SCRUMBAN SIMULATION

Facilitator guide

Sangeetha Sridhar & Koen Vastmans

this page is intentionally left blank

Scrumban simulation

Table of Contents

Version history

Version	Date	Changes
0.1	2018-10-22	Initial draft version
0.2	2018-10-23	Added more info on the event
0.3	2018-10-27	Logo + pictures added
0.4	2018-10-28	Typos corrected
1.0	2018-11-07	Metrics paragraph added. First final version after processing feedback from reviews

Introduction

This simulation came to existence because we needed a way to teach Scrumban. As agile coaches we had been using the Kanban pizza game (https://www.agile42.com/en/training/kanban-pizza-game/) for a while to teach Kanban, but we needed the mix of both planned and unplanned work. And even though the Pizza game is fun, not every participant is equally eager to learn by coloring, cutting and pasting…

We know that there exist several similar simulations already. The No Estimates game (https://mattphilip.wordpress.com/noestimates-game/) from Matthew Philip is one of the similar games we know and tried in the past. Also Okaloa seems to have a similar simulation, the FlowLab (http://www.okaloa.com/flowlab), but we never tried it. The basics of all these simulations are the same: a Kanban board. Apart from that all other similarities are coincidental. This is not intended as a copy or rip-off of either of these simulations. Each simulation has its own focus. This one focuses on combining planned work with unplanned work. But coincidentally the NoEstimates game happens to have a lot of common ideas and features we were not aware of when we started creating this simulation. At least, when we got to know the NoEstimates game, features like events were not part of it yet.

There is always room for improvement. The best way to discover that, is by doing the simulation. (Mostly) Sangeetha and I did several tryouts before this simulation got into its first really usable and shareable form. Now we have come to a point that others can benefit from our efforts too. We certainly hope you enjoy doing this simulation as much as we enjoyed creating it.

If you have feedback, or improvements you want to share with us, please do so. You can find our contact information on the last page, with the license information.

Sangeetha Sridhar & Koen Vastmans
November 2018

What?

The Scrumban simulation is a game-like way to learn how to deal with both planned and unplanned work. It is context-agnostic: there is no reference whatsoever to development, technology or whatever, making this simulation suitable for both IT-development, IT-operations and non-IT people.

The purpose of the game is to experience in a safe environment what the impact of unplanned events is on your planning and work organization and how to deal with this unplanned work in relation to your overall work.

The board

This simulation can be played with 3 to 6 persons, of which 1 person takes the role of the product owner, the one who decided on priorities.

In a normal work situation the team decides on the Kanban flow. In this simulation however, you don't have to spend time thinking about the best structure of your board. You simply get a predefined board of which the majority is meant for the planned work and a fast lane is reserved for taking care of unplanned work, incidents, for instance, if that is applicable in your context:
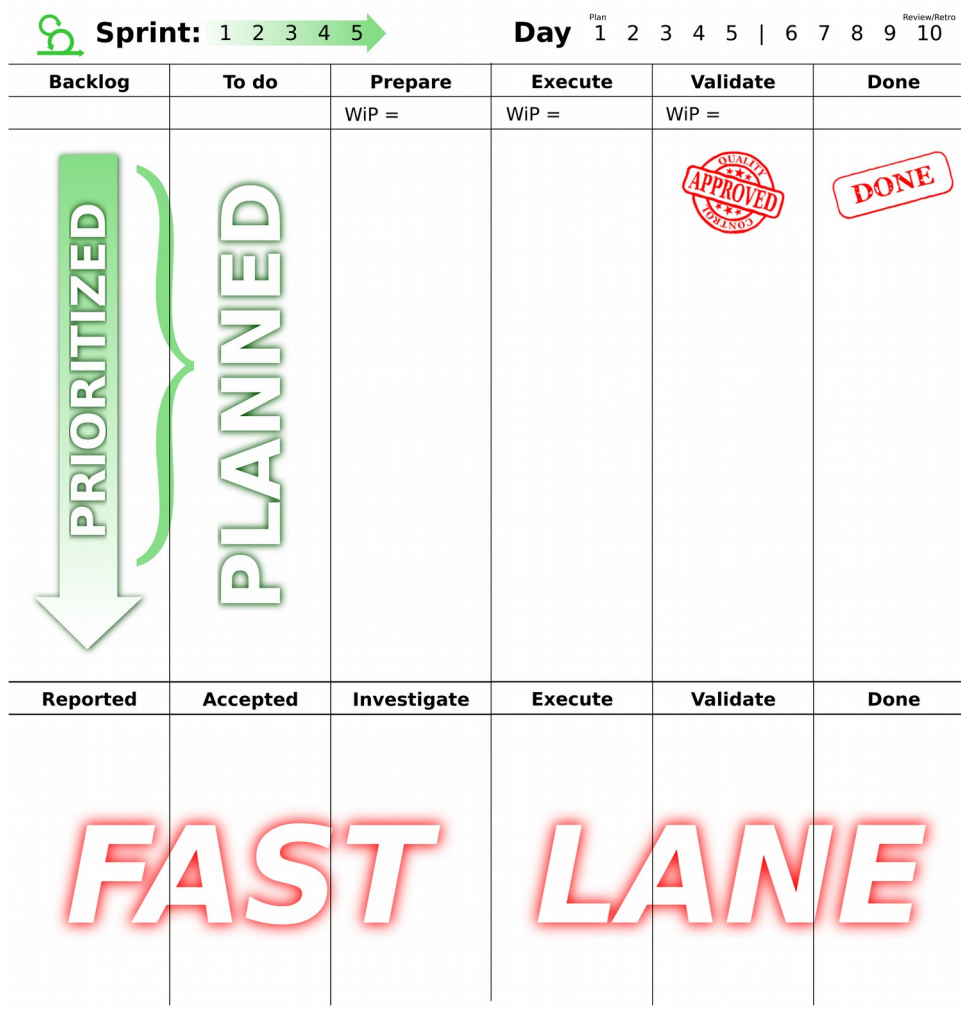
Since this is a Scrumban simulation, you work with sprints, iterations. It is up to you whether you choose to have sprints of 5 or of 10 days. Default – as indicated on the board – is 10 days.

