Aalto University
School of Science

How to be a Scrum Master in the Scrum LEGO Simulation

Ville T. Heikkilä, Maria Paasivaara, Jari Vanhanen, Ferrix Hovi
Towo Toivola

Based on the ”Scrum Simulation Compact Guide for Scrum Masters”, 2017.

Scrum Master Role in the Simulation (1/3)

- **Your main task is to get the team to**
  - work well together
  - learn how to use Scrum in a way that best benefits them

- **You are a full-time Scrum Master, you coach the team**
  - You do not work with the system or otherwise as a team member.

- **You are a process expert – not a technical expert**
  - You do not give your team instructions on how to “code” (build legos).
  - You give answers regarding the process

Scrum Master Role in the Simulation (2/3)

- **Team needs to learn solve problems together - do not solve problems for them, but take care that the team gets the problems solved. However, you are not their manager.**

- **Asks questions, for example**
  - Does your solution work together?
  - Are you satisfied with PO's input or do you need more clear priorities? Do you need more clarity?
  - How certain are you that you can deliver all you committed to?
  - Is there something you would need to do that should be added to the backlog?
  - Would you need to prepare some questions for the PO before he arrives?

Scrum Master Role in the Simulation (3/3)

- **The simulation is identical to the one played at Aalto since 2014**
  - some of you have played it earlier
- **Do not reveal too early what you learned last time**
  - Do not spoil the discovery and learning opportunity from anyone
- **The simulation is not about the correct end result - but the process getting there**

You and the team will constantly be under huge time pressure

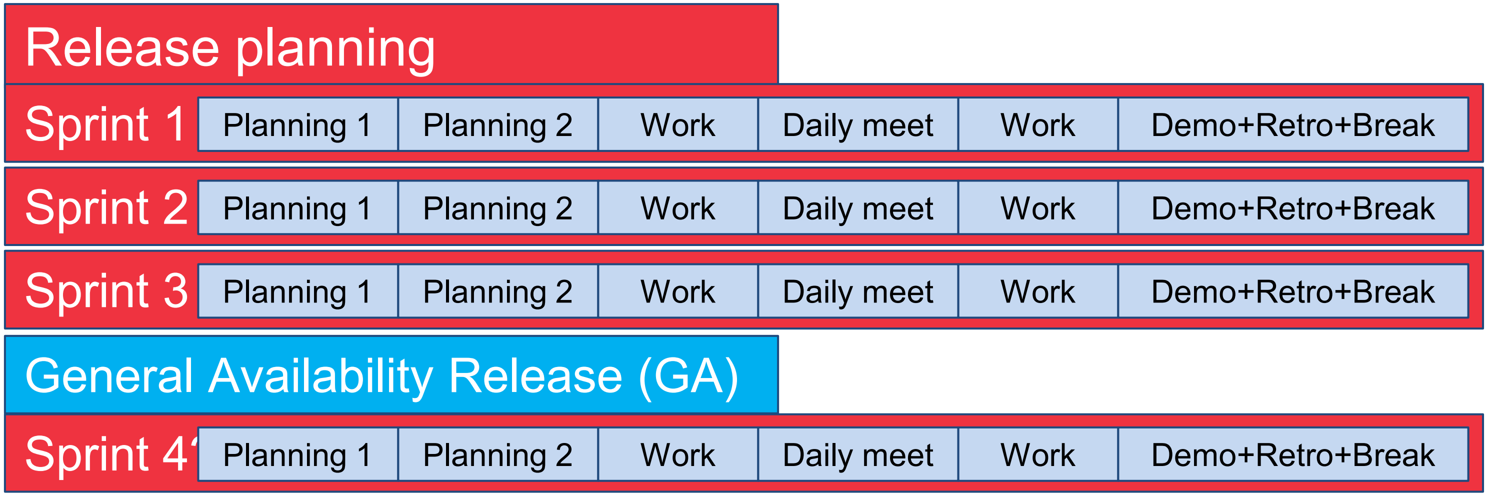
..like in many real projects.

