

# DAT257

## Lecture 3: Scrum and Software Quality

Jonas Petré

**Welcome! We will begin in a few minutes**

# Jonas Petrén

**10 years** in software testing, development and as test manager. **6+ years of Scrum Master experience.** Full-time Scrum Master since beginning of 2018. Employed by HiQ since 2012.

Civilingenjör Informationsteknologi, Linköping University, 2003-2008

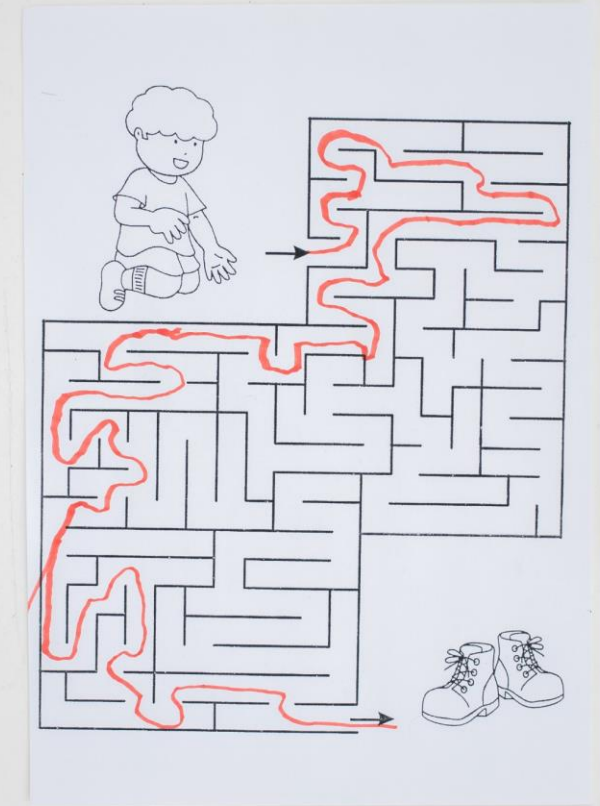
**Today:** Senior Scrum Master/Agile coach  
Certified **Professional Scrum Master III**

*HiQ*



# The goal for today

We go deeper into the world of agile and Scrum. You will learn more about the roles, events and artifacts that help build strong teams and quality products



