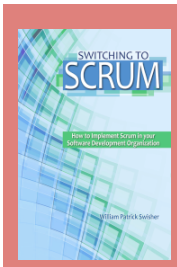


Implementing Scrumban

January 2014

A short guide to implementing Scrumban at your organization.

Introduction

This whitepaper was written for organizations already familiar with Scrum for software development and want to take advantage of Kanban's techniques for dealing with a continuous stream of work and the unpredictable needs of addressing priority business requests within a seamless flexible process.

Scrumban represents the best elements from Scrum and Kanban where the key concepts of a team working together to complete work (Scrum) and the amount of work limited to an optimized amount (Kanban) are combined into a methodology for high throughput and visibility into the development process.

As with any methodology, every organization must continually review and adapt the process and make adjustments to improve their experience to find what works best for them. This whitepaper describes the mechanics of the methodology and solutions to common scenarios to facilitate a smoother transition.

Before diving into the details, let's first look at what elements are being combined into the new methodology. The following table highlights the important ones from both methodologies in no particular order nor are they aligned for any reason.

Scrum	Kanban
• **Product Owner** – Liaison between the business and engineering organizations; owns the Product Backlog and validates completed work. • **Scrum Master** – Guardian of the process; clears impediments, and coordinates with external resources to facilitate completing Workitems. • **Developers** – Coders, Testers, Database Engineers, et cetera; anyone on the team that contributes to completing Workitems. • **Daily Standup** – A short meeting with entire team for discussing progress, identifying impediments, and addressing questions about work in progress. • **Demonstrations** – The Developers demonstrate completed work to the Product Owner for validation and acceptance. • **Delivery of Business Value** – Focus on delivering business value with each completed Workitem. • **Definition of Done** – A global standard that all Workitems must meet in order to be considered complete.	• **Pulling Work** – Work is pulled through the Workline as capacity is available versus preselecting a set of work to be done in an arbitrary period of time. • **Work in Progress (WIP) Limits** – Limits designed to maintain a balance of what the team is working on at any given time. Adjusted based on evidence with the goal of finding the optimal levels for each stage of the Workline. • **Progress Stages** – Stages for completion of work are clearly defined and performance metrics captured for each. • **Kanban Board** – A publically viewable board showing the stages of the Workline and the Workitems in progress. • **Expedited Channel** – Workitems deemed high priority and needed as soon as possible have a way of being addressed ahead of current work. • **Analysis & Adjustment** – Periodic reviews of process, tools, team performance, and WIP limits to find improvements and eliminate inefficiencies or waste.

Important Concepts to Really Understand

Key Scrumban Terms

- *Work Backlog* – Equivalent to Scrum's Product Backlog with minor differences for how Workitems are tracked.
- *Workitem* – The individual descriptions of functionality needed by the business that either define enhancements or defects.
- *Workline* – The series of stages in the software development process that pull work from the Work Backlog and produce shippable business value.
- *Stages* – The common and discrete steps required to deliver software within an organization.

Clear & Visible Stages of Execution

This concept is originally from Kanban and is a key differentiator from Scrum. The software development process is divided into discrete stages that are made very visible for anyone to see. This is known as the Workline (work + pipeline) and is represented in the real world by a Board showing the Workitems progressing with real, physical cards that can be moved around by the team members. The word *kanban* literally means 'signboard' in Japanese and lends its name to the methodology.

Pulling Work Based on Capacity & Limits

Pulling work according to capacity limits is also a key concept from Kanban. Each stage of the Workline has a limit to the number of Workitems that occupy each stage. This keeps the team members from working on too many things at once and eliminating waste by ensuring that one stage doesn't produce more work than can be handled by a subsequent stage.

As Workitems are completed within a stage, they wait until the next stage has capacity to pull new items to be processed. This is true for all stages especially the first one where Workitems are pulled directly from the Work Backlog.

Team Collaboration to Deliver Business Value

Within Scrumban, even though the discrete stages of software development are clearly identified by the Board and work is tracked through them, the entire team of Coders, Testers, Database Engineers, et cetera are expected to work together to complete Workitems. They may have specialized skills but they can all contribute to some degree and are expected to help out wherever they can when their capacity opens up. The team objective is to keep all stages of the Workline filled and Workitems progressing through them.