You will have a constant schedule of events with only minutes to get them done! For this reason, the ScrumMaster's #1 value is:

"Relentless focusing of the team's attention on the currently most important question"

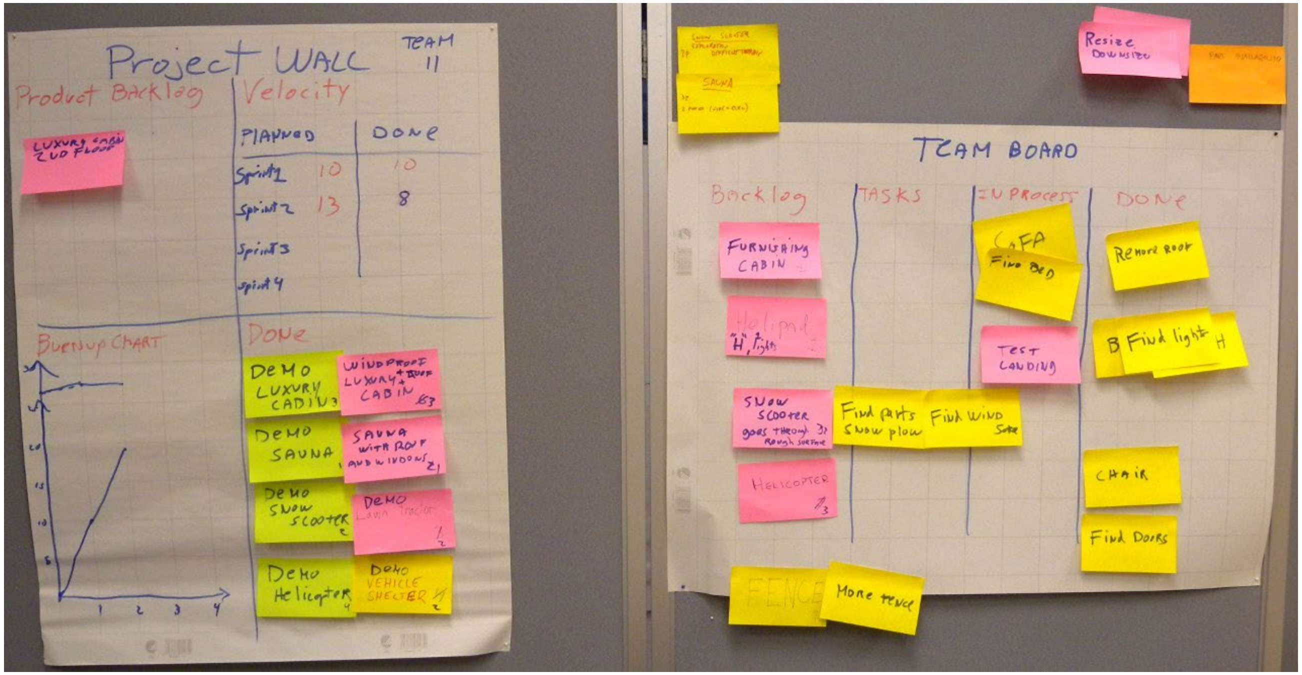
..this is so also in the real world of the industry. "What is the currently most important question", you ask? Knowing that is the ScrumMaster's professional skill.

This is how it goes..



Release planning						
Sprint 1	Planning 1	Planning 2	Work	Daily meet	Work	Demo+Retro+Break
Sprint 2	Planning 1	Planning 2	Work	Daily meet	Work	Demo+Retro+Break
Sprint 3	Planning 1	Planning 2	Work	Daily meet	Work	Demo+Retro+Break
General Availability Release (GA)						
Sprint 4	Planning 1	Planning 2	Work	Daily meet	Work	Demo+Retro+Break

The Scrum Boards



The Product Owner mostly cares about the Project Wall.

Team decides how they use the Team Board (a.k.a. Sprint Backlog)

Updating the Project Wall and the Team Board

- **The Project Wall is updated at least in Release Planning, Sprint Planning Part I and Sprint Review.**
  - i.e. when the PO is present in the team
  - Really it makes sense to keep it tidy and up to date at all times

- **The team decides on when and how the Team Board is updated.**
  - Good practices
    - *The Team Board is always up-to-date in the end of the ”day”*
    - *Every team member updates the board, making their understanding visible*
- **While the team is working, you will have time to put the boards and graphs in order**

Start of simulation: Release planning

- **Main objective**
  - Gain overall understanding of what the release is about and is likely to contain.
- **Key points**
  - Discuss **main use-cases** for each Product Backlog item with the Product Owner. ("***What is use-case #1 for that feature?***")
  - Estimate the Product Backlog items in some way.
  - *Obtain some priorities, at least which items can be sacrificed to manage risk, if that becomes necessary.*
  - Update project wall.
- **This is not exact science, be fast and efficient.**

