

# Scrum Revisited

Jan-Philipp Steghöfer

Chalmers | University of Gothenburg - Sweden

[jan-philipp.steghofer@gu.se](mailto:jan-philipp.steghofer@gu.se)

# Content

---

## Part 1:

- The Scrum Methodology
- Scrum roles
- Backlogs and Work breakdown charts

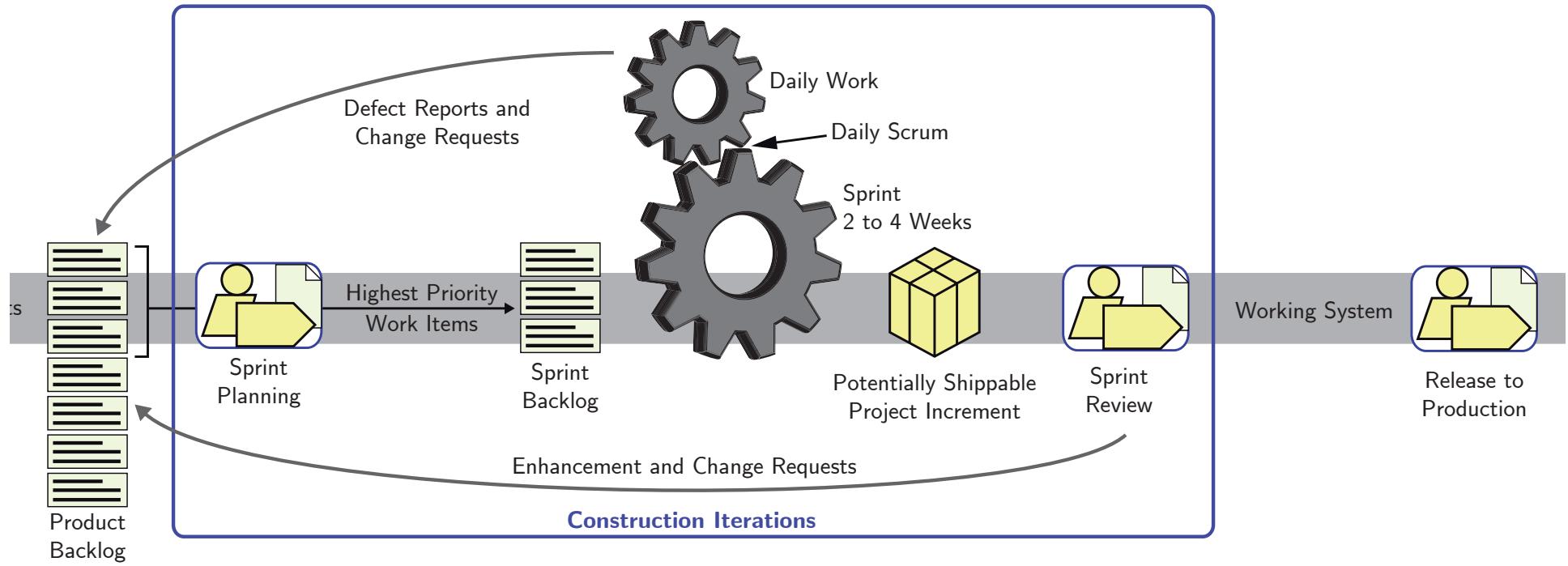
## Part 2:

- Breaking down tasks
- Agile estimation techniques
- Agile project management with Kanban

# The Scrum Methodology



# Scrum



- Iterative-Incremental Lifecycle focused on incorporating change
- Backlogs contain user stories with acceptance criteria

# Anatomy of a Sprint

1. Select user stories to work on
2. Break the user stories down into tasks and refine them!
3. Design the architecture to achieve the user stories
4. Implement/Test/Integrate
5. Integrate with external systems
6. Sprint Review
7. Sprint Retrospectives



Steps 1 to 5 need to be completed in 18 Minutes!  
You come up with the timings!