Sprint: 1 2 3 4 5   **Day** ^{Plan} 1 2 3 4 5 | 6 7 8 9 10 ^{Review/Retro}

You will need 2 paper clips here: 1 for indicating the sprint number and 1 for the day in the sprint.

Columns

The structure of the board is pretty straight forward. The planned part consists of the following columns:

Backlog	To do	Prepare	Execute	Validate	Done

- Backlog
  This contains all the product backlog items that are not yet assigned to the current iteration, ordered according to their priority

- To do
  This contains all the product backlog items that are planned for the current iteration and not yet started

- Prepare
  This column contain the backlog items for which preparation work is started or done (since we are dealing with a Kanban board, we don't push these cards to the next column)

- Execute
  This column contains the backlog items that are being or have been implemented

- Validate
  After execution is done, some form of validation or testing needs to be done

- Done

These are the columns of the fast lane:

Reported	Accepted	Investigate	Execute	Validate	Done

- Reported
  All unplanned work starts in this column

- Accepted
  The team will do something for this unplanned work item – which means that is important enough

- Investigate
  This is similar to the Prepare column in the top part of the board, but in case your inplanned work are incidents, this is meant to check what is going on

- Execute
  The same as with planned work

- Validate

  The same as with planned work

- Done

  The same as with planned work

There is no "Hold" column to park items. Instead items that are on hold during a certain stage are marked as an impediment (see Evolving insight) for more information.

Limiting work in progress

There are 2 ways applicable to limit work in progress:

- On the board you need to fill in the WiP (work in progress) limit for Prepare, Execute and Validate statuses.

Prepare	Execute	Validate
WiP =	WiP =	WiP =

  You fill in the WiP limits according to the number of people in the team that can execute tasks in that status. Remember: stop starting, start finishing. Work in progress is considered waste.

- Each team member receives 3 pawns so that they can only execute at most 3 tasks at a time. Each time a team member picks up a task, he/she put one of his/her pawns on the card. More than 1 team member can work on a task. Whenever a task is done, the team member(s) can remove the pawn.

The team

A team consists of at least 2 team members plus a product owner. The product owner is the one that decides about the priorities:

- the order of the backlog item

- which product backlog items should be implemented and which items not (yet)

- what to do with unplanned work

There are 3 types of activities to be done:

- preparation or (in case of unplanned work) investigation

- execution

- validation

The preparation and the validation work can be done by one type of role, whereas the execution is more for another type of role. Similarly to a software development team where functional people take care of the preparation and the validation, whereas the technical people take care of the execution.

This is not exclusive though: technical people can also do preparation and validation work and vice versa, but at a cost: you can do 2 units of work that matches your role (e.g. technical person doing execution work) but only 1 unit of work that does not match your role (e.g. a technical person doing preparation). That is what T-shaped profiles – generalizing specialists – are about.

