resistance to change.

Second, you should reinforce the fact that this event has just one achievable outcome

—each team walking away with a clear visualization of how it works. As with

threatened change, epic objectives create unnecessary psychological barriers.

Communicating an achievable outcome up front will help engage participants early on

and make the kickstart process considerably more meaningful. We define our “definition

of done” for this step as follows:

Each team member has a clear understanding of his or her team’s purpose.

Each team has created a visual representation of its ongoing work and current

workflow.

All team members agree on their most important and relevant work policies.

Current Concerns

Service orientation lies at the core of the Scrumban framework, so it’s critical for teams

to begin with a clear identification of their stakeholders and any dissatisfaction they

might have. Getting stakeholders to work with you, instead of against you, is one of your

greatest tools for achieving continuous improvement.

Though adjustments should always be made based on the context of your environment,

Klaus Leopold recently articulated one approach for visualizing stakeholder

relationships that seems particularly informative and useful for introducing Scrumban to

an organization. 6 Among the suggestions he offers are the following:

6. See http://tiny.cc/StakeholderDissatisfact.

Visualizing the power and influence of each stakeholder using different size cards.

Visualizing the degree to which stakeholders support your initiative through a

spatial reference point: The closer to the point they’re positioned, the stronger

their level of support. Although Leopold’s article specifically speaks to

stakeholder relationships relative to undertaking a change initiative in general,

there’s no reason this approach can’t be undertaken for any initiative.

Visualizing the frequency of the relationships among stakeholders with different

style lines (e.g., dotted for infrequent or tenuous connections, other styles for

different grades).

Visualizing the quality of the relationship with special symbols (e.g., friendly or

adversarial, healthy and strong or tenuous and weak).

Why engage in this evaluation of stakeholders? Because you need to understand your

stakeholders’ sources of dissatisfaction, and find the most effective ways to work with

them toward resolving those issues. In fact, teams should change their work policies and

work visualizations only if doing so will help resolve areas of dissatisfaction.

Visualizing the dynamics associated with key relationships also creates a clear view

of who holds the most power, the current state of each relationship, and the steps that

must be taken to move closer to a solution. It helps prioritize an approach to resolving

dissatisfaction.

Next-Level Learning:

Evolving to More Advanced Practices

If you’re looking to apply an even greater level of calibration to the process, this

arena is ripe for applying a model like the Cynefin framework to provide information on

how best to manage a given relationship.

In most circumstances, it is appropriate to facilitate discussions around current

issues. Simply invite members to identify the top three to five issues that represent

irritants or impediments to their work. Any issues that relate to how work is done or

how the team interacts with other groups can be set aside for possible discussion, as

these represent way-of-working or policy matters the team will be addressing during the

kickstart. Any others should be tabled, as they don’t relate to issues relevant to the

workshop.

Defining Purpose and Success Criteria

Teams and organizations that lack a common purpose typically cannot achieve or sustain

high levels of performance. Any exercise that moves them toward a common

understanding of purpose and criteria for success is all that matters. I like to have teams

answer the question, “Why does our team exist?”

You should skip this step only when you’re sure everyone agrees they’re on a

common journey to pursue incremental change toward working more effectively.

Identifying How Work Is Done

The goal for this segment is to have teams gain a realistic understanding of their current

situation—in other words, how they actually perform work versus how they think they

perform work. We get to this point by introducing a systems perspective—by asking the

team to visualize itself as a closed system having several basic components (Figure

5.8). This exercise may be the first time many teams acquire a realistic sense of the

complexity of how work flows in and out of their domain, plus the full scope of what

they’re responsible for producing on an ongoing basis.

FIGURE 5.8 Visualizing a closed system.

The team’s articulation of upstream demand represents which work they’re asked to

perform, who makes those demands (internal customers, external customers, or a

combination of both), how those demands are communicated, and what their relative

frequency and quantity are. These understandings are used to categorize work types later

in the workshop.

