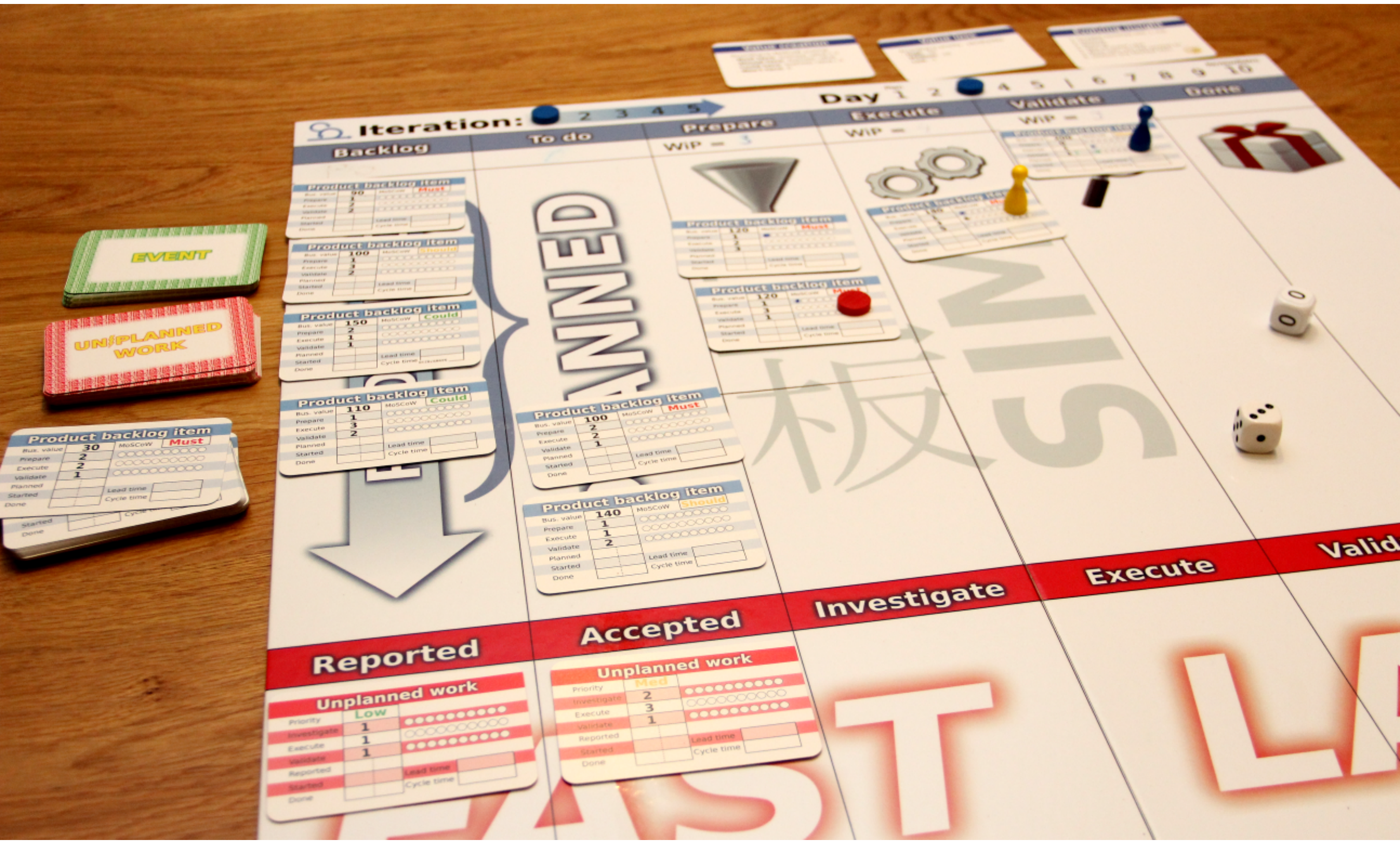
SCRUMBAN SIMULATION

Facilitator guide



SimuLearn

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
1.1	2018-12-02	New image of the value creation sheet Introducing penalties for not completing unplanned work
1.2	2019-02-11	MoSCoW score taken into account for value creation (as a multiplier) Penalty for not completing unplanned work modified
1.2.1	2019-02-27	Small correction in the reference to Okaloa Flowlab
1.2.2	2019-05-21	Higher penalty for not finishing unplanned work
1.3	2019-12-11	Typos corrected, small changes to wordings, changes to the visuals according to the box version (ready to use simulation kit) and company branding
1.4	2020-05-14	Difference between external blockers and blockers you can solve within the team (suggestion of iLean). Better explanation about combining or separating activities within 1 role. Role of the Scrum master and the product owner a bit more elaborated.
1.4.1	2020-06-23	Minor change in how team can unblock issues themselves.