# Definition of Done

---

- Tells the development team and the customer when the work is finished
- Contains general acceptance criteria
- Is usually defined by the development team

## What is your Definition of Done?

“Yes, it’s done. Just have to fix one or two small bugs. Of course, the handbook needs to be updated. And while we speak of it, maybe we should take a closer look how that works when integrated with the other thing...”

# Definition of Done – An Example

---

- Acceptance criteria is verified during testing
- Coding tasks completed.
- Exploratory Testing completed and signed.
- Regression test reviewed and passed.
- Unit testing – written and passed.
- Code reviews conducted.
- Defects are in an “acceptable” state to the Product Owner.
- User story accepted by the product owner.
- Regression tests run and passed
- Smoke / automation tests run (if applicable)
- Check for memory leaks
- Automated unit tests are checked in
- All code is documented (at least) on the method level

# The Daily Scrum

- “Stand up meeting”, maximum total time: 15 minutes
- Purpose: Synchronisation between members
- Only the members of the Scrum Team are allowed to talk, others may listen
- Everyone answers three questions:
  - What did I do?
  - What will I do?
  - What impedes me?
- Follow-up discussions after the meeting



# Scrum Roles



# The Scrum Master

---

Each group has one Scrum Master.

Her/his tasks:

- Monitor the process lifecycle: update tasks, perform the retrospective
- Implement changes agreed upon during the retrospective
- Remove obstacles
- Communicate with other teams
- Keep the Product Owner in check

The Scrum Master is **not** the manager for the team!

# Product Owner

---

- Serves as the proxy of the stakeholders
- Is usually employed by the customer and embedded with the development team

Her/his tasks:

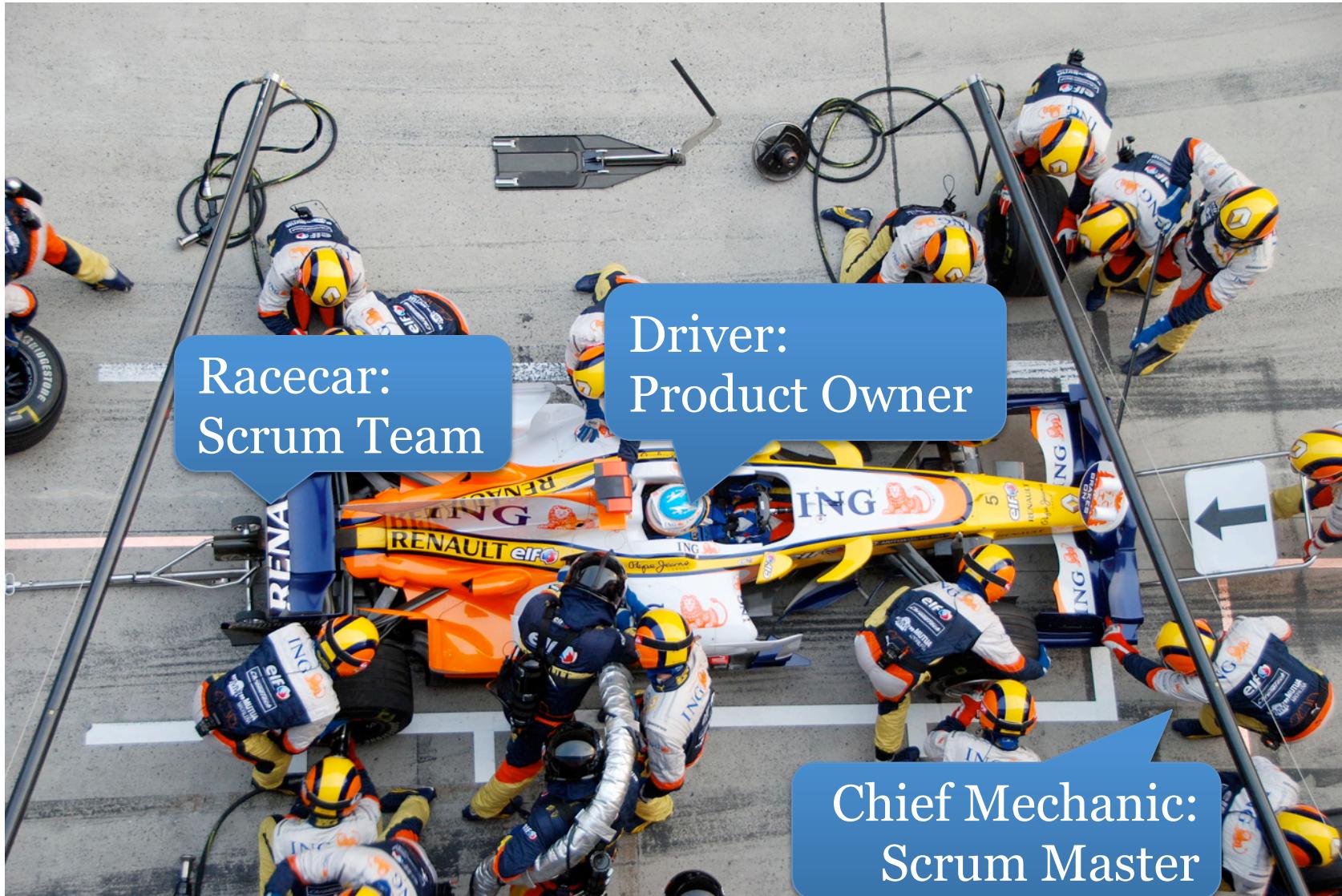
- Formulate requirements
- Prioritise the backlog
- Be available for discussion with the development team
- Define the project scope
- Report to the customer