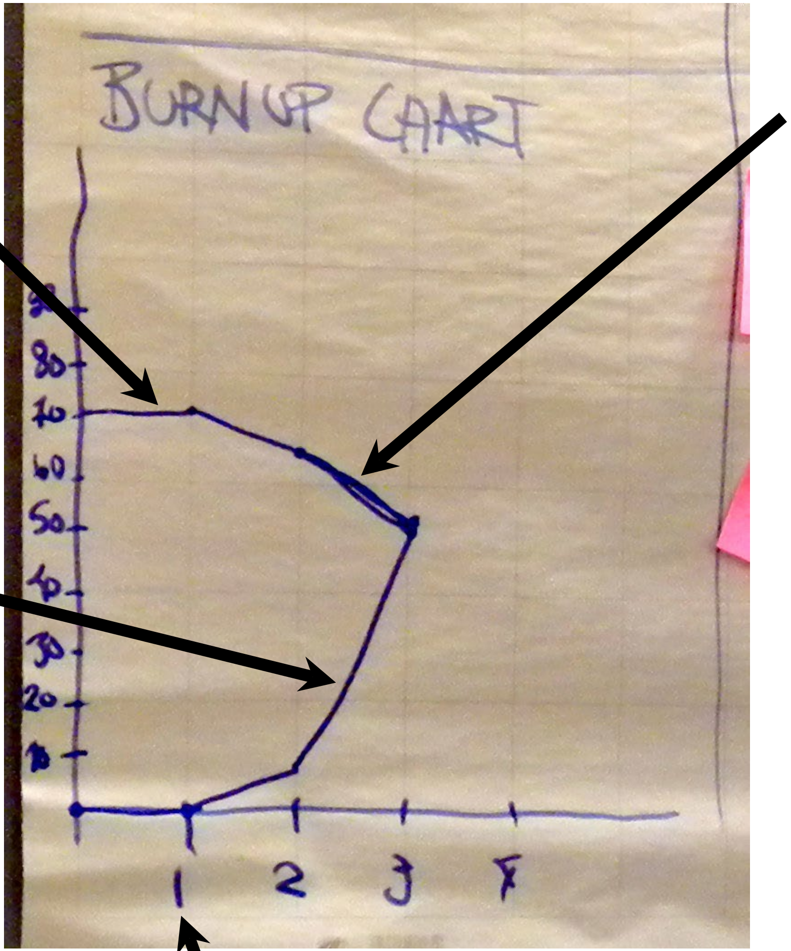
Definition of Done (DoD)

- In real Scrum projects, the team and PO decide DoD together.
- In this game, DoD is ”PO accepts the backlog item as Done”.
- A backlog item can be moved to the ”Done” column of the Project Wall only after it fulfills the DoD.

What is a burn-up chart?

The scope, i.e. the sum of the estimates of all backlog items in the product backlog.

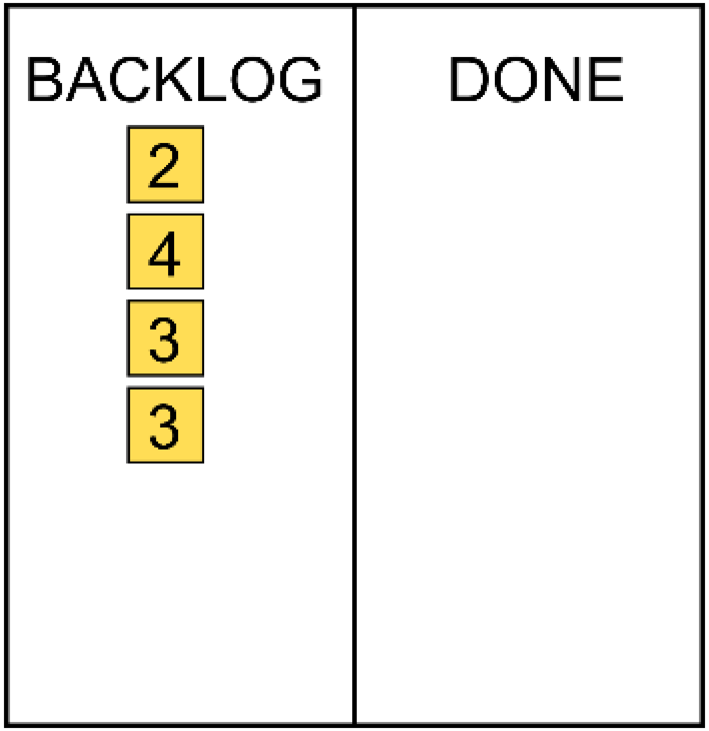
The burn-up, i.e. the sum of the estimates of all Done product backlog items.