Defining the downstream end reinforces the notion of recipients as consumers or

partners in a process, thereby reinforcing Kanban’s concept of knowledge work as

service delivery. These initial definitions of system boundaries clarify the scope of the

team’s responsibility—where their work begins and ends.

Articulating how the team performs its work should command most of the team’s

attention. Teams should be guided to creating a simple, high-level visualization of the

stages that each type of work must pass through before completion; this will be the

foundation upon which their working board is built. Simple and abstract is the key—no

more than 2–3 different kinds of workflows and no more than 7–10 stages in each.

Focusing on Work Types

A good kickstart process will incorporate activities that ensure team members acquire a

common understanding of work types and their significance. It may be relevant to your

organizational context to have teams identify and manage their work using predefined

types, but in any case it’s important for the team to understand the nature of the demands

placed upon them. Naturally, this is an ongoing process. Common modes of approaching

this include assessing which work types meet the following criteria:

Have the most value for their customers or partners

Are demanded the most and the least (in terms of quantity)

Are usually more urgent than others

Are best aligned with the team’s purpose (the kind of work the team should do to

a greater extent)

Are least aligned with the team’s purpose (the kind of work the team should do to

a lesser extent)

It may be relevant to your organizational context to have teams identify and manage

their work using predefined types. Nevertheless, regardless of how you elect to

proceed, it’s ultimately most important that they understand the nature of the demands

being placed upon them. Naturally, this is an ongoing process. Common categorizations

include the following:

By source (e.g., retail banking, product X, maintenance items)

By size (e.g., in terms of effort)

By outcome (e.g., production release, analysis report)

By type of flow (e.g., development, maintenance, analysis)

By risk profile (e.g., standard work, urgent work, regulatory compliance)

By relevance (how closely the work is aligned with team purpose)

Whatever scheme you choose, categorization provides a frame of reference against

which an appropriate balance of work in progress can be created and managed.

Basic Management

It’s one thing to have teams begin visualizing how they work; it’s another thing to

provide them with a framework for using these visualizations to discover better ways of

managing it. I recommend using the GetScrumban game (Figure 5.9) to introduce teams

to the basic principles behind managing their workflow. (Full disclosure: I designed

this game.) It’s usually employed in tandem with other “classroom” exercises to

illustrate and emphasize key principles and practices.

Source: GetScrumban.com.

FIGURE 5.9 The GetScrumban game lets players experience the typical evolution of

Scrum teams after introducing the Scrumban framework into their way of working.

The GetScrumban game simulates how a software development team that employs

Scrum as its chosen framework can use Scrumban’s core principles and practices to

amplify its current capabilities, overcome common challenges, or forge new paths to

improved agility. The game allows players to experiment with and experience the

impact of these principles and practices:

Expanded visualizations

Value streams

Types of work

Risk profiles

Pulling work versus assigning work

Evolutionary adjustments versus radical change

Cost of delay versus subjective prioritization

Distinct classes of service versus a single workflow

Continuous flow versus time-boxed iterations

Value of options

Whether you use an interactive tool like GetScrumban or employ some other mode of

instruction, your teams should walk away with an understanding of the following:

Concepts Already Familiar to Scrum Teams

Work items/cards (user stories): These are typically visualized on physical

boards in the form of a sticky.

Work size estimate (story points): As most Scrum teams already engage in

some form of estimation, the concept of estimating (and the notion of breaking

larger stories down into more manageable sizes) should be very familiar.

Definition of done: A short checklist of standards that must be present for a

work item to move from one column to another.

Daily stand-ups: Scrumban stand-ups tend to eliminate declarations of status

(what each team member completed, what the team member is committing to

complete, and the identification of impediments) because all of that information

is already visualized on the board. Instead, stand-ups evolve to focus on how

well work is flowing and which actions the team can take to improve overall

flow and delivery of value.