The work to be done

There are 2 types of work:

- planned work – the product backlog items

Product backlog item			
Bus. value	**150**	MoSCoW	**Must**
Prepare	**3**	○○○○○○○○○○	
Execute	**4**	○○○○○○○○○○	
Validate	**2**	○○○○○○○○○○	
Planned			
Started		Lead time	
Done		Cycle time	

- unplanned work – they come at random

Unplanned work			
Priority	**High**		
Investigate	**1**	○○○○○○○○○○	
Execute	**2**	○○○○○○○○○○	
Validate	**2**	○○○○○○○○○○	
Reported			
Started		Lead time	
Done		Cycle time	

The units of work to be executed are indicated next to the names of the activities. These correspond to the columns on the Kanban board. Notice that these units of work are on the left hand side of the cell. This is because you need to be able to modify this value based on evolving insight. The small circles next to the units of work are meant to indicate how many units have already been spent on that activity.

The cells below the work estimations are meant to write down when – which iteration and which day – the work item was planned or reported (for unplanned work), when someone started working

on the work item and when it was done. This is especially useful to calculate the cycle time and the lead time of the work item.

Playing the game

Planning

As already mentioned, the product owner has to prioritize all work that has to be done. How he/she decides about priorities is entirely his/her own responsibility, but to help, the following criteria can be considered (based on the information on the product backlog cards):

- MoSCoW scoring

- Business value

- Weighted Shortest Job First (WJSF – taking into account the workload)

So the first thing the product owner has to do, is ordering the entire product backlog and put the cards (or at least a first set of most important backlog items) in the Backlog column, ordered by priority.

Next, the team will pick cards from the backlog, based on their capacity and what they think is feasible to implement during the iteration. If they don't strictly follow the priorities of the product owner – e.g. for capacity reasons – they have to agree with the product owner. These cards are then put in the Planned column. And now the fun can really begin...

Iteration execution

First day of the iteration

During the first iteration you already spend half of your time on iteration planning, so you only have half of your time left to work on the first day of the iteration. In practice, this means that you can only color 1 dot in your own specialty. As a consequence, during the first sprint, since there is no execution work yet, people dedicated to execution work cannot spend any time yet. Let's assume that they will spend that half day on reading documentation. 🙂 The following sprints there will probably still be loose ends to do or unplanned work to be finished.

Last day of the iteration

Similarly, at the last day of the iteration, you can't do any implementation work anymore. You spend most of your time on meetings like retrospective and review meeting. In the context of this simulation, there is nothing to review, but you will do a retrospective in order to improve your way of working in the next iteration. No-one of the team will work on backlog items on the day of the retrospective, but if there are any urgent unplanned items to finish, you can continue working on these. After all, the world keeps spinning around…

Other days

Each player can put his/her pawn on a card to execute a task. Working on a task means filling a circle next to the task. To complete a task you have to fill as many circles as indicated in the workload cell next to it.

Don't forget: you can work on a task that is not your specialty but only on half of the capacity as working on a task of your specialty. So a technical person can do 2 units of execution work, but only 1 unit of preparation/investigation or validation work.

Four eyes principle

A team member that took care of the execution of a product backlog item or an unplanned work item is not allowed to do the validation of the work him-/herself. Someone else in the team has to do this. This is the four eyes principle.

Evolving insight
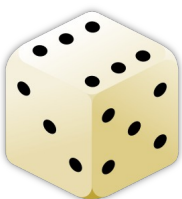
At the end of his/her round, when a task is done, a team member will not push the card to the next stage. That is against the principles of Kanban: you pull cards. Even if the task is done, you have to roll the dice. According to the value, you have to do the following:

1. add 1 unit of work to the last task you have worked on – even though it was finished

2. no action – lucky you

3. no action – lucky you

4. take an event card and act accordingly

5. block the last card you have been working on
   You do this by putting a red pawn on the card

6. you can unblock any blocked card on the board

Events

The stack of event cards is placed face down. When you throw a 4 with the dice, you have to take an event card. Some events only have impact on you as an individual, other events impact the entire team. This can either be an impact on capacity – you or the team are confronted with an unavailability – or on priority – someone decided that some backlog items are more important than others. This is not only in a negative way, but also in a positive way. Some examples of events:

- You are ill and are absent for the rest of the week

- Your Scrum master is ill and you need to take over his role for the rest of the week

- You can finally go to that training you wanted to attend

- Another team is helping you out with your unplanned work

- Management intervention makes that all impediments are cleared

Some events have immediate impact – or let's say as of the next day (because you pick up the card at the end of your working day) – whereas some events will have an impact as of the next iteration.