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 several similar simulations already exist. 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. Okaloa has a suite of simulations, the Okaloa Flowlab (http://www.okaloa.com/flowlab). We never tried the Okaloa Flowlab ourselves. The basics of all these simulations are the same: a Kanban board and work that needs to be done. 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. The aim of the Okaloa Flowlab is not to teach a certain process, but to teach an agile mindset instead. Therefore it is a suite of simulations that help you to grow from team level agility to organization wide strategic agility.

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 contact information.

Sangeetha Sridhar & Koen Vastmans
November 2018 – May 2020

What?

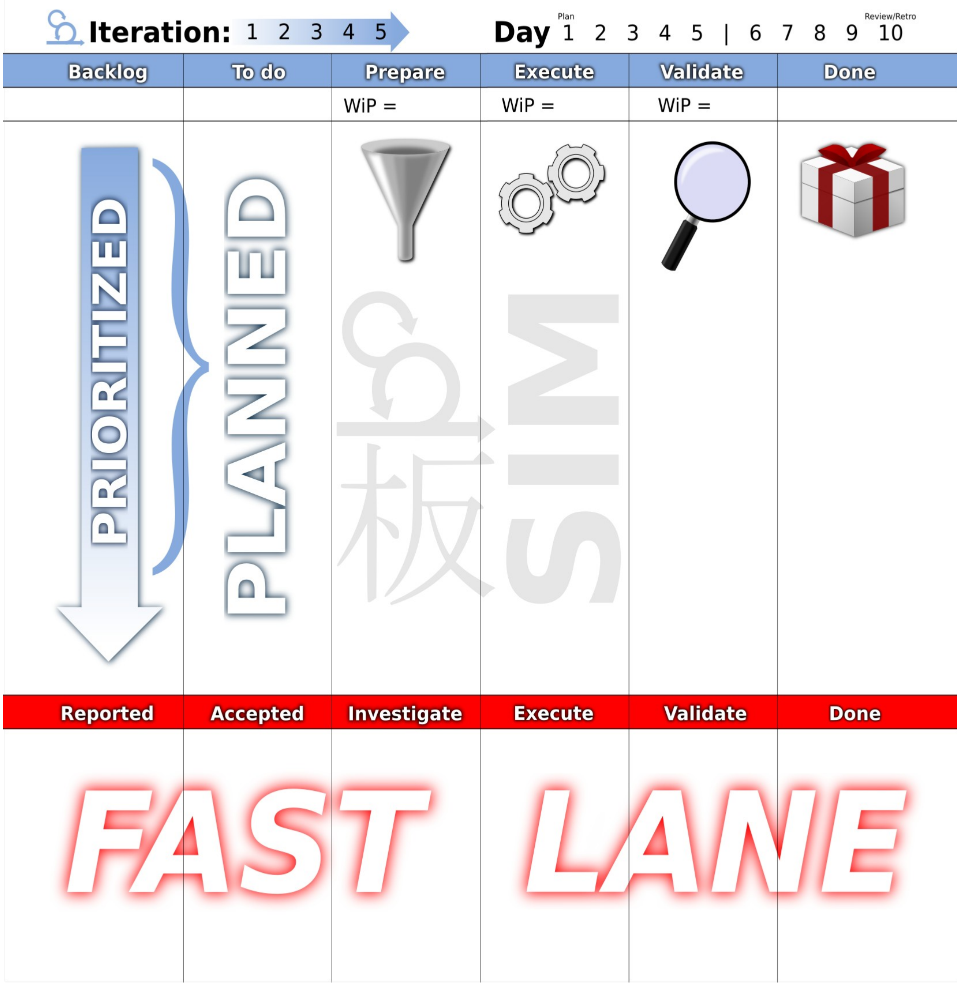
The Scrumban simulation is a game-like way to learn how to deal with both planned and unplanned work. It is context-agnostic: there is no reference whatsoever to development, technology or whatever, making this simulation suitable for both IT-development, IT-operations and non-IT people.