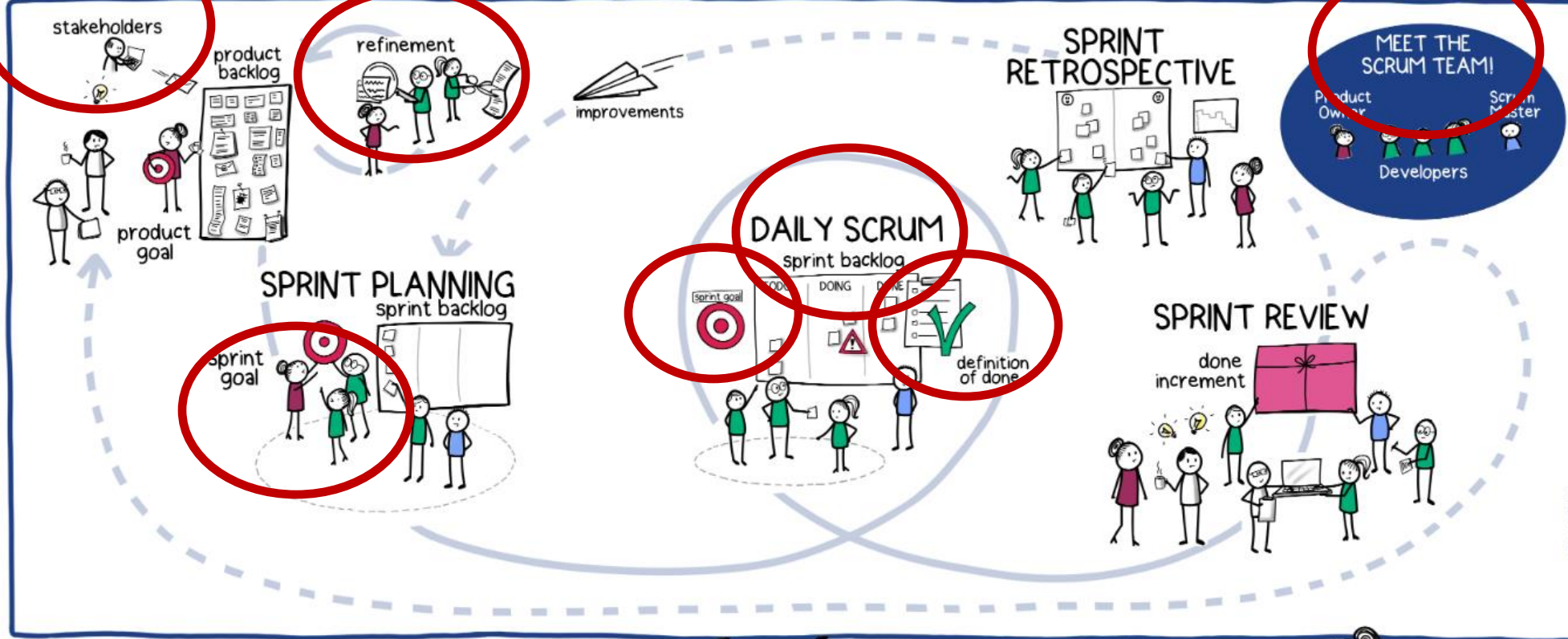
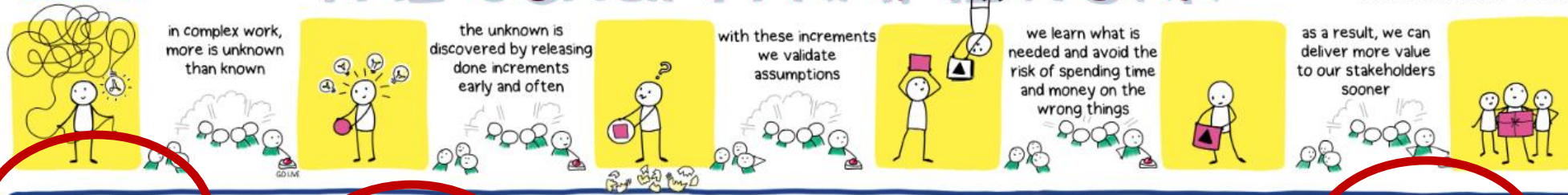
# Agenda

- More Scrum
  - More about the Scrum roles
  - Daily Scrum and Backlog refinement
  - Who is a stakeholder?
- Software quality
  - Definition of done
  - Sprint goals
- Misunderstandings about agile
- Ending





# THE SCRUM FRAMEWORK



The background of the slide features a light blue gradient. Scattered across this background are numerous 3D arrows. Most of the arrows are white or light gray, pointing in various directions. A single, prominent red arrow is located in the upper left quadrant, pointing towards the top right. The arrows have a slight shadow, giving them a three-dimensional appearance.

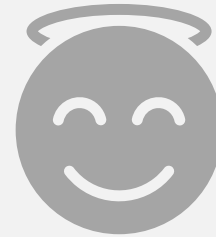
# Follow-up

From previous lecture

# Feedback from lecture 2



WHAT CAN BE IMPROVED:  
SLOW PACE



WHAT WAS APPRECIATED:  
REAL LIFE EXAMPLES



# A puzzle with 2000 pieces. How many pieces remain?

Don't say your answer out loud. Send it to [petrenjo@chalmers.se](mailto:petrenjo@chalmers.se), subject: "Puzzle"







76 answers



Mean value: 457



Median value: 400

Agile done right beats  
competition





# PROJECT SUCCESS RATES

## AGILE VS WATERFALL

METHOD

SUCCESSFUL

CHALLENGED

FAILED

**AGILE**

42%

47%

11%

**WATERFALL**

13%

59%

28%

WWW.VITALITYCHICAGO.COM

Source: Standish Group Report 2020



# Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

Individuals and interactions over processes and tools  
Working software over comprehensive documentation  
Customer collaboration over contract negotiation  
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck  
Mike Beedle  
Arie van Bennekum  
Alistair Cockburn  
Ward Cunningham  
Martin Fowler