Unplanned work

At the end of the day, someone of the team rolls this dice:

A 0 means you're lucky: no unplanned work pops up. Otherwise you have to take the number of unplanned work cards from the stack as indicated 10 the dice:
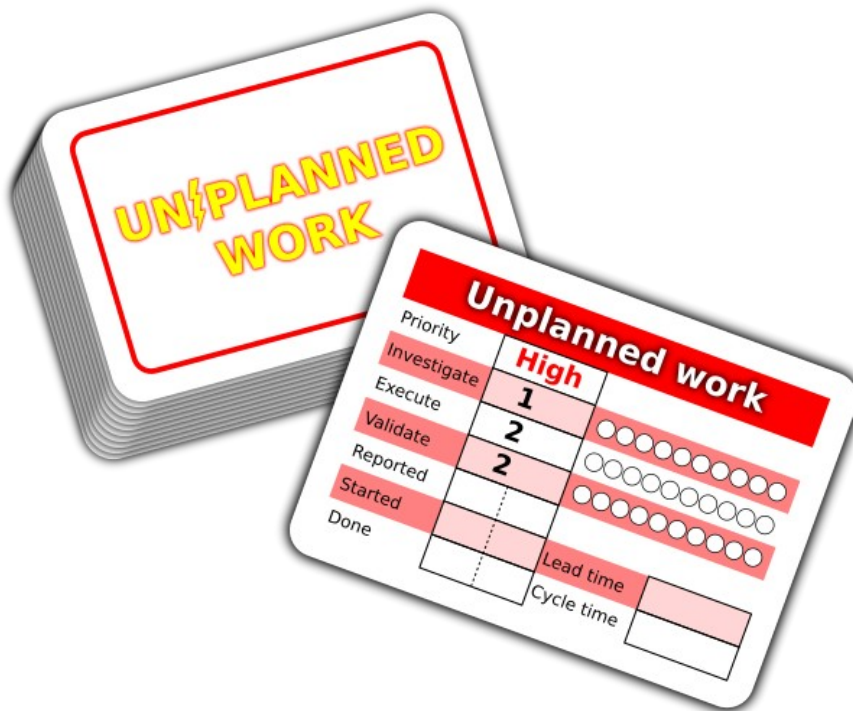
Similar to the events, the stack of unplanned work cards is placed face down and you just pick cards from the top of the stack.

Similarly to determining the priorities of the product backlog item, the product owner has to decide about priorities of unplanned work. He/she can take this decision based on:

- Priority
- Workload

Unplanned work can e.g. be:

- Executed ASAP
- Planned in a next iteration
- Postponed for a later stage

This all depends on the impact, both on the planned work and on the operational work.

Retrospective

At the end of each iteration the team does a retrospective. They will evaluate the way they worked the past iteration, what went well, what could be done better. Some things to help you:

- Were you able to finish what you had foreseen for this iteration? Didn't you over-plan? Did you create enough value?

- How was the flow of work? Was there enough work prepared for execution ? Did the team members assigned for execution have to assist in preparing items? What was the impact on efficiency?

- Did you take the right decisions on unplanned work?

- Did you respect the WiP limits?

- For more advanced teams:

  - How was your cycle time and lead time? What caused the longest cycle time and lead time?

  - How does your cumulative flow diagram look like? Aren't there too many open items?

Improve with small steps, don't try to change everything at once, but focus on the improvement has gives the biggest immediate impact in the next iteration.

The next iteration

After your retrospective you can start a new iteration. This is the right time to re-evaluate the priorities of the remaining backlog items, see what to do with unplanned work that wasn't accepted yet and introduce the improvement(s) you agreed during the retrospective.

Metrics

If you are more advanced in Kanban and Scrumban, you can keep track of the following metrics:

- Value creation

- Cumulative flow diagram

- Cycle time and lead time calculation

Value creation

The value creation is the simplest to keep track of. There is a separate document template available for this:

Value creation

Day	Value	Cumulative value
1		
2		
3		
4		
5		
6		
7		
...		

The usage of this document is very simple: day after day you write down how many value has been created that day and how much in total. You could also draw a line to indicate the end of the iteration and see how much value has been created during that iteration.