# The cross-functional Scrum team

---

- Small (ideally 5 to 7 people,  $\pm 2$ )
- Titles are insignificant
- People with different skill sets train each other
- Each team member must be able to design, code, test, document, manage, motivate, debug, ...
- Team is working towards a goal together
- Team takes joint responsibility for process, work environment, and delivered work

# Scrum Roles



# Traditional Roles in an Agile Environment

---

- Project Manager
  - The team manages the project itself.
  - Depending on corporate culture, Scrum Master might be responsible for reporting to management
  - Product Owner reports to customer
- Product Manager
  - Can serve as the Product Owner
  - Can also be the “customer” the Product Owner reports to
- Architect/Developer/Tester
  - These roles are embraced by the members of the cross-functional team

# Backlogs, Tasks, and Burndown Charts



# User Story Cards

- Describes the requirements and the acceptance criteria
- Can also hold information about the estimate (from Scrum Team) and the priority (from Product Owner)
- Can be updated continuously

Front of Card

1B

As a student I want to purchase  
a parking pass so that I can  
drive to school

Priority: ~~Must~~ Should  
Estimate: 4

Back of Card

Confirmations:

~~The student must pay the correct amount~~  
One pass for one month is issued at a time  
The student will not receive a pass if the payment  
isn't sufficient  
The person buying the pass must be a currently  
enrolled student.  
The student may only buy one pass per month.