James Grenning  
Jim Highsmith  
Andrew Hunt  
Ron Jeffries  
Jon Kern  
Brian Marick

Robert C. Martin  
Steve Mellor  
Ken Schwaber  
Jeff Sutherland  
Dave Thomas

Team-  
based  
culture

S





Scrum is good for  
complex work



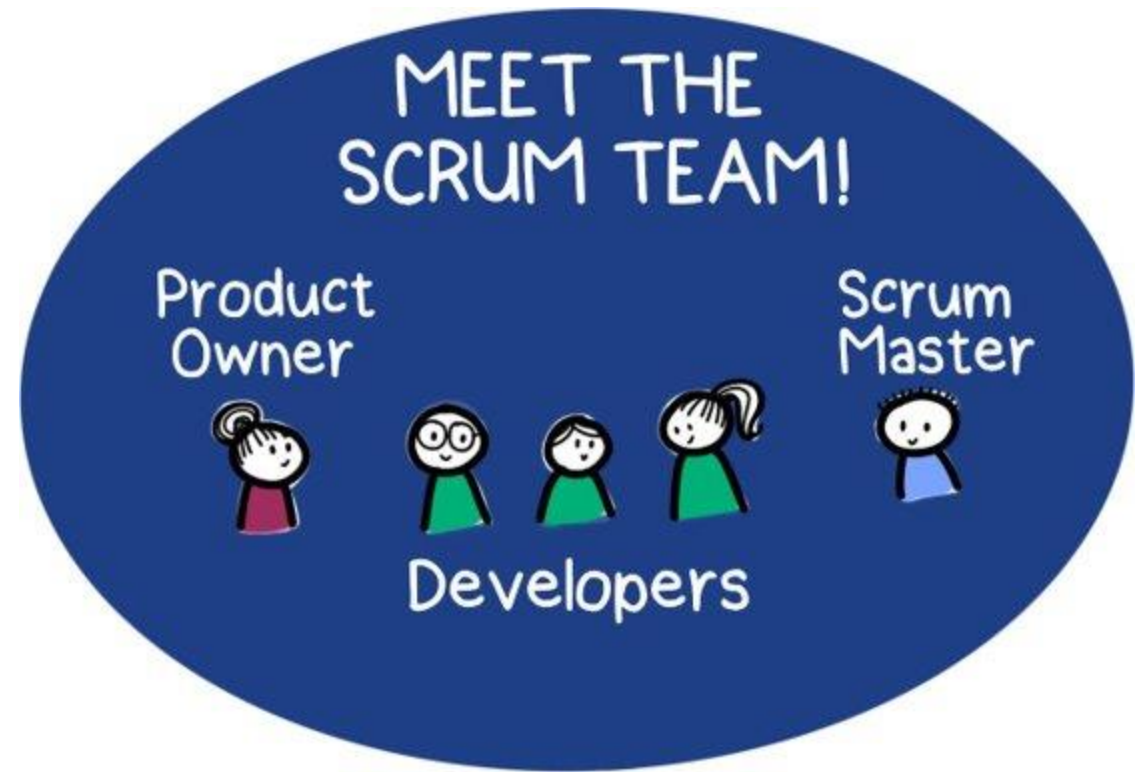
# Scrum ~~roles~~ accountabilities

**The Product Owner** is accountable for  
**MAXIMIZING THE VALUE OF THE PRODUCT**

**The Developers** are accountable for  
**CREATING A FLEXIBLE PLAN** for the sprint  
and to **MAINTAIN QUALITY** by adhering  
to the Definition of Done

**The Scrum Master** is accountable for the  
**SCRUM TEAM'S EFFECTIVENESS**

**The entire Scrum Team** is accountable for  
creating **A VALUEABLE AND USEFUL  
INCREMENT EACH SPRINT**





# Characteristics of a good... Product owner

Knows the product

Knows the market (the users, how they use it, their willingness to pay for it...)