Empirical Analysis to Improve

The concept of reviewing the development process and team performance is the same as with Scrum but in Scrumban the statistics kept on how Workitems transit the Workline are also analyzed to determine the optimal work in progress values used to control the flow of work.

Getting Started

Identifying Stages for the Software Development Processes

A visual representation of the development process is important so the first step is to identify the stages any work goes through from when the Developers start to when they finish. Theoretically there could be dozens of steps but if the Board has too many, it will look cluttered and Workitems will be constantly jumping stages within a day. It's better to roll up many steps into discrete stages that may take a day or more to complete for each.

In addition to the common *Design, Coding,* Testing stages, if an organization had other steps such as *Legal Review* or *Webpage Design & Graphics* that were always required and typically took more than a day to complete, then those could be added to the Workline. For example:

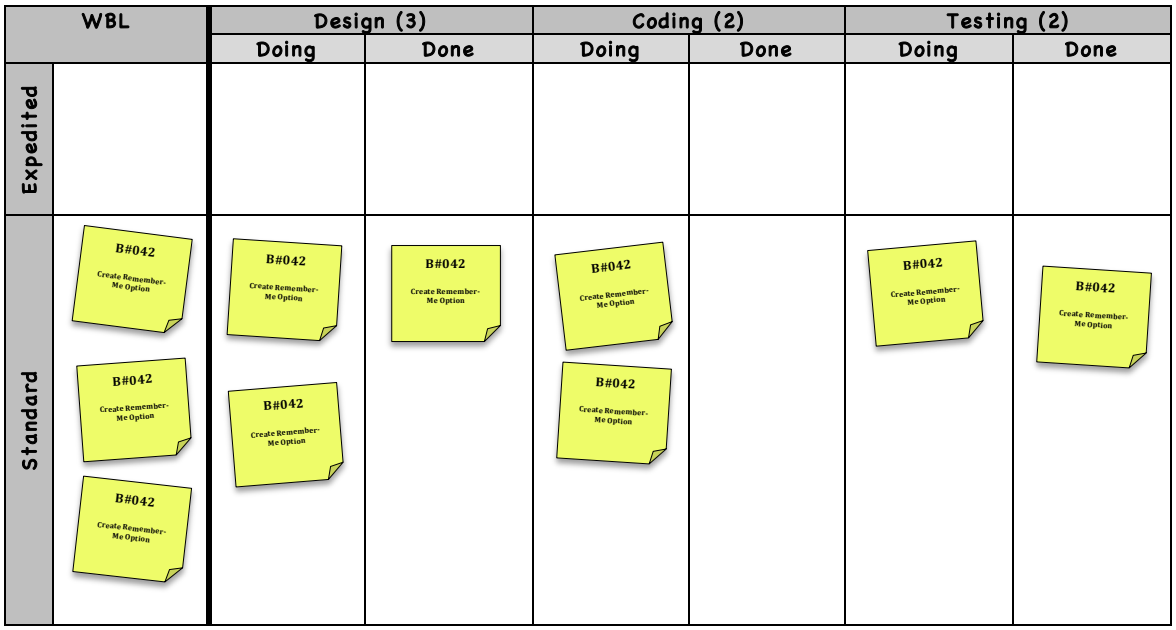
→ → → Workitems Flow Through the Workline → → →				
Webpage Design & Graphics	Code Design	Coding	Legal Review	Testing

Creating the Kanban Board

The Kanban board itself can be created in any number of ways so long as it is visible. It could be on a whiteboard, the wall, or any flat surface that labels can be written on and paper stuck to. The size should be big enough that when team members are nearby they can read the labels and titles of the work items written on the cards.

Kanban boards can also be represented through a number of online tools that are free or require a subscription. These provide the benefit of having a centralized view available to all team members, especially those in other locations. When using online tools, it's advisable to have the Board on display during the Daily Standup meeting and this can be accomplished with a large monitor in the meeting area.

The following example shows what a typical Board would look like as if it was created on a large whiteboard with sticky notes serving as the cards.



In this example there are two Workline channels, one for *Expedited* work items and the other for *Standard*. Most of the time cards will flow through the *Standard* channel with the occasional *Expedited* item going through the Workline ahead of everything else. How *Expedited* items are dealt with is covered later in this document.

When the cards show limited information about Workitems, it is advisable to have the Work Backlog easily assessable in case the team needs to lookup the details. Having a monitor displaying the backlog works well.