Copyright 2005-2009 Scott W. Ambler

# INVEST Criteria

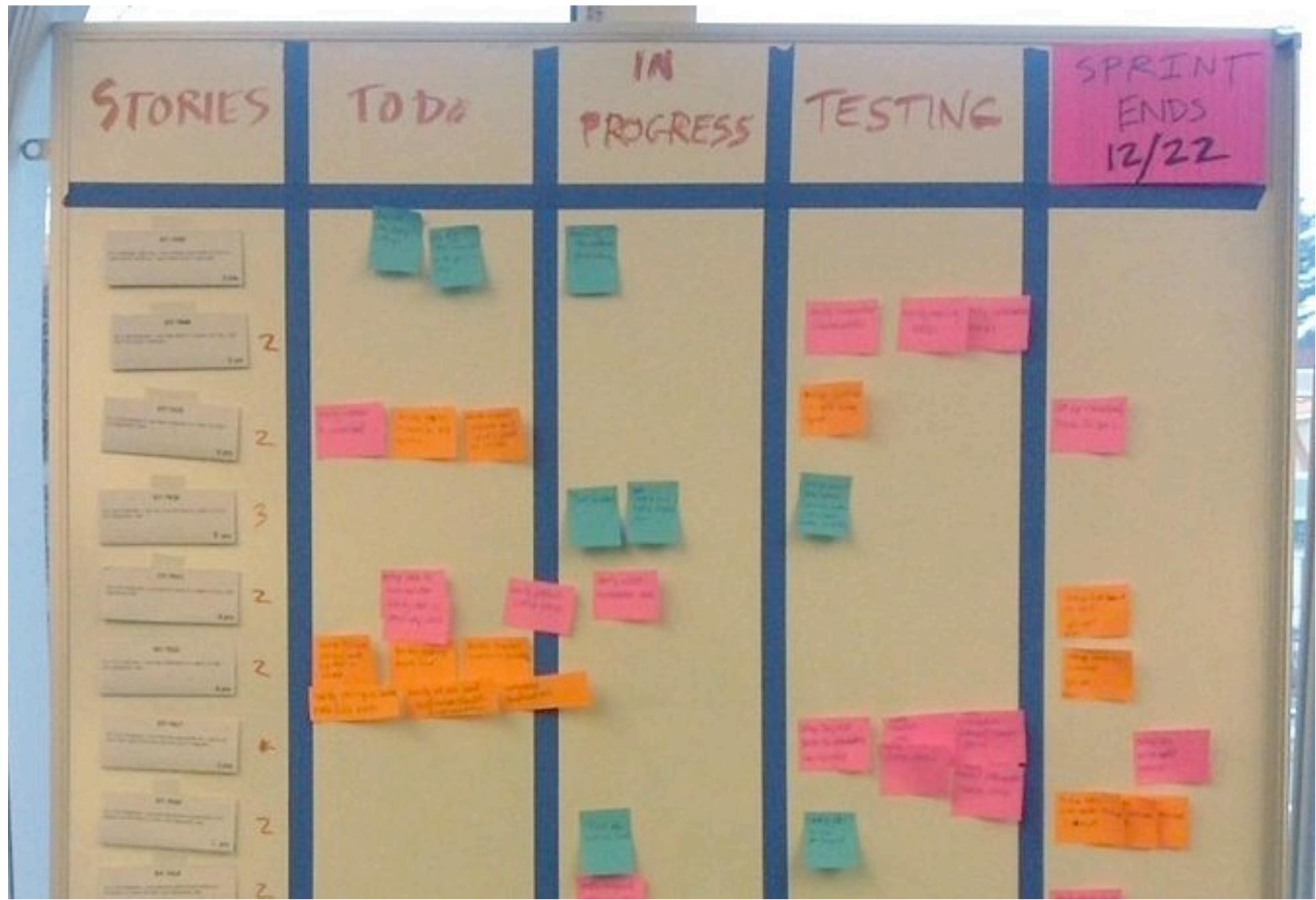
I	Independent	User stories should not overlap and they should be formulated so they can be implemented in any order.
N	Negotiable	A user story should be an invitation for a conversation. It can be changed, augmented, and redacted; of course, always in dialog with the Product Owner!
V	Valuable	Each user story should deliver value, either to the Product Owner or to Scrum Team.
E	Estimable	It must be possible to assign effort to each user story. A story that can not be estimated is not complete!
S	Small	A user story must be a manageable task. If its completion takes longer than 3 or 4 days, it must be broken down!
T	Testable	There must be clear, testable criteria to define when the story is done in the eyes of the Product Owner and the Scrum Team.