Can make decisions

Is leading the team with a clear vision



# Characteristics of a good... Developer

Knows the technical domain

Cares for the code

Knows how to behave in a team

Knows how to finish things

Knows how to get started

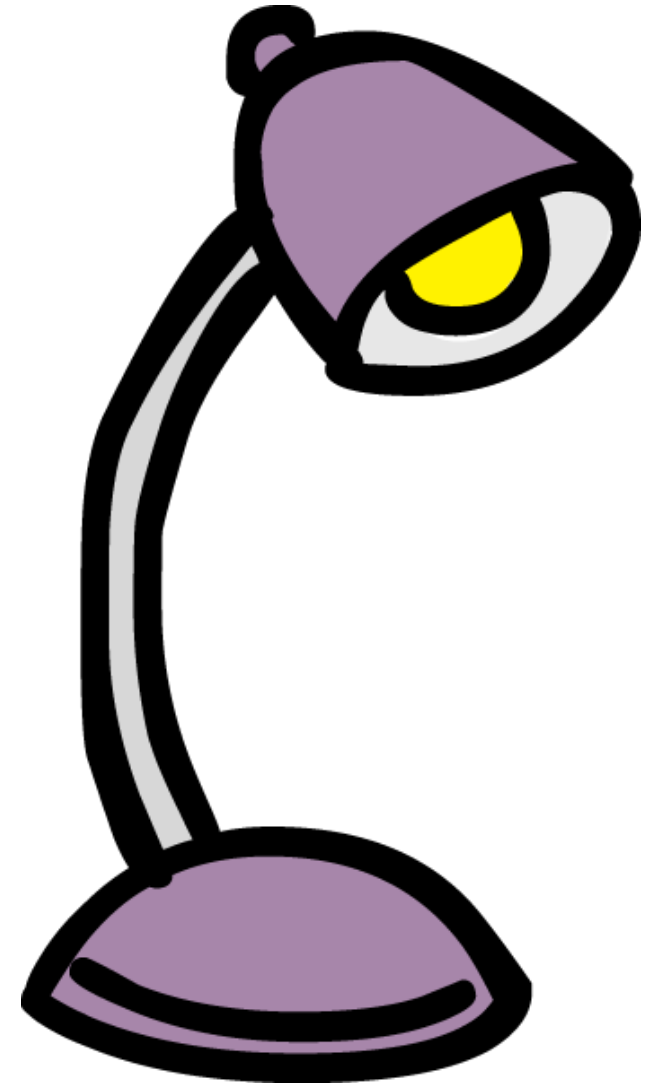
Is communicative

Helps out



# Characteristics of a good... Scrum Master

- Knows Scrum and agile
- Is a good listener
- Can lead by example
- Knows how to build a great team
- Knows sufficiently about the product and market
- Knows sufficiently about the technology used







# What does a Scrum Master do all day?

Takeaway: Above all, empower the team

# Characteristics of a good... Team

---

Reaches most of their  
sprint goals

Gets better over time

Everyone is included

Members want to  
stay



# About Daily Scrum and Backlog refinement



# Daily Scrum

What it is: An opportunity to **PLAN COLLABORATION** and see that we are heading towards the sprint goal

It's **not** a status meeting where people are held to account for their work items