The scope goes down if backlog items are removed from the project scope or re-estimated smaller.

The scope goes up if a new backlog item is added or an existing one is re-estimated bigger (not shown).

-burn-up value updated after Sprint 1 Review
-scope value updated after Sprint 2 Planning



BACKLOG

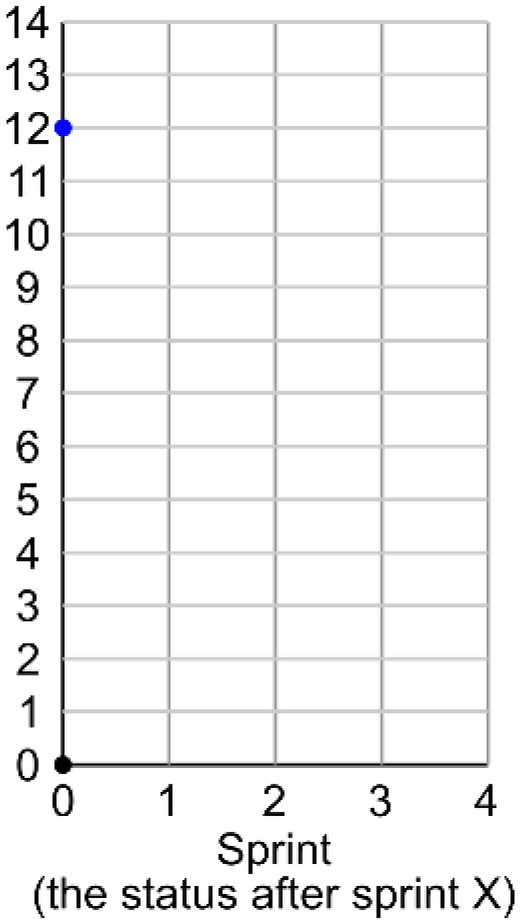
2

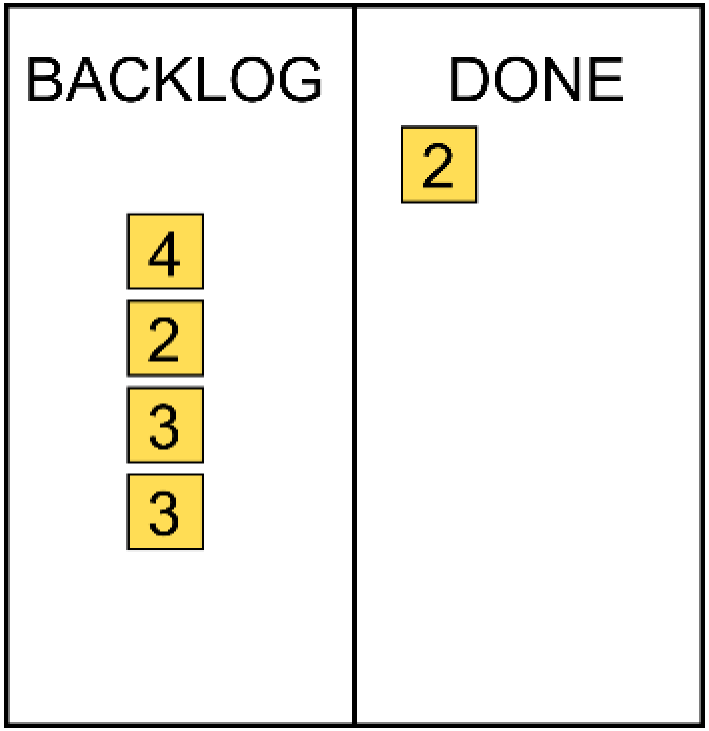
4

3

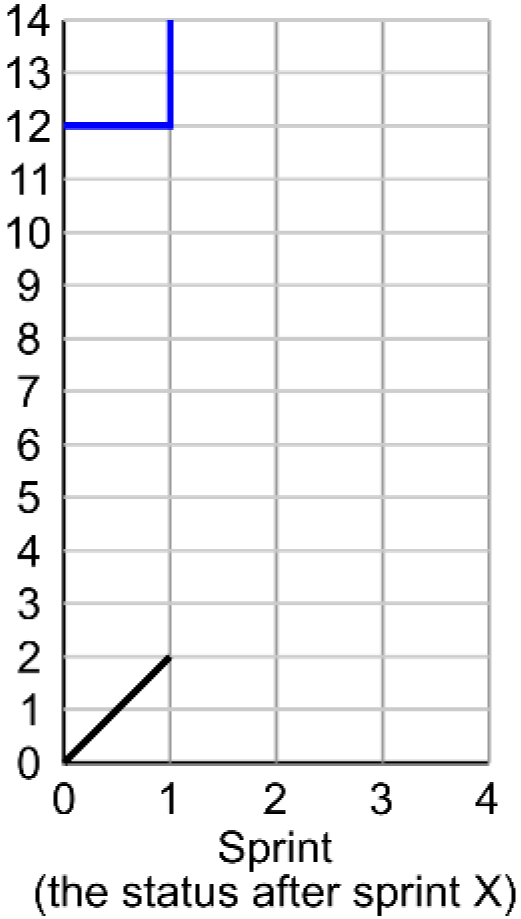
3

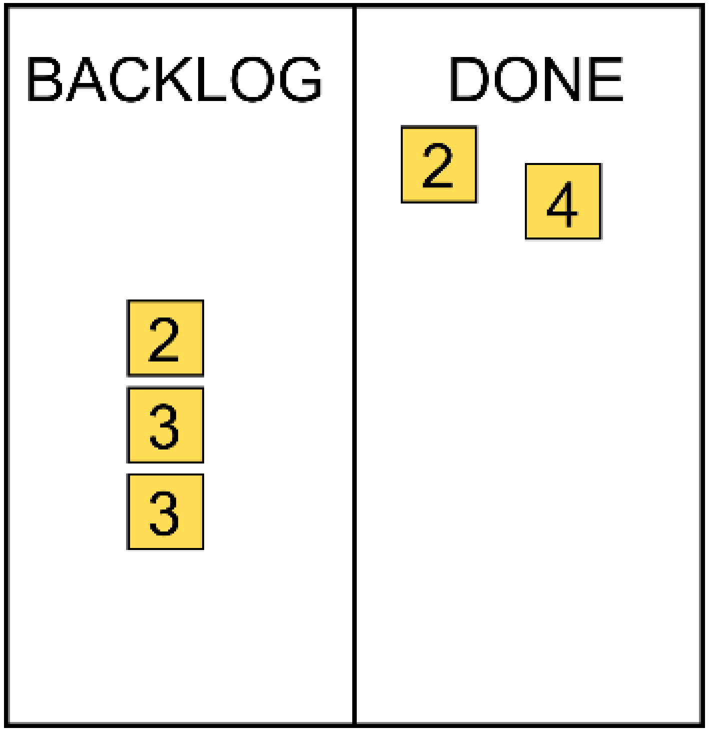
DONE





BACKLOG	DONE
4	2
2	
3	
3	





BACKLOG

2

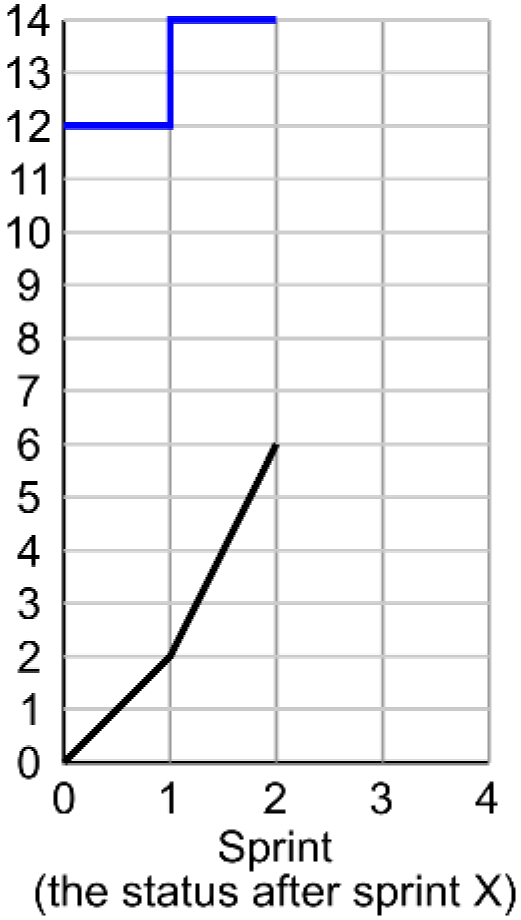
3

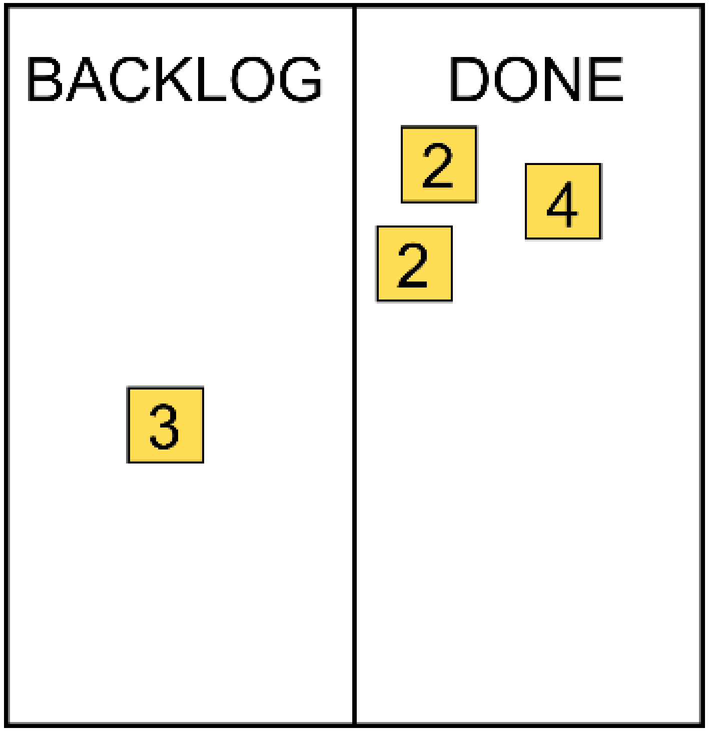
3

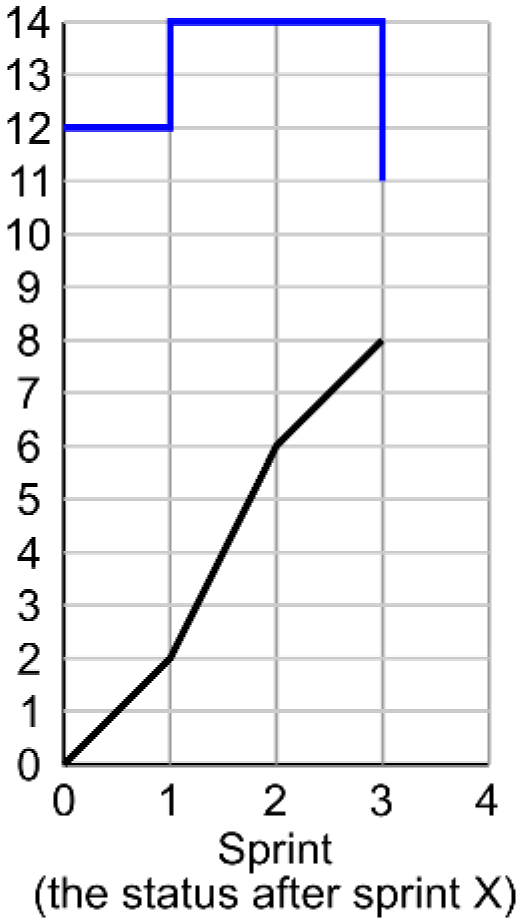