New Concepts

Work type: Often represented by the color of a work card; reflects the mix of

work in progress. Visualizing and actively managing the mix of work (i.e.,

standard user story, bug fix, maintenance item, estimations) is usually a novel

concept for Scrum teams.

Workflow: Columns on a work board represent the value-adding stages work

passes through to completion. These usually start with “Ready” and conclude

with “Done,” “Ready to Deploy,” or some similar terminology. Scrum teams

often welcome this change because story progress and individual contributions

toward it are made more visible.

Pull: Though some Scrum teams may be familiar with pull-based systems,

many others are new to this mechanism. Pull mechanisms avoid clogging the

system with too much work. Rather than work being “pushed” onto a team by

those with a demand, the team selects work to pull into their work stream when

they have the capacity to handle it.

Ready for pull: These “holding” areas are visualized as columns within a

column. Typically used when a handover occurs (such as from a development

phase to a test phase), they help manage bottlenecks.

Definition of ready: A short checklist at the bottom of a “Ready” column that

visualizes the relationship a team has with those requesting work. The

definition should specify the information or resources a team needs to

effectively begin working on an item. Though many Scrum teams may employ

definitions of ready in their work, they are rarely explicitly defined and

visualized within a working framework.

Blockers: Flags that indicate when work on an item is suspended because of

dependencies on others. Blockers are usually visualized as an additional sticky

or magnet (pink or red) on a work item. The purpose is to call attention to such

items so the team attends to removing the impediment. Some Scrum teams may

already visualize blockers on their task boards.

Classes of service: Different “swim lanes” used to call out different risk

profiles associated with given work items. We can choose to visualize separate

classes of service to reflect and manage risk better (e.g., helping to ensure

higher risk profiles attract more attention from the team than lower risk profiles.

Similarly, you may want to help the team recognize it’s okay to take longer to

complete lower-risk items.

Explicit WIP limits: Teams should limit the amount of work in progress at any

given time. We can do this by establishing explicit WIP limits across the board,

within each column (preferred), within each swim lane/class of service, by

work type, by team member, or any combination. Limiting WIP improves flow

efficiency (by reducing or eliminating the cost of context-switching, among

other things).

Common Language (Optional)

If you’re looking to align Scrumban practices across a large number of teams, it’s

usually beneficial to establish a “common language” around common concepts. It’s

possible to borrow some terms from Scrum (for example, teams might all carry a

“backlog” or items that are ready to be worked on). Similarly, it may make sense for

teams working on different aspects of the same program to use the same visualization

scheme and share common policies.

Visualization Policies

One of Scrumban’s core practices is to ensure that all work policies are explicit. We do

so to ensure that everyone is on the same page, and that the work policies can be easily

remembered and shared with others. Common practices include the following:

Work items: Most practices will be natural carry-overs from Scrum:

Due date (if any).

External reference (e.g., from a management/tracking tool).

Size of work (e.g., story points or person-hour estimates).

Start date (important to track for measurement purposes).

End date (date work was fully completed—some metrics can be impacted by

how you measure this, but any agreed-upon policy is sufficient when starting

out).

Workflow: The basic elements should already be incorporated in the systems

diagram the team developed earlier. Some columns may be adjusted and rows

added, however, as the team’s understanding develops.

Pull Criteria: Scrum practitioners familiar with Definition of Done can

optionally break it up into more granular lightweight “Pull criteria” visualized

as a combination of mandatory and optional conditions before which a pull can

be made.

What not to visualize: Cluttering up your visualization with unnecessary items

defeats the objective of bringing greater clarity and understanding to how you

work. Although teams should capture as much of their work as possible, there can

be legitimate omissions, as in the following examples:

Administrative activities (such as meetings unrelated to ongoing work).

However, there may be great value to capturing how much time meetings are

taking away from actual work, or whether certain resources are more

encumbered than others.