Defining the Initial Work-in-Progress (WIP) Limits

When you first start using Scrumban, the optimal WIP limits won't be known. It's not important to get the values perfect in the beginning because the best values will be more apparent after the team has completed a number of Workitems and there are statistics available to review. A good way to start is to take the number of team members and divide by two. If the result is an odd number, round up. For example, a team of four would result in WIP values of two and a team of five would use three. Apply the same number to all stages and see how that works out over time.

Kanban Card Contents

Kanban cards serve as a representation of the work items defined in the Work Backlog. They will not have all the up-to-date information needed by the Developers and that's not their purpose. They provide the team with a visual symbol of work and facilitate coordination between team members and status to the Product Owner.

The level of detail can vary but at a minimum, having a clear title representing the Workitem is good practice. Adding a unique identifier will make it easier to look up the item in the Work Backlog for when the team needs additional details. To the right is an example written on a typical office sticky note that can be placed on the board and moved as work progresses.

A more detailed version, but still not a complete duplicate of what's in the Work Backlog, might also include the *Priority, User Story* and space for recording statistics on the Workitem's progression through the Workline. The dates can be filled in as the card moves between stages. The addition of the *User Story* provides the Developer with the context around the request as a reminder.

B#042 Standard	**Create Remember-Me Login Option**	
As a customer, I want an option for the website to remember my login information so that I don't have to type it in each time.		
Workline Transit Statistics (in | out)		
Design	2013-08-01	2013-08-03
Coding	2013-08-03	2013-08-06
Testing	2013-08-07	2013-08-09

Whatever level of detail is included on a card, it is important to remember that it is not a substitute for the Workitem details found in the Work Backlog. The team should rely on the combination of the User Story and the Acceptance Criteria to ultimately define what should be created.

Preparing the Work Backlog

The Work Backlog under Scrumban is basically the same as with Scrum. It is a collection of *User Stories* and *Acceptance Criteria* written and maintained by the Product Owner. These individual Workitems describe the business value desired.

To keep the collection organized, additional information is associated with each item:

- *ID* – A unique identifier (e.g. B#042) for easy searches and reference.
- *Project* – A label to group a set of work items together according to a theme or purpose.
- *Title* – A concise, action-oriented description of the business value desired.
- *User Story* – A formulaic description of who will benefit from the item, what exactly is wanted, and why it is wanted. The User Story provides the context behind the request and enables the Developers to understand the goal, and not just take orders.
- *Acceptance Criteria* – A list of the specific conditions required for the Workitem to be considered complete.
- *Status* – An indicator of the readiness of the Workitem, when it is being worked on, and the final disposition. There are five elemental statuses:

↞ Progression ↞		
	Draft	The Product Owner is still working on the item and any content should not be considered final.
	New	The Product Owner has completed adding material and the item is awaiting feedback from the team.
	Ready	The team has reviewed the item and they have a good enough understanding of the item to pull it into the Workline.
	In Progress	The item is currently going through the Workline, irrespective of the stage.
	Accepted	The item has been demonstrated to the Product Owner and meets all Acceptance Criteria.

The key to having a useful Work Backlog is for the Product Owner to continually spend time "grooming" it. This includes adding new items, updating items, prioritizing, clarifying content and adapting to the needs of the Developers as he learns more about how they operate.

Scope of a Workitem

Ideally the scope of an individual Workitem should be consistent with all other Workitems so that when the average throughput of the Workline is known, predicting when a particular item in the Work Backlog will be done is possible and somewhat accurate.

However, that's not very likely as the scope of business needs can vary greatly. The Product Owner should make every attempt to break down business needs into the smallest possible size that still produces something that can be deployed or shipped and therefore provides value to the business. This is commonly known as a *Minimum Marketable Feature*.

If the Product Owner is unsure about how to reduce the scope of a Workitem, then he can consult with the team during the Weekly Timeblock when they review upcoming items. They will be able to suggest how the item can be split along logical lines or execution steps to reduce the size and still deliver value.

Establishing Reoccurring Meetings

Scrumban has only two standard meetings that provide the contact points necessary for the team to coordinate their activities (Daily Standup) and review their work (Weekly Timeblock).