DONE

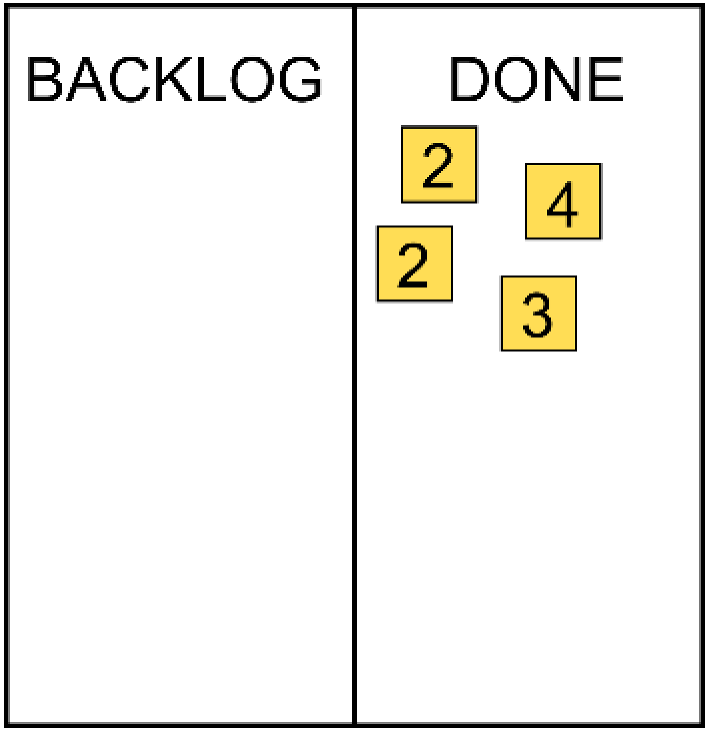
2

4




BACKLOG
DONE
2
4
2
3


14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
Sprint
(the status after sprint X)



BACKLOG

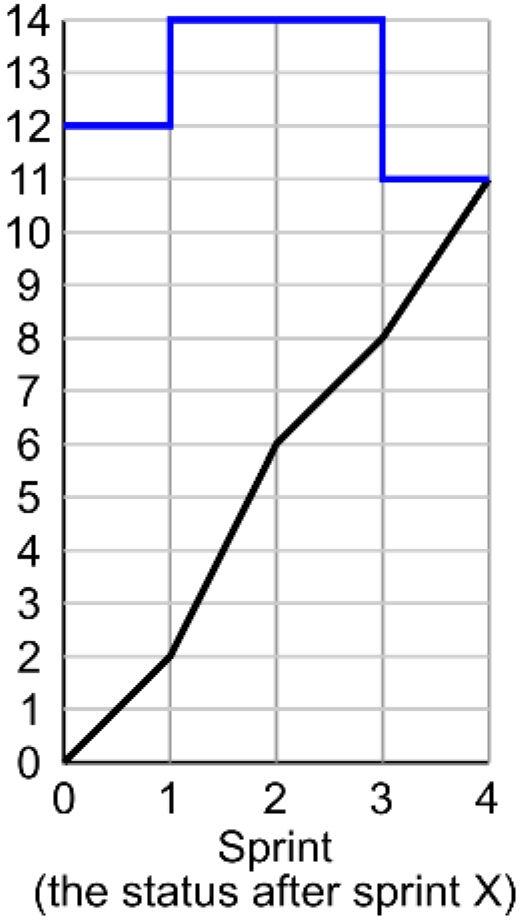
DONE

2

4

2

3



What is (Sprint) Velocity?

- **Velocity is the amount of work completed during a Sprint**
  - i.e. the sum of the estimates of the product backlog items Done during that Sprint.

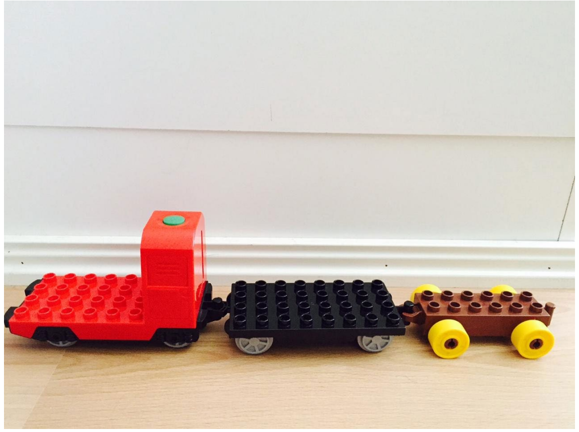
- **If your Sprint velocity is relatively stable, you can use the average velocity of the previous Sprints to estimate Sprint capacity in Sprint Planning Part I.**