# Daily Scrum – Some things to talk about

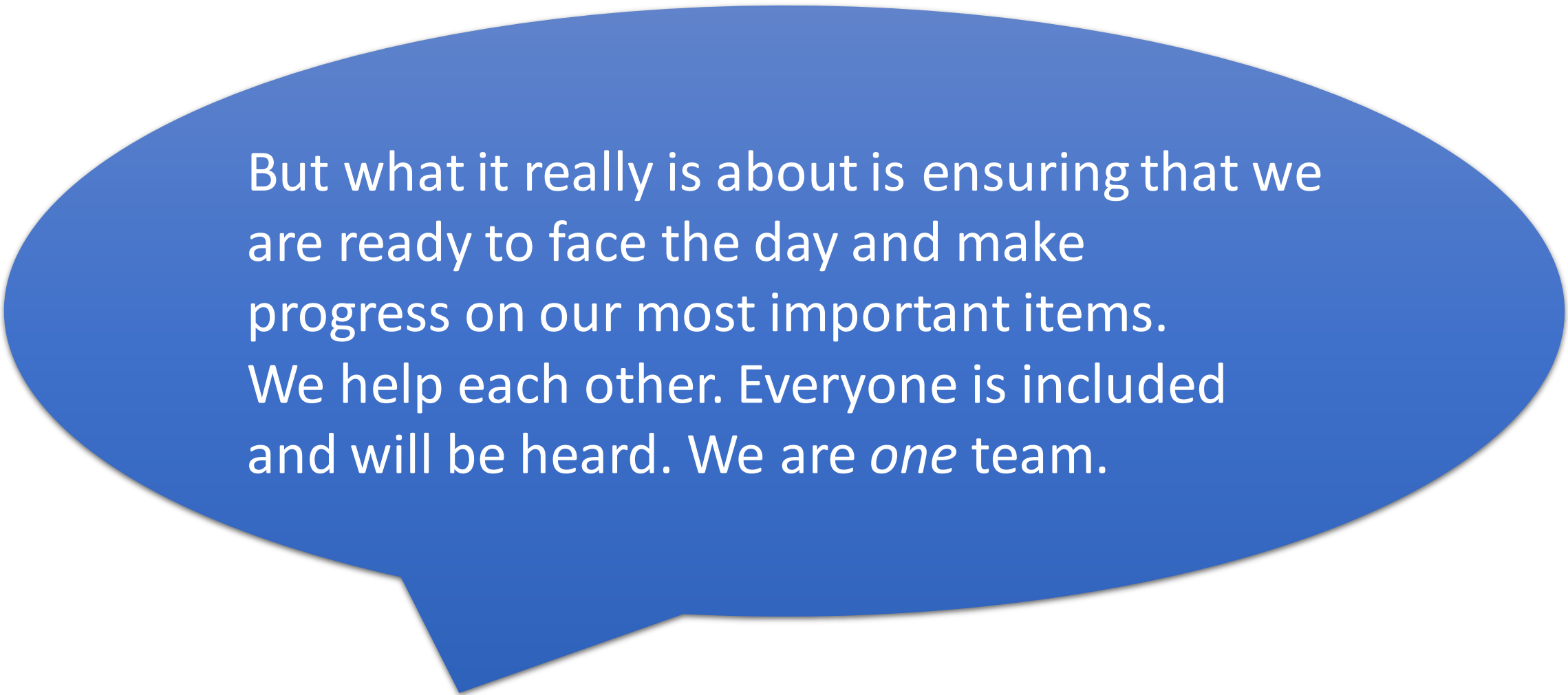
Are we on track with reaching the sprint goal?

Do we have work items that are stuck?

How can we prepare us for the sprint review? What would we like feedback on?

Is someone working on things that are not on the board? Why?

# Daily Scrum



But what it really is about is ensuring that we are ready to face the day and make progress on our most important items. We help each other. Everyone is included and will be heard. We are *one* team.



# What can a Scrum Master do during Daily Scrum?

- Listen closely. Is someone stuck and need help? Are we making progress?
- Take notes and ask follow-up questions (*"Sofie, you said yesterday that... how did it go?"*)
- Make sure everyone is heard. Don't let one or a few members take all airtime.
- Ask how you can help
- Remind everyone about the sprint goal

Takeaway: Above all, empower the team



# BACKLOG REFINEMENT is like cleaning your refrigerator

Some things will soon pass the best before-date and need to be eaten soon **(HAVE HIGH PRIORITY)**

For some food storage containers it's not clear what they contain **(DETAIL, CLARIFY)**

Some big storage containers with meat need to be divided to separate containers **(WORK BREAKDOWN)**

You can make a nice salad dressing when combining things **(MERGING OR LINKING ISSUES)**

Some stuff is **THROWN AWAY**



# Who is a stakeholder?

People outside the Scrum team who have an interest in the product developed by the team

For example, people that:

- are affected by the product

- are accountable/held responsible for it

- have decision power

- are supporters

- are slowing us down/hindering us

- are influential and want to steer us in their direction







- users
- other development teams
- direct managers
- sales department
- legal department
- If you have people that will work with maintenance of the product
- Investors



## Discussion 2 minutes

For the Lego cities that you built, give me examples of stakeholders? E.g. who is a stakeholder of a park/hotel/office building in your city?