The Daily Standup (DSU)

The Daily Standup meeting is a concept borrowed from Scrum and serves the same purpose; to coordinate activities and identify impediments. A daily reoccurring meeting should be set up for 30 minutes where the first half is focused on the team members updating everyone on what they did since the last meeting, what they intend to do before the next meeting, and any impediments that are blocking, or threaten to block progress on their work. The second half is for any impromptu discussions necessary to address questions or plan activities. The meeting should be held near the Board so discussions focus around work in progress and updates can be made publically. Ideally the meeting should occur early in the day, just after everyone has had a chance to get to work, reorient themselves, and think about what they'll focus on.

The Weekly Timeblock (WTB)

The Weekly Timeblock is a reoccurring meeting used for multiple purposes. It should be set up in the middle of the week (Wednesday are best) so as to avoid holidays and vacation days that typically occur at the beginning and end of each workweek.

Two to four hours is recommended and set to occur in the afternoon so any preparation work can be completed ahead of time. The team will use the time they need and leave the rest. Having a projector, screen, and ample whiteboard space will be conducive for any technical discussions and problem solving.

The first use of the Weekly Timeblock is for demonstrating any completed work. This is the time where Developers can show the Product Owner what was completed and get Acceptance. Completed work can be demonstrated outside of the meeting anytime but it is good for the team to have everyone witness the conclusion of work.

Once Workitems have been demonstrated, the team should reflect on how their development process is going, any tools they use, and review the data on how Workitems are flowing through the Workline. They should have frank discussions about any new innovation that should be reinforced or impediments to be overcome. Clear action items and their owners should be identified along with a target resolution date.

After the review of recent activity, the team should have a preview of upcoming Workitems the Product Owner is preparing. The Developers should seek to understand the intent of each item, asking questions and getting clarity so everyone is familiar with what's needed. The Developers should also provide feedback on Workitem scope, structure, and the optimal execution order so the Product Owner can make any needed adjustments to content and priority.

10-Day Scrumban Simulation

Now that the basic concepts and process have been covered, the best way to learn how Scrumban works is to step through a simulation. Let's say there's a software company who has a Scrum Team that creates various products for customers. The team has two Coders, two Testers, one Database Engineer, a Scrum Master and a Product Owner.

This team is already very familiar with Scrum and has been practicing it for two years so they understand the core concepts of the team working together to deliver completed work at the end of

each sprint. The Product Owner already has a backlog defined for an upcoming product and the team understands what the first three items are asking for.

Their last Scrum sprint ended on the previous week's Tuesday and since then they have been learning about the Scrumban process in anticipation of switching on the coming Monday. The Scrum Master has set up two reoccurring meetings. One every morning from 9-9:30am for the *Daily Standup* and the other on Wednesdays from 3-5pm for the *Weekly Timeblock* to be used by the team for any whole-team discussions, demonstrations or previews of upcoming Workitems.

In the rest of this section, each day is represented with a snapshot of the Board. On the left-most side is the Work Backlog (WBL) containing all the Workitems yet to be completed. They are created and maintained by the Product Owner and describe business needs in the form of User Stories and Acceptance Criteria. The middle section is comprised of the three standard stages the Developers go through to complete work: Design, Coding and Testing. Each of those are sub-divided into *Doing* and *Done* to indicate when a Workitem is being actively worked on or completed for that stage. Once an item is available to be demonstrated to the Product Owner, it moves to the *Demo* section of the board.

WBL	**Design** (2)		**Coding** (2)		**Testing** (2)		**Demo**
	Doing	Done	Doing	Done	Doing	Done	
[A] [B] [C] [d] [e]							

For each day of the simulation, the Board reflects the status as it is seen at the Daily Standup meeting. Workitems can move between stages or sub-stages throughout the day but the Board should represent what the current status is when the entire team is attendance.

The Work Backlog has a collection of Workitems in various states. Those represented by capital letters (e.g. [A], [B], [C]) are "Ready", that is understood by the team well enough to begin work. Items represented by lowercase letters (e.g. [d], [e], [f]…) have a state of "New" which indicates the Product Owner believes he's added enough for the team to review and possibly start working on but the team hasn't yet had a chance to review and provide feedback. Workitems can have additional states within the Work Backlog and those have been covered earlier in this document.