[Buglione & Abran, 2013]

# Working with the backlog

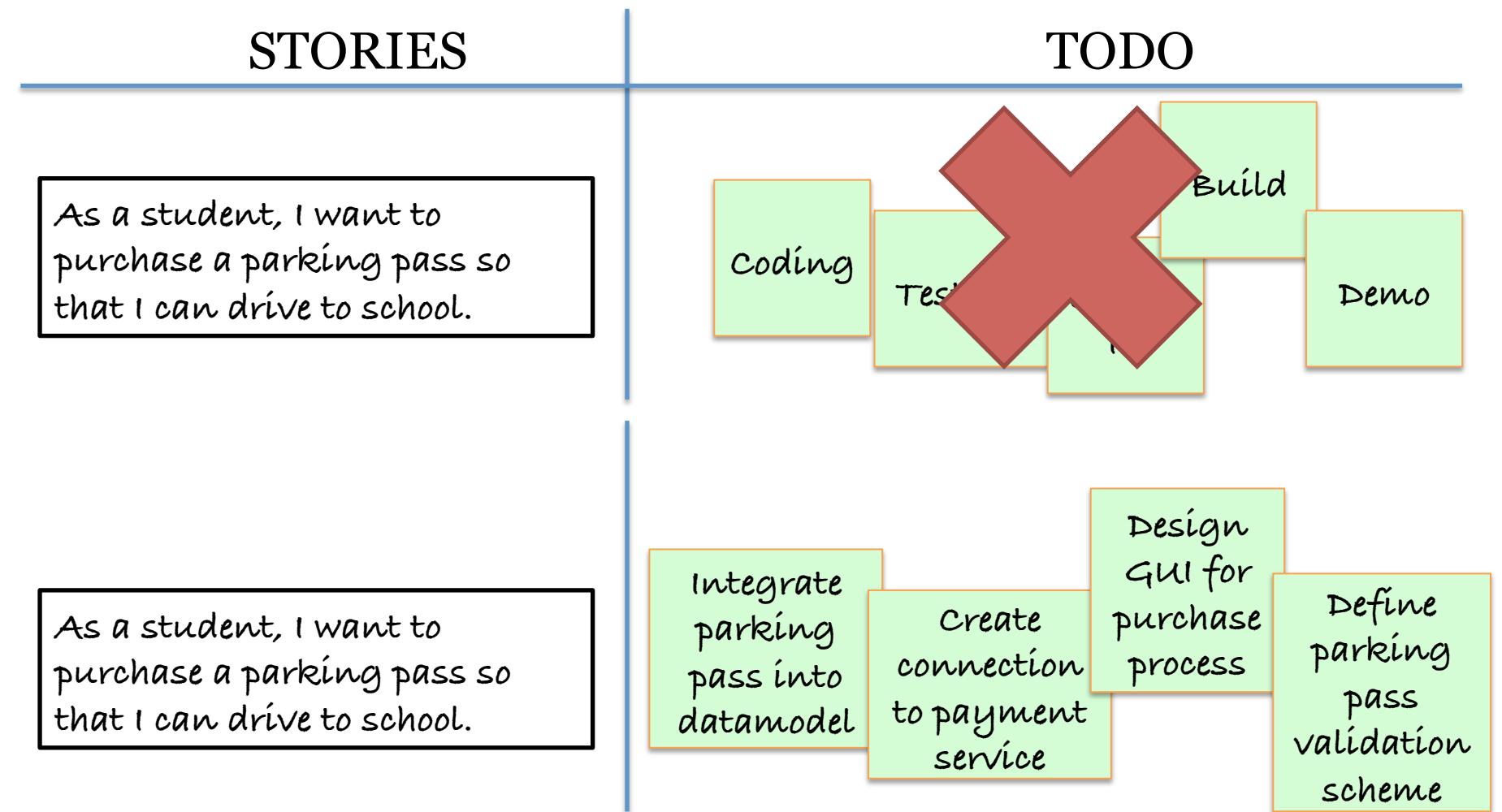
---

- Differentiation product backlog/sprint backlog
  - Product backlog: Prioritised list of **all** product requirements
  - Sprint backlog: List of requirements selected **for the current sprint**
- Product backlog can be continuously updated by the Product Owner
- Sprint backlog contains more detailed information: tasks
- Sprint backlog is only updated by Scrum Team