# Characteristics of a good... Stakeholder

Knows what add value to the product

Provides feedback

Collaborates often

Let the team run their game,  
does not interfere



# Quality

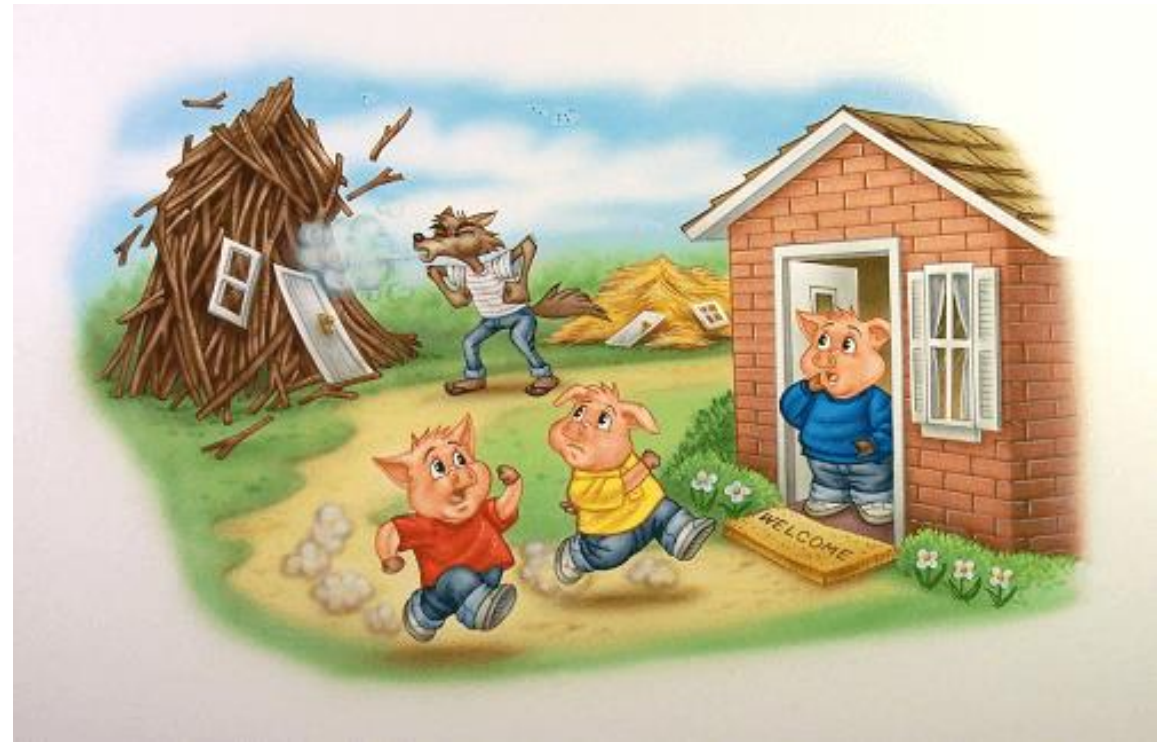
*“The most expensive way to test your idea is to build production quality software.”*

Jeff Patton, agile coach and trainer

# DEFINITION OF DONE – Applicable to all items

*The house must have*

- ☐ Four walls
- ☐ A door
- ☐ Windows
- ☐ Pointy roof



[Picture](#)