Day 1 (Monday)

WBL	**Design** (2)		**Coding** (2)		**Testing** (2)		**Demo**
	Doing	Done	Doing	Done	Doing	Done	
[C] [d] [e] [f] [g]	[A] [B]						

- During the first Daily Standup meeting, the board is clear and so the team pulls in two items, [A] & [B], from the top of the Work Backlog. They select a maximum of two because that's all the *work in progress (WIP) limit* allows. The WIP limit is kept on the Board next to each stage name as a reminder.
- After the Daily Standup is over, the Developers get together and go about designing what they'll need, the implementation strategy, the testing strategy and anything else that might help them plan out how to complete the work. This group activity helps the entire team understand and prepare themselves to complete the work.
- The design effort doesn't take all day so after a couple of hours, the Coders get down to coding and the Testers start their preparation work.

Day 2 (Tuesday)

WBL	Design (2)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[d] [e] [f] [g] [h]	[B] [C]		[A]				

- During the Daily Standup the card representing [A]'s work is moved to the *Doing* column of the *Coding* stage. It could have been placed in the *Done* column of the *Design* stage for the brief time before the Coders started working but since it transitioned quickly it was moved to the next stage directly during the Daily Standup.
- At this point, the *Design* stage has capacity and so the Developers pull [C] into the Workline.
- The Scrum Master volunteers to set up a design meeting that day meeting to finish off [B] and start [C].
- The *Coding* stage has a WIP limit of two but since only one item is available the capacity goes unfilled.

Day 3 (Wednesday)

WBL	Design (2)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[d] [e] [f] [g] [h]		[C]	[B]		[A]		

- The Scrum Master points out that there is capacity within the *Design* stage but that no work items are in the 'Ready' state. The team could try to work out the details about item [d] after the standup but decides to wait until the *Weekly Time Block* meeting at 3pm to review it and the next few items all at once.
- One of the Testers, seeing that [A] was *Done* and ready for testing, moves [A]'s card to the *Doing* column of the *Testing* stage to indicate that's what he'll work on next.

Day 4 (Thursday)

WBL	Design (2)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[F] [g] [h] [i] [j]	[D] [E]		[C]		[B]		[A]

- Coding for [B] was completed and the card is moved to the *Done* column of the *Coding* stage. One of the Testers, knowing that the stage has capacity, immediately moves the card to the *Doing* column in the *Testing* stage.
- Testing for [A] was also completed and is ready to demonstrate to the Product Owner. This could be done anytime but the team decides to do it at the *Weekly Time Block* next Wednesday. This is so they don't interrupt their rhythm right now and it ensures everyone on the team can participate. The card stays in the *Demo* column until that time.
- The Scrum Master points out that the Coding & Testing stages have capacity but since no Workitems are available, they are not filled. Chronic spare capacity suggests the WIP values

for previous stages should be higher and this should be discussed at the upcoming *Weekly Timeblock*.

Day 5 (Friday)

WBL	Design (3)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[G]	[D]		[E]		[C]		[A]
[I]	[F]						[B]
[j]	[H]						
[k]							
[l]							

- Testing for [B] was completed and it was moved to the *Demo* column.
- [D] has proven to be difficult to design and there are a number of outstanding questions being researched by the Developers so it remains in the Design stage.
- Coding has started on [E] and testing has started on [C].
- The Developers pull [F] into Design and decide that because of their chronic spare coding capacity to increase the WIP for the Design stage to 3. Because no Workitems are Ready, they have an impromptu meeting with the Product Owner to review the next few items. [H] turns out to be relatively easy to implement and it is pulled into the Workline.

Day 6 (Monday)

WBL	Design (3)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[j]	[G]	[H]	[D]		[E]		[A]
[k]	[I]		[F]				[B]
[l]							[C]
[m]							
[n]							

- [D] & [F] are pulled into the Coding stage which opens up capacity into Design stage so [G] & [I] are selected from the work backlog.
- With a higher WIP value in the Design stage, the entire team is spending a little more time each week in design meetings, which then results in more preparation by the Testers. This isn't necessarily something to be concerned about but could be a topic for discussion at the next *Weekly Timeblock* after Workitem demonstrations.