Short, ad hoc work (5- to 10-minute requests or incidents). As with

administrative activities, there can be value in capturing these items. Measuring

the number of such work items in the course of a day or week could reveal a

significant and ongoing demand that would otherwise fly beneath the radar.

Frequency of Synchronization

Daily meetings are as much a ceremony with Scrumban as they are with Scrum. Unlike

in Scrum, however, teams can move beyond sharing status and making commitments to

collaborating on impediments to workflow and recognizing opportunities for

improvement. This requires the team’s visual board to be in sync with reality. Some

development teams may be able to get by with once-a-day synchronization, whereas

others will need real-time updates. Decisions made in this context can impact tool

choices and other related factors.

Create a Working Board

The kickstart session is an ideal time to assist teams with setting up their working

boards. The sooner a team starts to see and work with actual items, the more relevant

the process becomes. This effort typically involves the following steps:

Drawing the workflow: Creating columns that represent the value stream of the

team’s work process.

Creating the board: I recommend that the teams create the board together,

whether it’s an electronic tool or physical board. This accelerates team learning

and ownership.

Ticket design: The information to incorporate on a work ticket will vary from

team to team. We discuss these considerations in more detail in Chapter 7.

Adding current work (self-explanatory).

This is also an ideal time for teams to establish their definition of ready and

definition of done for work items and lanes. Ready definitions can be easy to neglect,

but they help avoid potential blockages once work begins. Teams might also discuss

prioritization, but in a Scrum context this issue should have been already addressed by

product owners.

Way of Working Policies

It’s critical that all team members agree how work will be handled in their visual

boards so that it is done in a consistent manner. Areas to address include the following:

Which individual(s) will be responsible for managing the ready buffer (placing

new work items in the buffer and prioritizing them for the team to address as

capacity allows). This topic is not relevant for teams that continue to use time-

boxed sprints.

When and how the ready buffer will be replenished. For teams that continue

practicing Scrum, this usually coincides with the sprint planning process. In

complex environments, developing policies could become quite involved. The

immediate objective is simply to have a workable starting point.

Which individual(s) will be responsible for managing completed work

(especially important when work is to be forwarded to downstream partners).

How ad hoc work or requests from outside normal channels should be managed.

Consideration must be given to the needs of the system making the request.

When work should be pulled from the ready buffer (typically whenever a team

member has capacity and can’t contribute to any ongoing work). Consideration

should be included for managing WIP limits and what should occur if exceptions

are to be made.

Most of these considerations involve assessing risk (addressed further in Chapter 6).

Having teams explicitly address risk as part of the kickstart process is the best way of

ensuring more considered management as they mature. At a minimum, teams should be

encouraged to explicitly articulate how work will be prioritized and which

considerations are expected to be addressed as part of this process (i.e., is

prioritization based exclusively on market/business risks, or should the decision

incorporate an assessment of risks associated with the underlying technology, the

complexity of the work, the team’s familiarity with the domain, and other factors).

Limiting WIP

Coaching Tip!

This is an especially important topic.

Setting explicit limits on work in progress will be a new concept for Scrum teams, and

the science behind this mandate can be counterintuitive. For teams already practicing

Scrum, it may be beneficial to point out how the Sprint ceremony is an implicit WIP-

limiting mechanic. The idea is not to dive into detail, but rather to provide enough of an

overview so the team understands why limiting work in progress matters.

“Stop starting and start finishing” should become every team’s mantra. Pull

mechanisms are one way to ensure new work items are started only when a team

member has capacity to do the work, thereby enabling the team to eventually attain a

stable WIP level that matches its total capacity. Explicit WIP limits are another strategy

(but should really be reserved for more mature teams, as they require a more complete

understanding and practice of many core concepts).

To help ensure system agility, it’s essential to maximize your options. The more

tightly work is packed within a system, the less agile you become (tightly packed work

reduces your available options to respond to changes in circumstances). Limiting WIP is

one mechanism for maintaining a sufficient amount of slack to ensure a smoother flow

through the system.