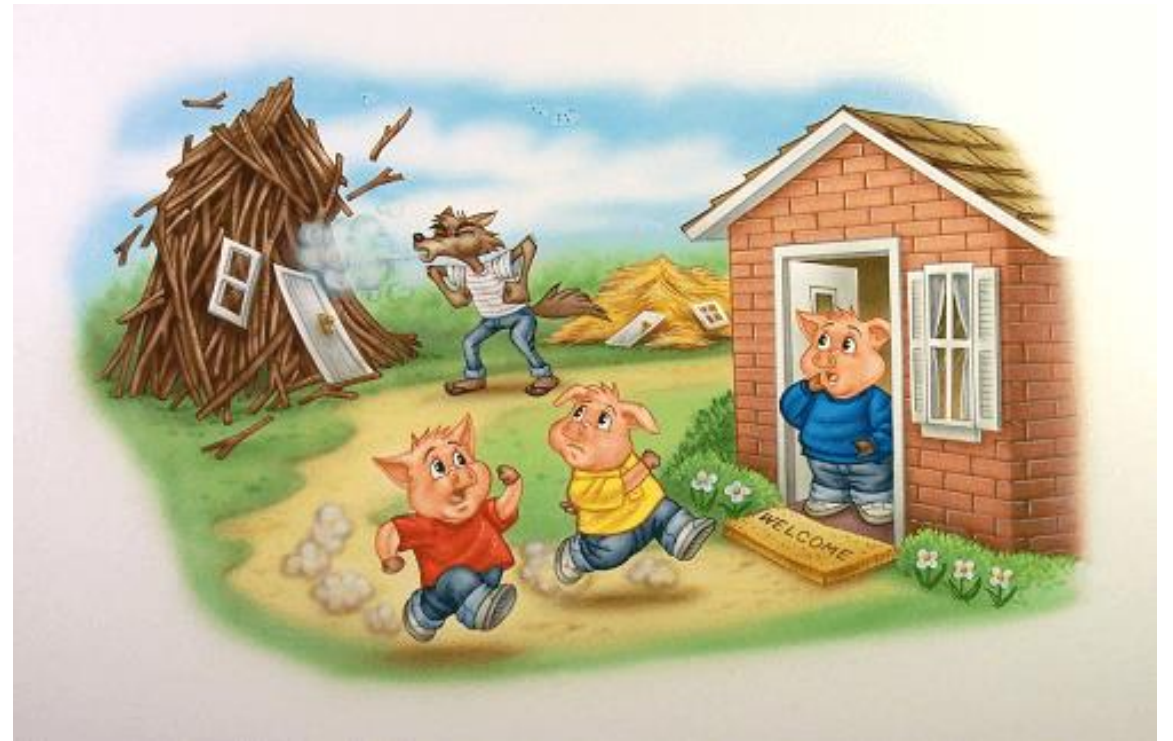


# ACCEPTANCE CRITERIA – Applicable for a single item

**Fifer Pig:** The house should be built of **straw**

**Fiddler Pig:** The house should be built of **sticks**

**Practical Pig:** The house should be built of **bricks**



[Picture](#)

# Advantages with Definition of Done

Improved **QUALITY**

**COMMON RULES** for when a work item is done

**DECREASED RISK** that we miss something that will bite us back later





## Definition of done

user  
story  
level

### Skog & Fjäll

- Unit tests, component tests, lambdas and/or test bots created or updated
- Code review
- Manual test by yourself and someone else
  - check all acceptance criteria
- Ready for deployment
  - feature flags
  - deploy notes
  - permissions

Showcase  
in Jira  
what is done





### Golden Definition of Done

- Acceptance criteria's
  - Provide evidence of acceptance criteria achieved
  - Non functional requirements met
- Solution reviewed in detail
  - Code should be reviewed
  - Code follow the standards
    - Pushed to central repository
    - No warnings in code or during build
    - No decrease in SonarQube quality statistics
- Tests added
  - Automated, executed and passing
  - Unit, component, integration, UI, load, performance, ...
  - Behaviour tests (verify the acceptance criteria's) automated or manually
- Documentation in place
  - Flow Help
  - Jira / Azure DevOps
  - Architecture (flows, decisions etc.)
  - Release notes
  - Test cases updated
- Changes in tools, configuration, prerequisites etc should be reviewed
- Tested/demoed in a deployed environment
  - Prove acceptance criteria's
  - Show how it works
  - Use large amount of data/users
  - Changes that might affect installations should also be demoed (local installation is ok)
  - Recorded demos are sufficient
- Verify that crossteams communication has been in place
- Production ready
  - No known defects affecting the overall business value/process
  - No undocumented changes needed to release
- Reviewed by specialist/stakeholders
  - QA
  - PO
  - Stakeholder