Day 7 (Tuesday)

WBL	Design (3)		Coding (2)		Testing (2)		Demo
	Doing	Done	Doing	Done	Doing	Done	
[l]	[J]	[I]	[G]	[H]	[F]		[A]
[m]	[K]				[D]		[B]
[n]							[C]
[o]							[E]
[p]							

- The Developers needed to pull in two additional items into the Design stage and since no items were Ready, they stayed after the Daily Standup to review the next two [J] & [K], and pull them into the Workline. This is the second time they've had to review work items outside of the Weekly Timeblock meeting so it suggests they need to dedicate more time to review further ahead in the Work Backlog. This will be a topic for discussion on Wednesday.
- The change in WIP for the Design stage seems to have taken care of the chronic over capacity as the Board now shows plenty of activity across all stages.

- The Demo column also now has four items to be demonstrated to the Product Owner at the next *Weekly Timeblock* meeting so they'll make good use of the two hours.

Day 8 (Wednesday)

WBL
[M]
[n]
[o]
[p]
[q]

Design (3)		**Coding** (2)		**Testing** (2)	
Doing	Done	Doing	Done	Doing	Done
[L]	[J] [K]	[I]	[H]	[F] [G]	

Demo
[A]
[B]
[C]
[E]
[D]

- The team pulled [L] into the Workline after having reviewed it with the Product Owner in a mid-morning meeting. Before then, there were no items Ready. The repeated shortage of Ready items will definitely need to be addressed.
- The Coders finished off [G] & [H] and pulled [I] into the Coding stage.
- At the *Weekly Timeblock* meeting, the Developers demonstrate work items [A] – [E] which are all accepted by the Product Owner.
- After the demonstrations, the Scrum Master facilities the analysis discussion. The goal of which is to identify *innovation* that could help the team deliver work or *impediments* that are inhibiting them. The team discusses the recent WIP limit increase for the Design stage and all agree it was a good idea. However, because they are pulling more Workitems into the Workline, they're running out of items in the Ready state. They could continue to have impromptu review meetings but find that disruptive and so decide to make sure they spend enough time each Wednesday during the *Weekly Timeblock* to get more Ready.
- In the remaining time, the team reviews the upcoming Workitems and the next four are switched to the Ready state.

Day 9 (Thursday)

WBL
[O]
[P]
[Q]
[r]
[s]

Design (3)		**Coding** (2)		**Testing** (2)	
Doing	Done	Doing	Done	Doing	Done
[N]	[L] [M]	[I] [J]		[H] [K]	

Demo
[F]
[G]

- The Demo column was cleared because all the items were accepted during the Weekly Timeblock meeting.
- Testing for [G] was completed and it moves to the Demo column. [K] is pulled into the Testing stage.
- The Coders complete [L] & [M] and pull [N] into the Design stage. The team has a lot going on and postpones its design meeting until the following day.
- The Scrum Master updates and posts the Workline Transit Statistics showing how long each Workitem spent in each stage and the average transit time for the past five items. The average transit time is useful for knowing how long a Workitem is expected to take once it enters the Workline. Knowing this, the Product Owner can predict when specific items still waiting in the Work Backlog will be completed. Using the values for the last five items completed, the average transit time is 3.6 days.

Workline Transit Statistics					
Work Item	**Design (days)**	**Coding (days)**	**Testing (days)**	**Total (days)**	**Five Day Average**
[A]	1	1	1	3	3
[B]	2	1	1	4	3.5
[C]	2	1	1	4	3.7
[E]	1	1	1	3	3.5
[D]	2	1	1	4	3.6
…	…	…	…	…	…

Day 10 (Friday)

WBL	**Design** (3)		**Coding** (2)		**Testing** (2)		**Demo**
	Doing	Done	Doing	Done	Doing	Done	
[O] [P] [Q] [r] [s]	[O]	[M] [N]	[I] [L]		[K] [J]		[F] [G] [H]

- Testing for [H] is completed and it moves to the Demo column so the Testers pull in [J].
- The Coders pull in [L] but decide to concentrate together on [I] because of the complexity.
- The Design stage was down to two items so [O] is pulled into the Workline although the team decides to hold the design meeting the following day since so much is in progress.

Common Process Questions

When the progress of a Workitem through the Workline is blocked, how should this be handled?

As a team is working on items, it will be fairly common that progress from one stage to another is blocked. These blockages can be *internal* to the team where the work was more challenging than expected. Whatever the cause the team should "swarm" to help wherever possible to remove the blockage. All team members can assist to one degree or another, even crossing the traditional role boundaries of Coders & Testers to help test, code, prepare environments, research or write documentation.