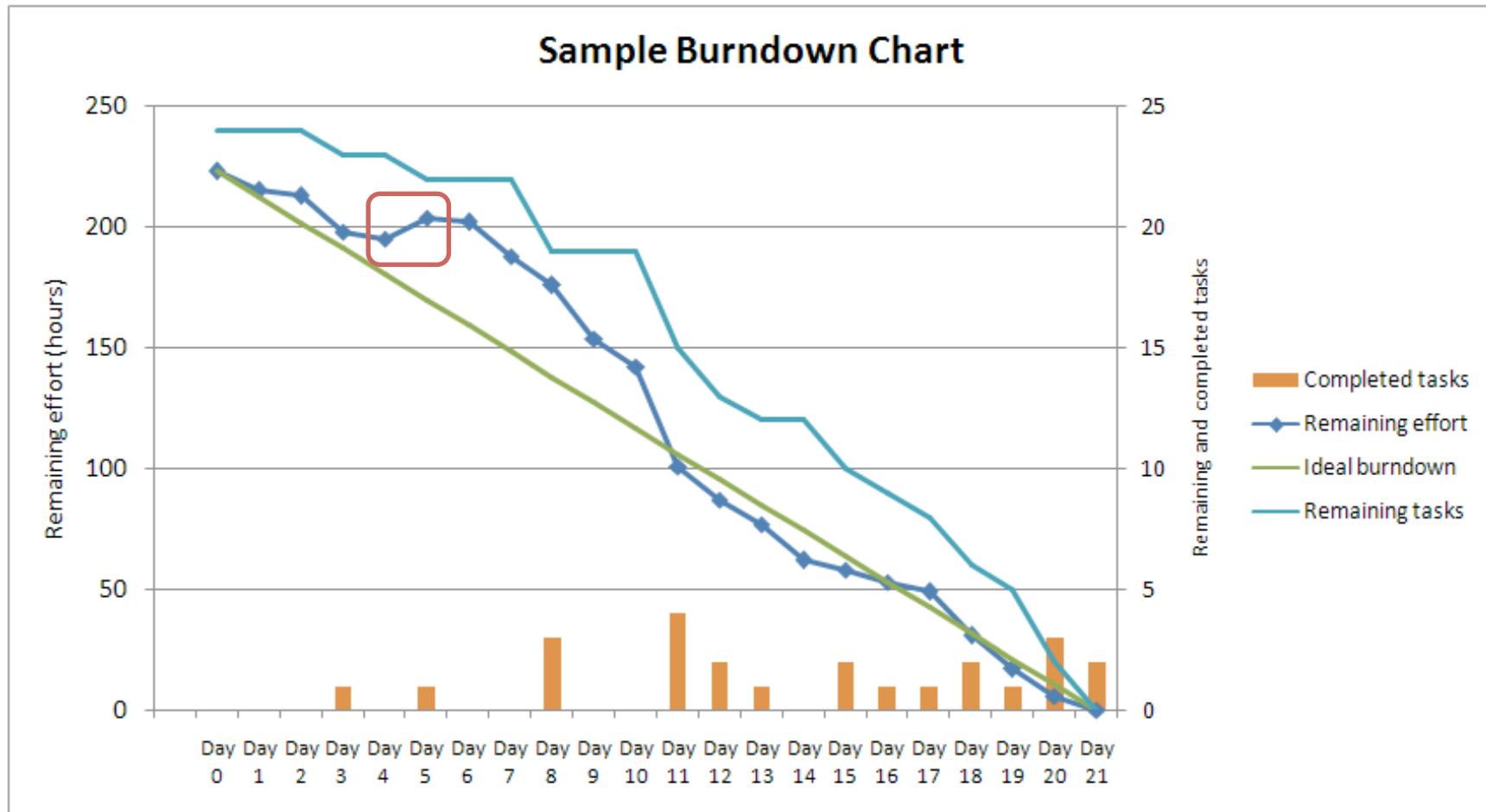


# Tasks

- User stories are usually “too big” to tackle as one



# Creating Burndown Charts

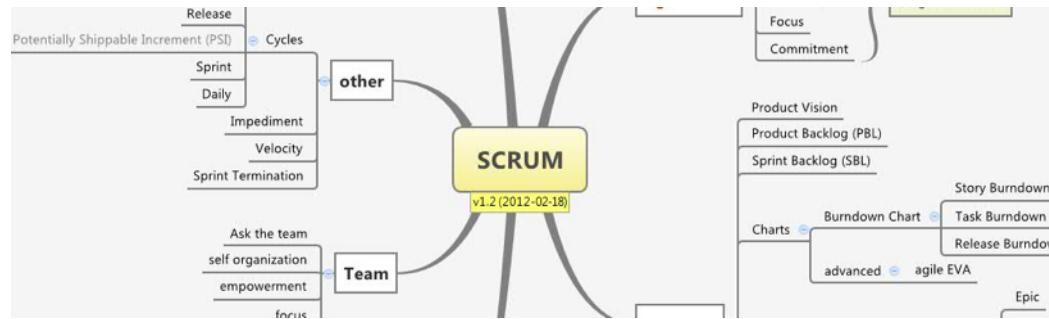


<https://upload.wikimedia.org/wikipedia/commons/0/05/SampleBurndownChart.png>

# Exercise



- Have a look at the mind map at <http://www.xmind.net/embed/9uVM/>



- Identify two topics you haven't heard about or you know little about