Step 2
Sprint planning part I

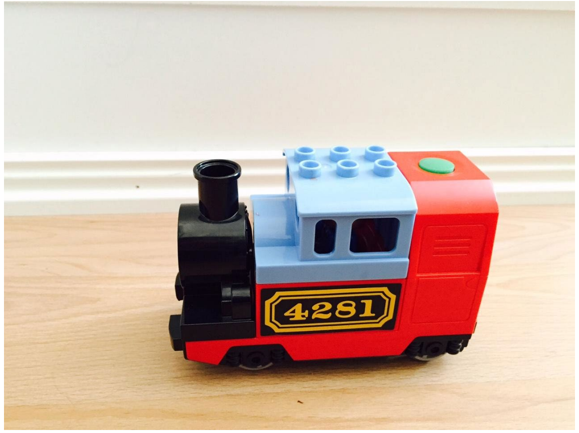
- **Main objective**
  - Ensure that team and PO understand the top items in Product Backlog in the same way. Agree on what will be delivered after the Sprint.
- **Key points**
  - Verify priority of top items compared to the rest of Product Backlog.
  - Verify team understands the purpose ('what' and 'why', use case #1) of the Product Backlog items.
  - Ensure that the team feels confident in achieving their Sprint commitment, and that the confidence seems reasonable compared to past performance.

Splitting product backlog items

- Any backlog item that does NOT fit into a single sprint MUST be split into smaller parts that individually do fit.
- Try to do vertical slices of completed functionality that produce business value instead of horizontal slices.



A horizontal slice



A vertical slice

Step 3
Sprint planning part II

- **Main objective**
  - Create the team's plan for reaching the Sprint commitment.
- **Key points**
  - The Product Backlog items selected for the Sprint are called the "Sprint Backlog"
  - Break them down into smaller tasks or "Actions". The team uses these to plan and coordinate teamwork.
  - Agree on who will start with what initially, but leave enough room for changes later on.
  - Do not forget supporting roles, such as tools engineering (arranging pieces) and system testing (verifying the quality and interoperability of items developed).

Sprint Daily Meeting

- **Main objective**
  - Agree on actions to ensure completing Sprint commitment as well as possible.
- **Key points**
  - Gain situational awareness.
  - Make decisions.

You are free to change the practice the way you see fit, but try to do it in the following way at least once:

> **Go through your Sprint objectives** (the Product Backlog items you took into the Sprint) **one by one, and discuss "Will we get this Done by the end of the Sprint?"**
>
> - If "Yes" with confidence, then no further discussion is necessary
> - If team is unsure, have them figure out what can they do to ensure best possible end result for the Sprint

Sprint Demo

- **Main objective**
  - Demonstrate completed functionality from business perspective to PO to gain feedback for future work.
- **Key points**
  - Show what you have done, tell what is not completely done.
  - Make notes on defects, CR:s, and things to be addressed in the retrospective - write them down!
  - Update project wall.

Sprint retrospective

- **Main objective**
  - Step back from the work and look at how to improve the team's work process.
- **Key points**
  - Identify problems and decide on improvements aimed at solving them.
  - Agree on strong points of teams' process to ensure they are upheld.

Sprint retrospective – An example

- **There are many ways to conduct a sprint retrospective. Here is an example:**
  - Ask everyone to write what went well and what went badly on sticky notes.
  - Gather the sticky notes and read them out loud.
  - Ask everyone to write improvement suggestions on sticky notes.
  - Read the improvement suggestions out loud one by one and discuss them, if needed.
  - Ask everyone to vote by raising their hand if they want to take that improvement suggestion into use.

Breaks

- **Main objective**
  - Let the team lower intensity for a while and take care of themselves, so that they can maintain a sustainable pace over sprints.
- **Key points**
  - Biological needs, humor, rest.

Please remember this is a simulation

- **This is not the optimal way to build Legos.**
  - *If you have attended the simulation before, respect the learning of others*
- **Overhead to work ratio is absurd, that's ok.**
- **Do not touch the code outside Sprint time.**
- **In the first Sprint, do not talk while coding and testing**
  - Scrum Masters are to enforce
- **Product Owner is not exemplary, but rather stereotypical, serving the simulation needs and your learning**
  - *Stereotypical PO does not know Scrum..*

Aalto University
School of Science

If nothing else, remember this:

1) Scrum Simulation Compact Guide for Scrum Masters is your dear friend

2) Keep your walls clear

3) You don't have time! Relentlessly focus the team's attention on the most important question right now

*Ville T. Heikkilä, Maria Paasivaara, Jari Vanhanen,
Towo Toivola, Ferrix Hovi*