When a blocking issue is *external* to the team, such as when a required server hasn't yet been delivered or subject matter experts are needed for questions, the Scrum Master should attempt to resolve it. The rest of the team can be engaged as needed to help with external issues.

If a Workitem is blocked completely with no resolution in sight, it should be pulled from the Workline and its status changed from 'In Progress' to 'New' or 'Draft' depending on what is needed before the team can pull it back into the Workline again.

What if a specific Workitem needs to be completed as soon as possible?

There are two ways to address this need. First, the Product Owner should be continually maintaining the order of the Work Backlog such that the items the business considers the most valuable are at or near the top. It is these top items that the team reviews and makes 'Ready' through discussions and feedback and are pulled into the Workline as capacity opens up.

If there a more urgent need for a work item to be completed than what the *standard* process provides, the item's priority can be changed to *expedited* and be fast-tracked through to completion.

E*xpedited* items go to the head of the line for each stage including demonstrations. Any work currently in progress will be moved aside for the priority item.

Borrowing from the example Board for Day 10 of the simulation, the Product Owner has declared item [T] as *expedited* and is displayed in the figure as '*[T]*'. Obviously there's no need for a card in every column, this is done here only to illustrate that the *expedited* item gets to go through the top channel.

WBL		**Code Design** (3)		**Coding** (2)		**Testing** (2)	
		Doing	Done	Doing	Done	Doing	Done
[T] [O] [P] [Q] [r] [s] [t] [u] [v]	Expedited	*[T]*	*[T]*	*[T]*	*[T]*	*[T]*	*[T]*
	Standard	[N]	[L] [M]	[I] [J]		[H] [K]	

As the team works on the item, its transit through the Workline is like any other *standard* priority item with the exception that it always goes first. When individual team members have finished their contribution to the *expedited* item, they can resume work on what was in the Workline previously.

Obviously the appearance of an *expedited* item will be disruptive to the work in progress. Not only will the team have to switch context mentally but they may have to set up new environments or retool for the *expedited* work. Then of course once they are done it may take time to return to the interrupted work.

How are defects handled in Scrumban?

Defects are a natural part of any software development effort and dealing with them is straightforward. When defects associated with current Workitems are found they should be addressed before the item is considered complete and ready for demonstration.

Assuming the defect was found during the Testing stage and it is relatively minor, then the item can stay within the Testing stage as the Coder works closely with the Testing to diagnose the issue and provide updated code.

If the diagnosing and fixing effort is much larger however, such as requiring more than a day to complete, then the Workitem should move back to the Coding stage even if the WIP limits are exceeded. The priority for the Coder should be to address the defect and get the Workline flowing again in the right direction.

Sometimes defects are found in areas of the code not currently part of a Workitem in progress. These should be brought to the attention of the Product Owner who can create a new Workitem representing the defect and prioritize it accordingly in the Work Backlog.

How do Developers decide which Workitems should be pulled from the Workline?

The Product Owner should be maintaining the Work Backlog and keeping the highest value Workitems at the top. When the Developers want to pull new items into the Workline, they should simply be taking the top one.

However, there may be times when the very top item isn't the best choice based on the Developer judgment. Their tools or the environment may not support working on it or it may make more sense

to start with a different Workitem. In any case, the Developers should share their thoughts with the Product Owner so he understands and agrees. There may be certain circumstances where the top Workitem is at the top for a good reason and worth the difficulties faced by the Developers.

Should Workitem sizes be estimated?

The statistics kept on how long Workitems take to transit the Workline should be enough to estimate how long any particular item will take to be completed. For example, if the average number of days is four, then any new item pulled into the Workline is likely to take four days. By the same logic, if there was a set of Workitems that comprised a release, then the time to complete the entire set is four times the number of items in the set.

WBL	Stage 1	Stage 2	Stage 3
[4 days]	**→ Four days to transit the Workline. →**		
[8 days]			
[12 days]			
[16 days]			
[20 days]			

This estimation technique works well when the scopes of the Workitems are somewhat consistent. When they are not, some rough estimates of size, combined with the transit time for those specific sizes can produce a more accurate estimate although it will take longer to gather enough data to be confident with the results. Be mindful that any additional time spent by the team should be worth the effort and if not, abandoned for the simple statistics Kanban provides.