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.

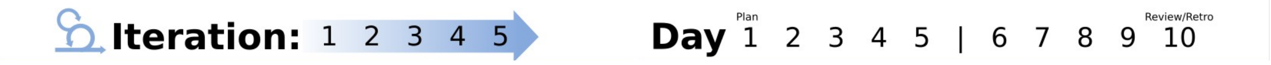
The board

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.



The 2 blue disks are used for marking the iteration number and the day of the iteration.

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 unplanned 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 Scrum master and 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)
- more specific: the sprint goals (which backlog items should absolutely be delivered by the end of the iteration)
- what to do with unplanned work (except for high priority items)

In this simulation the Scrum master has a minor role: he/she receives a card – Scrum master super hero, with which he/she can remove all impediments (all blocked items on the board) at once. Attention: this card can only be used once!

Both the Scrum master and the product owner can combine their role with working on backlog items and unplanned work (they can choose a specialty), but only at half the capacity, because they spend the other half of their time on their Scrum role.

There are 3 types of activities to be done:

- preparation or (in case of unplanned work) investigation
- execution

- validation

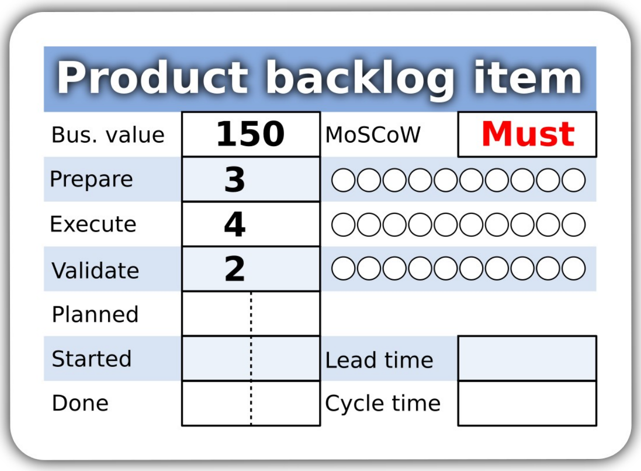
It is up to you to decide how you divide the roles within the team. You could choose for separate roles for each activity – 1 role doing only preparation, 1 role doing only execution and 1 role only doing validation – or you could choose to combine 2 (and only 2, not all 3!) activities within 1 role: e.g. combine preparation and execution or preparation and validation. Watch out if you combine execution and validation: you have to respect the four eyes principle, which means that you cannot validate the work you have just executed (see further)!

Choosing a role, be it single or combined, is not exclusive though: people whose specialty is execution, 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. an executing person doing execution work) but only 1 unit of work that does not match your role (e.g. an executing person doing preparation). That is what T-shaped profiles – generalizing specialists – are about.

The work to be done

There are 2 types of work:

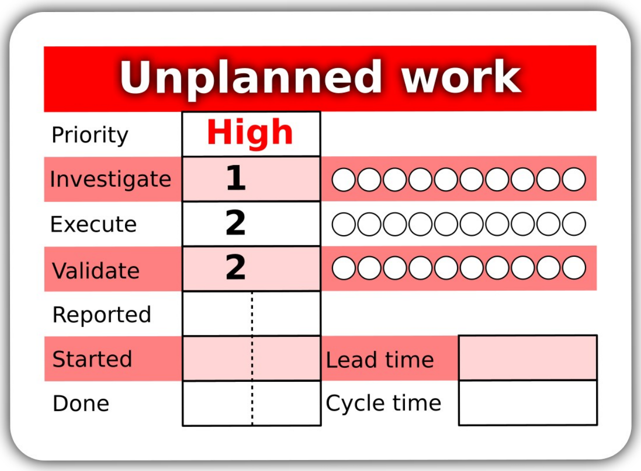
- planned work – the product backlog items



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.