# SPRINT GOALS

So that we are working on something  
that is valuable and inspiring

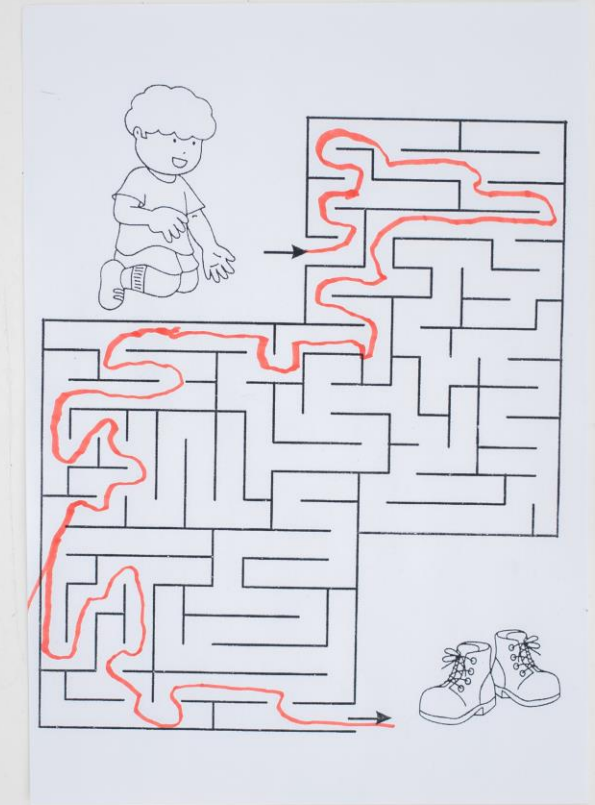
# Advantages

Increases **SYNERGIES** and **CREATIVITY**

Increase likelihood of **TEAM WORK** –

Reduced risk for individual or sub-team work

Decreased risk of **MONOTONY** ("we are just a task-solving machine")





"Top down" – start with an inspiring goal, then pull work

1. What is our **sprint goal**?
2. **Pull in the necessary work items** which are required to meet the sprint goal

### Examples of good sprint goals

- Get feature X ready for release (hereby the Sprint Goal is **delivering a feature**)
- Check if the architecture enables the desired performance (hereby the Sprint Goal is **addressing a risk**)
- Test if users are willing to register before using the product features (hereby the Sprint Goal is **testing an assumption**)

[From Barry Overeem](#)

Don't do this! "Forced fit"

1. Pull from the top of the backlog as much as you can

2. Try to force a sprint goal that fits



Bad sprint goal:  
Complete all items



Misunderstandings about agile



Agile is only for developers?

No, agile is for everyone. Remember:  
agile is a mindset and we are stronger  
together if we all act accordingly



Agile means “we don’t have a plan”

No, we have short-term plans (the sprint goal) and long-term plans, but we review them regularly. And plans are *expected* to change



Agile means we can adjust to *any*  
situation

No, but we think it is more important  
to respond to changes over following a  
plan



## Does agile mean no documentation?

No, we still have to do the necessary documentation, but we value working software more. And face-to-face communication is preferred over written communication



## Blaming Agile: “I Tried It Once and Didn’t Like It”

Don’t blame Agile for your failure;  
that's just bad workmen blaming the  
tools. Having a successful work totally  
depends on your hard work.



## Agile is undisciplined

The truth is Agile is a very disciplined way of delivering software.

- You have to test
- You have to get feedback
- You have to regularly ship software
- You have to change and update the plan
- You have to deliver bad news early



Success for a Scrum Master is a  
successful project?

No, success for the Scrum Master is if  
the team is continuously improving



Success for a project is on time and  
budget?

No, success for a project is measured  
in business value and customer  
satisfaction



# Key takeaways





Agile done right beats  
competition





The results of a team/product will be better if everyone knows how to play their part i.e. what behaviors that are valuable



Definition of done  
and **sprint goals** are  
key factors for  
quality









Thea Schukken  
[www.beeldinwerking.nl](http://www.beeldinwerking.nl)