A commonly used approach when introducing teams to the concept of explicit WIP

limits is to establish them based on available resources or some other factor that’s not

associated with actual demand and capacity. As with most things, the best approach will

be dictated by the team’s specific context.

As the team performs work, circumstances may call for violating WIP limits. These

are learning opportunities, and should always trigger a discussion. Perhaps current

limits are too low and should be raised. Or perhaps circumstances are such that the need

constitutes a one-time exception. Regardless of the context, WIP limits represent an

essential constraint that forces teams to improve their process and way of working.

Practice Tip

The amount of work a system should have in progress at any given time is a

matter of balance. Just as tightly packed work can impede flow, so carrying too

many options can represent waste. The proper balance is achieved over time, and

is not something to target when starting off. Nonetheless, understanding what

proper balance is expected to resemble is important to establishing any “proto”

constraints with which you elect to begin.

Teams will confront some common challenges with regard to establishing and

abiding by WIP limits (we address these in greater detail in Chapter 7). These include

the following issues:

Variability: In the form of demand, the work, the risks, and numerous other

factors.

Constraints: Constraints ultimately determine system capacity. In knowledge

work, they tend to move around, making them difficult to identify and manage.

This makes discovering and establishing the right WIP limits very challenging.

Human nature: People are funny. They often mask their true desires and

intentions in less than obvious ways.

Planning and Feedback Loops

Scrum practitioners are already used to daily check-ins. Employing Scrumban,

however, presents the opportunity to shift the meeting’s focus from status and

commitment to more proactive planning. The kanban board already provides status

information and should detail who is working on what. Impediments should also be

visualized. Consequently, the focus of the team’s discussion can shift to a collaborative

effort geared toward identifying potential impediments, dealing with requested

exceptions to policies, and otherwise addressing discoveries about how work is being

performed.

If your Scrum teams are made up of inexperienced practitioners, it’s possible they

don’t know whether their existing stand-ups are even effective. If the teams to which

you’re about to introduce Scrumban fall into this category, the kickstart is an opportunity

to help them improve.

Jason Yip, a principal consultant at the firm Thoughtworks, has effectively

summarized patterns to establish for a good stand-up (easily remembered using the

mnemonic GIFTS):

Good start: Good stand-ups should be energizing, not demotivating.

Improvement: The primary purpose of the meeting is to support improvement, not

to discuss status. 7

7. To this end, helping team members learn how to effectively engage in difficult conversations can be immensely

beneficial. See our references in the Appendix for additional guidance in this area.

Focus: Stay focused on the right things, which should be to move work through

the system (rather than dwelling on pointless activities).

Team: Good stand-ups foster effective communication and collaboration. If

people aren’t helping one another during stand-ups, something is awry.

Status: Stand-ups should communicate a basic sense of what’s going on. As

previously noted, the actual conversations move away from this information in

Scrumban (i.e., this information should be communicated through the kanban

board).

Scrum’s time-boxed Sprint remains the main vehicle for coordinating the delivery of

completed work and replenishing the team with new work. Over time, teams can opt to

modify the process for replenishment and commitment to delivery (and even de-couple

these cadences) to adopt approaches more focused on actual demand and capacity.

Sprint Reviews and Retrospectives are existing feedback loops within the Scrum

framework that we tweak in some measure when practicing Scrumban. For example,

whereas Sprint Reviews are focused on soliciting customer feedback on the delivery of

completed product, in Scrumban we incorporate the concept of reviewing overall

service delivery (borrowing from the Kanban Method’s Service Delivery and Risk

Review cadences). Similarly, the Sprint Retrospective takes on more of the flavor of the

Kanban Method’s Operations Review.

It is also important to mention the importance of conducting organizational level

feedback loops, such as the Strategy Review cadence integrated within the Kanban

Method.