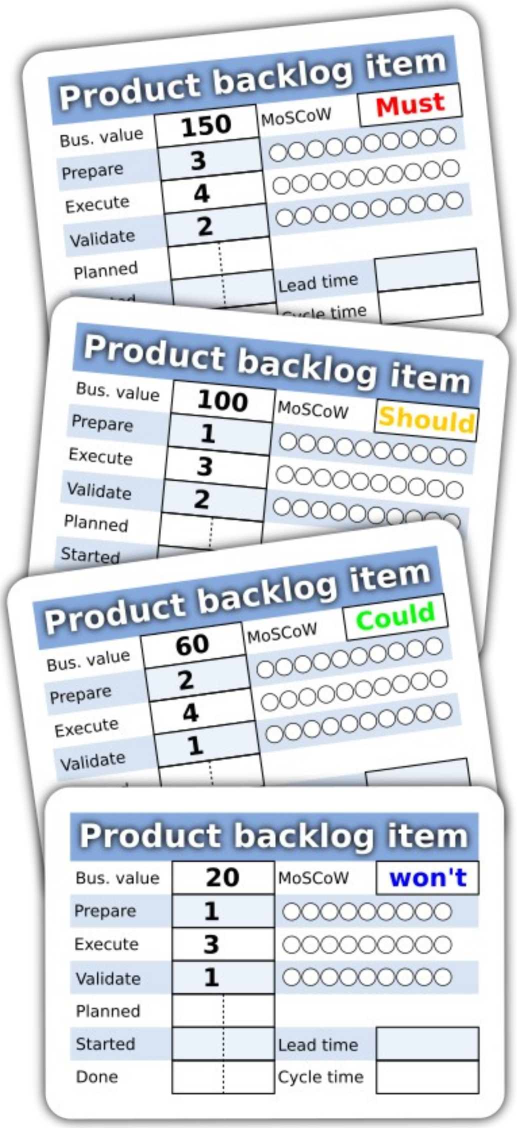
The cells below the work estimations are meant to write down when – which iteration and which day – the work item was planned or reported (for unplanned work), when someone started working on the work item and when it was done. This is especially useful to calculate the cycle time and the lead time of 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: importance
- Business value: biggest benefit
- Workload: fastest benefit
- Or a combination (like WJSF – Weighted Shortest Job First)



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

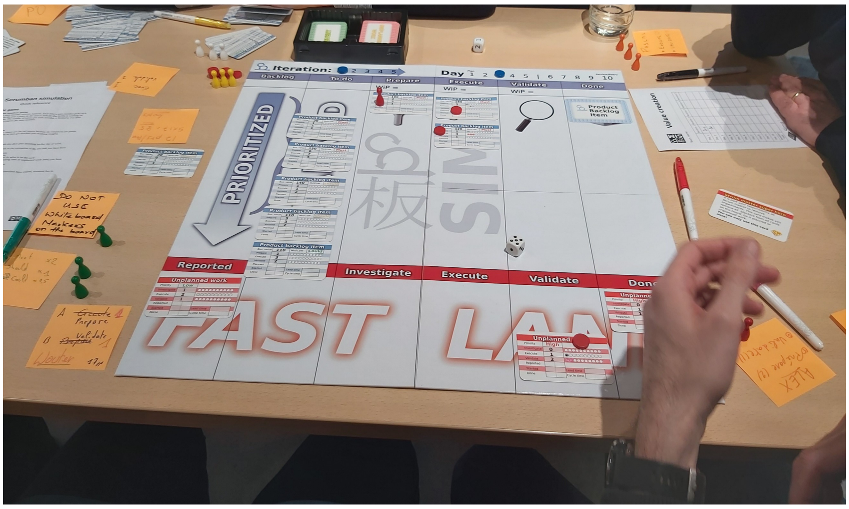
During the first iteration you already spend half of your time on iteration planning, so you only have half of your time left to work on the first day of the iteration. In practice, this means that you can only color 1 dot in your own specialty. As a consequence, during the first sprint, since there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

Alternatively, you could start the simulation by introducing unplanned work, before you pull the first product backlog item from the To do column. Role the special die to see how many unplanned work cards you bring to the board to start with (see Unplanned work).