- Look for information about those topics and try to understand them
- Prepare some notes that allows you to explain the topic to your fellow students

# Breaking Down Tasks



# Vertical slices

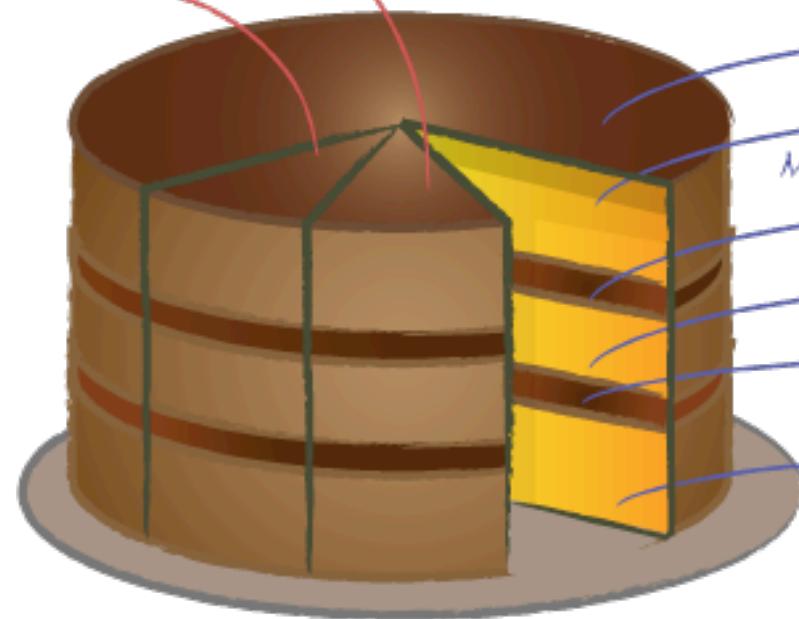
Automated Teller Machine (ATM)

Horizontal and Vertical User Stories - Slicing the Cake

## Vertical User Stories

Cash Withdrawal (90% usage)