Cumulative flow diagram

The cumulative flow diagram gives a day to day visualization of the quantity of work at each stage. It is called cumulative, because the graphs per stage are stacked. In an ideal situation, the graph for planned work timeboxed in an iteration looks something like this:
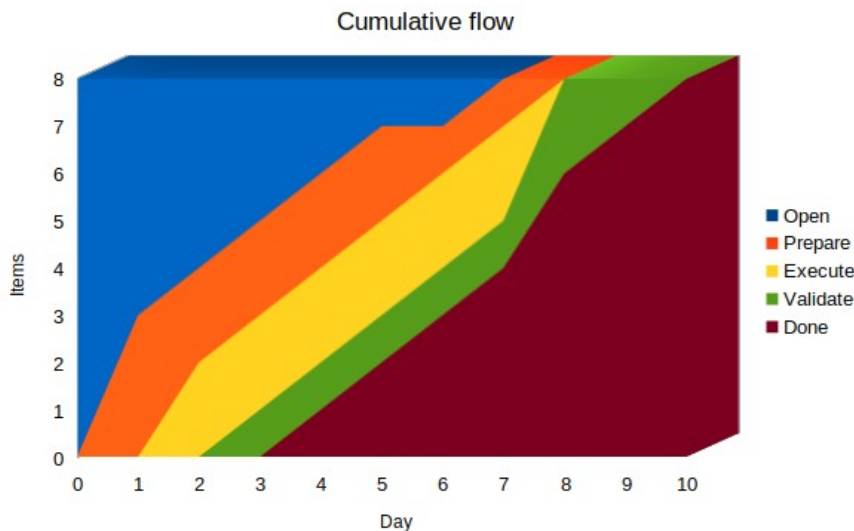
In this fictitious example, at the start of the iteration all items are open. After 1 day of working you see 3 item being pulled, being prepared. Typically at the start of an iteration the focus is on getting execution started as quickly as possible, so in the beginning you pull more work into preparation. Then the execution starts at day 2. On day 3 people can start validating items. And as of day 4 the first item gets validated and done. The team continues pulling items. At day 7 all the preparation work is done. As of day 8 the focus is entirely on validating. And on day 10 all the scope of the iteration is properly done.

Off course in reality this graph will probably be more spiky, especially in a Scrumban context, where also unplanned items need to be pulled in. And most likely not all items will be done at the end of the iteration.

There is also a separate template available for the cumulative flow diagram, if you want to visualize the work at all stages for your team.

Cycle time versus lead time

Lead time is the amount of time it takes between the request and the final delivery of an item. Cycle time is the amount of time it takes from start of work until it is done.

On the cards of both the product backlog items and the unplanned work items there are boxes to write down:

Planned		
Started		Lead time
Done		Cycle time

Planned work

Reported		
Started		Lead time
Done		Cycle time

Unplanned work

- iteration number and day when the item got planned/reported (in case of unplanned work)

- iteration number and day when the work started

- iteration number and day when the item was done

These numbers can then be used to calculate the cycle time and the lead time. The difference between cycle time and lead time tells you something about efficiency of the flow: the closer the cycle time is to the lead time, the better the flow. You can also compare these values with the total workload to get an idea about your cost of delay.

How can I win this?

That is a question I was asked already several times… You know, if you play Flight Simulator, you "win" if you can properly take off and land the plane without crashing. This is a simulation: you win if you can improve your planning and take the right decisions to create value and handle unplanned work.

But if you really want to introduce a game aspect? Then you would need at least 2 teams and keep track of who creates the most business value and/or whose cumulative flow diagram show the best balance between open, in progress and done items.

Needed materials

The board, the cards and all the reference material are available for download. The board consists of 2 sheets of size A3 for the planned work and 2 sheets of size A4 for the fast lane. For durability and re-usability it is better to laminate these sheets and stick them together with tape in such a way that you can still fold the board to a single size A3 page.

The cards are measured 5 cm high by 8 cm wide. Because you need to mark the progress dots and correct the estimates if needed, you need to laminate these as well. Credit card size laminating pouches are best fit for that (approximate size 5,4 cm by 8,6 cm).

You need pawns, any color you like, 3 per team member, to indicate what they are working on, plus red pawns for marking the blocked items. You better foresee enough red pawns, e.g. 10 per team.

2 paperclips will do to mark the iteration and the day of the iteration.

A regular dice is needed for the evolving insight and the events. To introduce the unplanned work, you will need an empty dice so that you can add the values 0, 1 and 2 yourself. Or you could go for a digital version: download a dice app. Configurable dice usually aren't for free though...

To write on the cards (re-estimating and indicating progress) and the board (for indicating the WiP limits) you need some thin whiteboard markers or non-permanent markers.

License

Thank you!