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 last day because you don't want to mess up your demo, 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 validate his/her own work. 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 because you can still have new insights (more work, blocked). 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 disk on the card

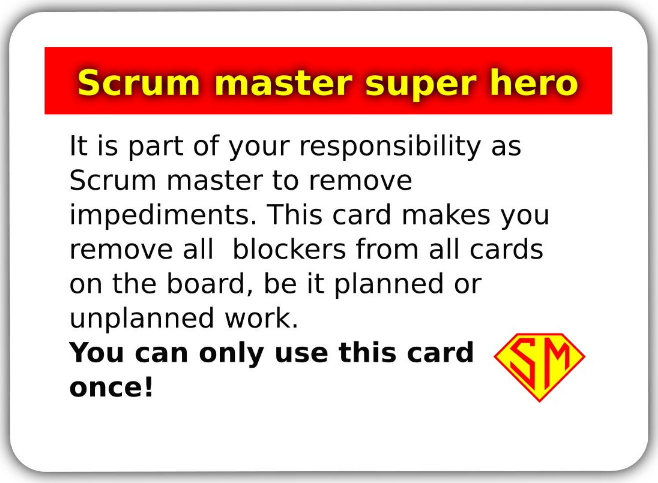
6. you can unblock any blocked card on the board

Blocked cards – can the team solve it or not?

When someone throws a 5 which means the last item he/she worked on gets blocked, he/she also needs to throw the special die, with the values 0, 1 and 2. Primarily this is die is used for bringing unplanned work to the board. But in this case, this die will determine whether or not the team can solve the problem themselves.

- If you throw 0, this means you have 0 impact to solve the problem. The item gets blocked because of external circumstances (you wait for a decision, another team, …). You remove the pawns from the card and put a red disk on it, to mark it as blocked.
  The item remains blocked until it is unblocked (when someone throws a 6 or the Scrum master intervenes – see further).
- Any other value means that someone in the team can help you unblock the item. This means: if you have a card blocked in the Prepare stage, someone else from the team, preferably with the Prepare specialty, can help you unblock the issue. You put a red disk on the card to mark it as blocked, but you don't need to remove the pawns.
  How do you do this? The next day you and the other colleague will spend the number of units of work as indicated by the die (1 or 2) to remove the impediment. The red disk can then be removed.

Scrum master super hero



Scrum master super hero

It is part of your responsibility as Scrum master to remove impediments. This card makes you remove all blockers from all cards on the board, be it planned or unplanned work.
You can only use this card once!

It can happen that several cards on the board get blocked. If you are in the unlucky situation that no-one manages to throw 6 (to unlock any item), this can really stop you from progressing and creating value.

Removing impediments is one of the responsibilities of the Scrum master. He/she will have to do anything to make sure the team can go on delivering value. To achieve this, the Scrum master super hero is introduced.

This card allows the Scrum master to remove all impediments at once.

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

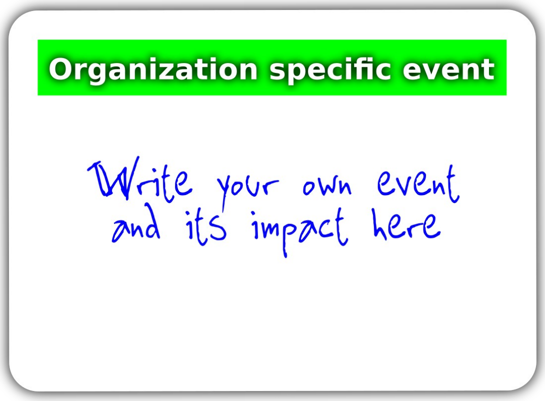
Event

Change of priorities. The entire team has to do a new planning session and start the iteration over again.

The entire team loses half a day + start a new iteration

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.

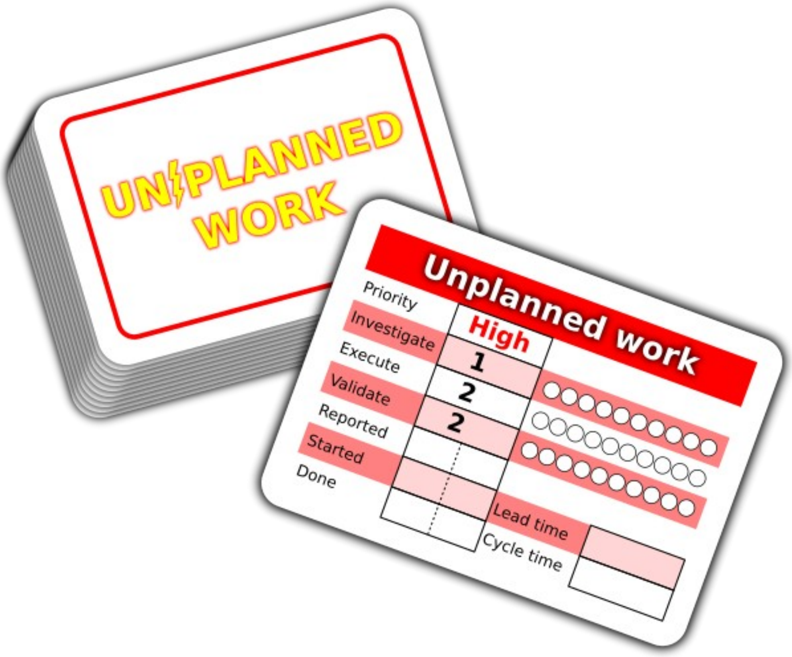
On request a number of empty event cards were added, so that organizations can introduce their own events in the simulation:


Organization specific event
Write your own event and its impact here

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 on 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, except for high priority work, which has to be executed immediately. He/she can take this decision based on:

- Priority
- Workload

Unplanned work can e.g. be:

- Executed ASAP
  This is mandatory for high priority work!
- 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. But postponing unplanned work does not come without a penalty! Each day an unplanned work item did not finish, you have to reduce your value creation (see further in Metrics), according to the priority of the unplanned work item.

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/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 that has the biggest immediate impact in the next iteration.

The next iteration

Depending on how much time you can spend to do this simulation, you can start a new iteration after your retrospective. 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.

It could be that you only have time to do a single iteration, for now. At a later moment you can do the simulation again, start from scratch, remember what your improvements of the retrospective were and try something new, like calculating the cycle time and lead time or plotting the cumulative flow diagram.

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 – value loss

The value creation is the simplest to keep track of. In fact this is more like the game and fun element of the simulation, not really a metric from which you learn something (see How can I win this?). The value creation is the business value multiplied by a factor based on the MoSCoW score of the product backlog item:

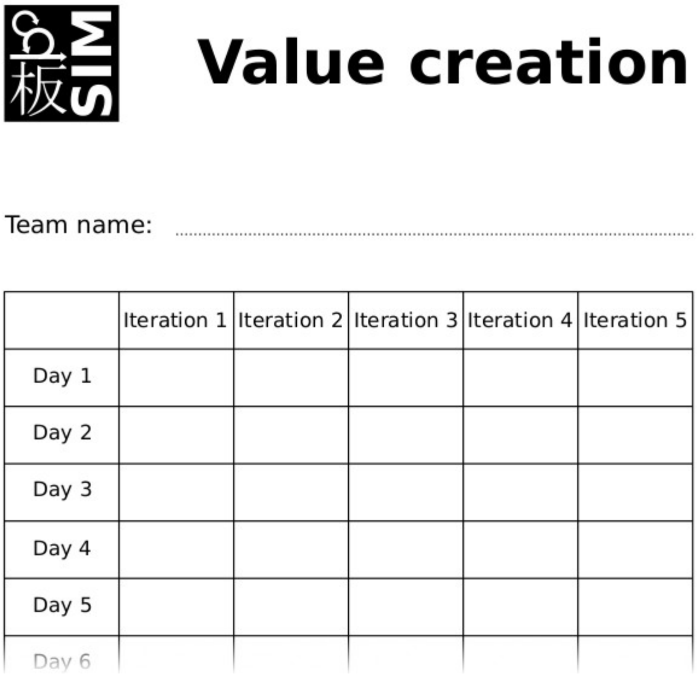
- Must have: business value x 2
- Should have: business value x 1
- Could have: business value x 0,5
- Won’t have: business value = 0

You can create value, by finishing backlog items, but you can also loose value if you don’t finish unplanned work in time.