Bank Statement



## Horizontal Stories

UI - PIN and Card Reader

Security Layer

Middleware - Transaction Protocol

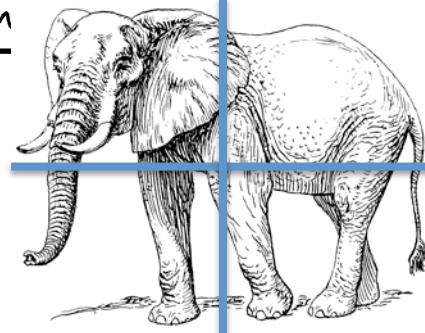
Tuxedo DB Interface

Transport Protocol

Bank Mainframe Database

# Elephant Carpaccio/Laminating

As a student, I want to purchase a parking pass so that I can



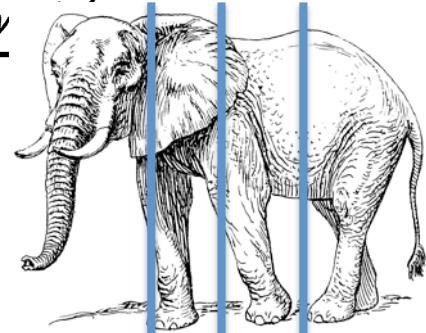
Integrate parking pass into datamodel

Create connection to payment service

Design GUI for purchase process

Define parking pass validation scheme

As a student, I want to purchase a parking pass so that I can



Implement purchase process without payment

Integrate payment into the purchase process

Allow users to see their past orders

Define parking pass validation scheme

# Agile Estimation Techniques



# Planning Poker

Purpose: Estimate the relative effort required to complete user stories

- Introduce the user story
- Each team member puts down a card with their estimate (face down)
- All team members turn card at the same time
- Outliers justify their estimate
- Repeat until consensus is reached



Cards use a modified Fibonacci sequence:  
forces developers to account for estimation uncertainty

# Alternatives to Planning Poker

---

## Team Estimation

- General idea:
  - Put the stories in a relative order
  - Assign points to the ordered stories
- Relative order can be created by using one story as an anchor and placing stories above/below if less/more complex
- Stories of the same complexity can be stacked
- Assign points by starting with a “1” and assigning it to the correct pile. Then work your way up from there.

<http://nerds-central.blogspot.com/2011/10/agile-alternative-to-planning-poker.html>  
<http://www.netobjectives.com/files/books/lasd/TeamEstimationGame.pdf>

# Alternatives to Planning Poker

## Planning Chips

- Each team member gets a number of poker chips that corresponds to two hours of their availability
- Team members distribute their chips to those stories they plan to work on
- Shows who is available clearly
- Makes the team that works on the same story explicit



<http://blog.controlgroup.com/2015/07/13/a-modest-proposal-for-planning-poker-alternatives/>

# Agile Project Planning with Kanban



# Kanban and Scrum

---

- For some projects, Scrum's fixed length iterations are not suitable
- Code should either be shipped continuously (continuous deployments) or in flexible increments determined by the development team or the customer
- Kanban provides lean project management embracing many of the Scrum principles while providing flexibility

# Kanban Board

Backlog	Ongoing	In Review	Done
Define parking pass validation scheme	Allow users to see their past orders Emil	Allow returning unused parking passes Mia	Implement purchase process without payment John LT: 6PH

Work in Progress is limited to avoid overloading developers  
Lead time (average time to complete) is measured

# More Resources



# More resources

---

- <http://scrumtrainingseries.com/>  
Series of webinars on different Scrum topics (with quiz!)
- <https://scrumalliance.org/community>  
Articles about Scrum in practice
- <http://www.solutionsiq.com/what-is-scrumban/>  
Discusses combining Scrum and Kanban
- <http://scrumandkanban.co.uk/slicing-stories-vertically/>  
A chart on splitting user stories

# Questions



# Bibliography

---

[Cohn, 2005] Mike Cohn. *Agile Estimating and Planning*.  
Prentice Hall, November 2005

[Buglione & Abran, 2013] Buglione, Luigi, and Alain Abran.  
*Improving the user story agile technique using the invest criteria*. Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), IEEE, 2013.