Individual Flow (Optional)

Some of the concepts and techniques Scrumban employs to give teams greater control

over their collective focus and workflow are not obvious and are sometimes

counterintuitive. Devoting a portion of your kickstart program to techniques for

managing personal workflow is one way to enhance appreciation of their effectiveness.

Topics of particular relevance include the following:

Managing energy and not time (introduced via the Pomodoro technique, for

example)

The power of habits in knowledge work (by way of a brief introduction to

disciplined approaches in problem solving such as A3 Thinking, for example)

Wrapping Up

Kickstart workshops should always be closed with a summary of what the teams

achieved and the next steps that will be taken moving forward. In all instances, some

degree of continued coaching and guidance is necessary for teams to optimize their

understanding and practice.

Some Final Thoughts

Coaching Tip!

Important Concept Ahead

We’ve encountered a common misconception among Agile consultants and coaches

regarding Scrumban implementations—the notion that because kanban systems are

grounded in queuing and systems theories, there is no value in implementing them until

the systems in which they’re used are stabilized. 8

8. This mindset was recently communicated by a leading consultant in the Boston Agile community during a regular

monthly gathering.

Key Take-Away

As Deming recognized decades ago, systems should be stabilized before

undertaking efforts to improve them. Unfortunately, many Agile experts fail to

recognize that kanban systems provide a path to stability in and of themselves

(from which we can apply scientific principles in pursuit of more considered

improvements).

For example, WIP limits should ultimately be established based on the characteristics

of the stable system to which they apply. However, initial-WIP limits can be used to

bring a system into stability. I’ve pursued this path in many contexts, and this approach

is specifically cited in the Siemens Health Services Scrumban story. It’s worthy to call

out some of the particulars here.

By way of background, the Siemens group elected to refrain from imposing WIP

limits on their initial rollout. They soon discovered that the absence of WIP limits did

nothing to stabilize or reverse their existing trend of ever-increasing cycle times. A

cumulative flow diagram generated from their data showed the system was out of

balance, with teams accepting new work faster than they were delivering completed

work (Figure 5.10). Past patterns of increasing cycle times had not been influenced.

FIGURE 5.10 A cumulative flow diagram showing effort being committed to new

work increasing faster than existing work can be completed.

Upon establishing initial-WIP limits, the teams immediately began to see a

stabilization in both system lead times 9 and overall system performance (Figure 5.11

and Figure 5.12).

9. Depicted here as cycle time. The terminology in the industry can be quite confusing. Cycle time here is the time

taken to complete a work item. Refer to Chapter 7 for more precise definitions.

FIGURE 5.11 System lead times being stabilized.

FIGURE 5.12 Cumulative flow diagram indicating a stable system.

The moral of the story is clear: Yes, systems need to be stable before we can

improve them, but they also afford us mechanisms to achieve the stabilization needed to

undertake our true journey.

Tying It All Together

This chapter outlines one approach for introducing Scrumban to a team or organization.

Though not explicitly called out, it assumes the rollout is taking place in a mid-size or

larger organization with an objective of catalyzing change to key outcomes across the

enterprise.

Software engineers, project and product managers, and the other professionals with

whom they interact are generally intelligent and capable people. While this chapter has

provided enough detail for the uninitiated to begin introducing Scrumban on their own,

don’t discount the value of calling in outside experts, especially those who have

seasoned systems thinking capabilities and a deep understanding of the Kanban Method.

Indeed, the folks at Siemens Health Services made a point to call this out in the case

study of their own experience.

1. 在有些情况下，甚至Scrum Master角色是可选的。在各种类型的组织工作多年后，Frank Vega 指出在他帮助团队建立更加迭代和增量的工作方式的最早期，曾经辅导某一个团队完全去掉了这个角色，这个团队一直是他经历过的最有效的团队之一。我并不认为这是一个理想的方法, 但提到它只是为了强调基于上下文探索和决策的重要性 [↑](#footnote-ref-1)