For each day an unplanned work card is not finished, you loose value according to the priority of the work:

- -50 per day for high priority work
- -10 per day for medium priority work
- -1 per day for low priority work

There is a separate document template available for the business value calculation. The usage of this document is very simple: day after day you write down how many value has been created or lost (for unplanned work), per iteration and a total per iteration. You can give your team a name, to make the fun complete:



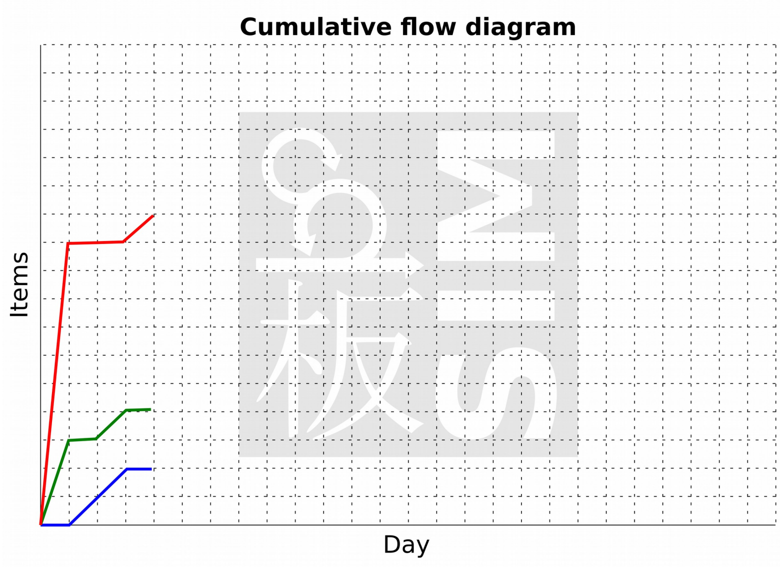
Value creation

Team name: ....

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Day 1					
Day 2					
Day 3					
Day 4					
Day 5					
Day 6					

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. It represents the flow of work. Ideally the lines of planned, in progress and done are close to each other. That means that you actually finish work, instead of planning new items (stop starting, start finishing).



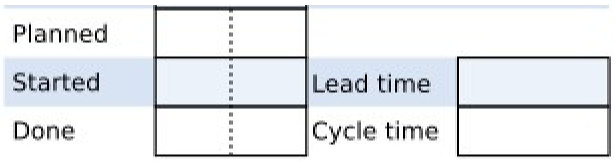
In this example the red line represents the work to do, the green line the work in progress and the blue line the done work. At day 1 a lot of work is to do and 3 items are in progress. At day 2 the first item is done. Meanwhile, at day 4 some more work to do is added, most likely unplanned work got accepted. The example only shows 3 stages. If you want to, you can also visualize the prepare and validate stage, to get a more complete view on the flow of work.

There is a template (empty chart) available for the cumulative flow diagram, if you want to visualize the work at all stages for your team.

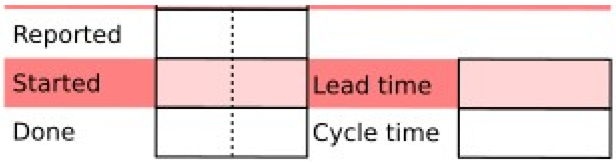
Cycle time versus lead time

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 work



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.

Contact information

The Scrumban simulation is a realization of SimuLearn. Sangeetha Sridhar and Koen Vastmans created this simulation based on an initial pre-mature idea of Dave Janssens and inspired by several other similar simulations. You can find more information about SimuLearn and their offering on this web site:

https://www.simu-learn.net

You can contact us via the contact page of our web site:

https://www.simu-learn.net/contact

Thank you!

A special thank you goes to the following people who gave suggestions to improve the simulation:

- Chris Verlinden, Marie Jacqmin and Karen De Boeck of Adjugo:
  First ever meetup with a very early prototype of the Scrumban simulation
- Sheryar Malik & Andy Verbunt:
  First real usable version of the Scrumban simulation
  Suggestion about adding events – things that impact the planning or capacity of the team, not only in a negative but also in a positive way
- Dominique Jacobs:
  Suggestions like the right penalty for not completing unplanned work, introducing unplanned work at the start of the simulation instead of at the end of day 1
- Frederik Vannieuwenhuyse, Vincent Vanderheeren and Kasia Sikora of iLean:
  Suggestions like impediments the team can remove themselves, sprint goals

We would also like to thank the people who reviewed this document and gave feedback:

- Matthew Philip (#NoEstimates game)
- Madhavan Elango
- Davy Benoot
- Nele Van Beveren
- Dominique Jacobs
- Patrick Steyaert (Okaloa Flowlab)

And off course to all the people who tried out this simulation on several occasions (meetups, workshops, training sessions) and gave their feedback.