

# Scrum Events and Artifacts

## Lecture 4

SoftUni Team  
Technical Trainers



**SoftUni**



Software University

<https://about.softuni.bg>

# Table of Contents

1. Recap Lecture 3
2. Scrum Artifacts and Events
3. Phases in Scrum
4. Q&A



# Have a Question?



**sli.do**

**#Agile**



# **Scrum Artifacts and Events**

## **How?**

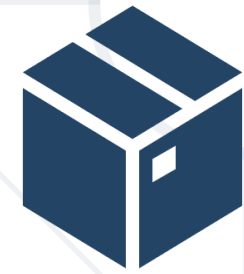
# Scrum Artifacts (1)



Product Backlog



Sprint Backlog



Product Increment

# Scrum Artifacts (2)



Product Backlog

The 1<sup>st</sup> artefact



Sprint Backlog



Product Increment

# The Product Backlog

- An **ordered** list of **everything** we **know** is needed
- **Single** source of truth for any product changes
- **Responsibility** of the **Product Owner**
- A never complete, **living** document
- As close as it gets to a **contract**

# The Product Backlog - Example

PRODUCT BACKLOG EXAMPLE						
	As a...	I want to be able to...	So that...	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators	Must	2	Work in Progress
10	Member	update my email preferences	I'm not bombarded with junk email	Should	2	Work in Progress
11	Member	share content to social networks	I can promote what I find interesting	Could	2	Work in Progress
12	Visitor	create an account	I can benefit from member discounts	Must		To be started
13	Visitor	login	I can post new entries	Must		To be started
14	Visitor	add comments	I can have a say	Must		To be started
15	Visitor	suggest improvements	I can contribute to the site usability	Should		To be started
16	Visitor	contact the Administrators	I can directly submit a query	Could		To be started
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting	Should		To be started
18	Visitor	view a member's profile	I can know more about a member	Must		To be started
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from	Must		To be started



# The Product Backlog (1)

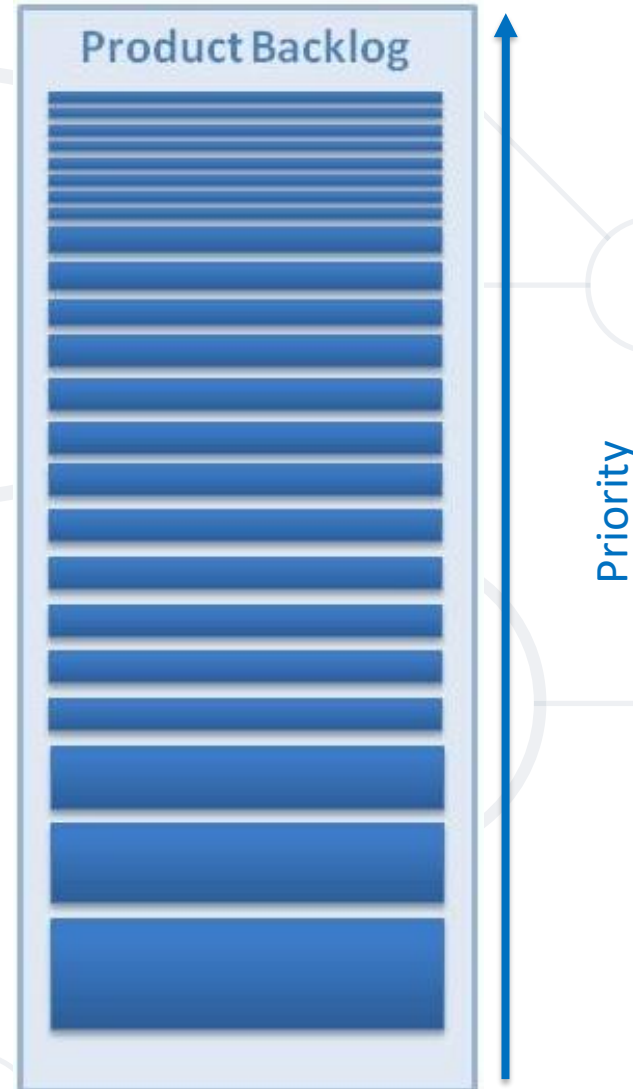
- The most important document which determines the success or the failure of the project
- It represents all goals and desired outcomes
- The Product Backlog is Agile in its nature...it evolves and can change based on the current needs
  - How is this a benefit?
  - Wouldn't that create more of a confusion?

# The Product Backlog (2)

- Characteristics of a good Product Backlog
  - It is visible to **all stakeholders**
  - It could be a means to justify the work produced by the Scrum Team
  - ...so what does this mean?

# The DEEP Product Backlog

- **D**etailed *Appropriately*
- **E**stimated
- **E**mergent
- **P**rioritized



# The Product Backlog – How to Make It "Detailed"?

- There is no single format that should be used
- The backlog should give us just the right level of detail
- The Product Owner owns the Product Backlog

PRODUCT BACKLOG EXAMPLE						
	As a...	I want to be able to...	So that...	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators			

- What is a "User Story"?
- User stories are short, textual descriptions of required functionality – Role → Goal → Reason

As a <ROLE>, I want to <GOAL>, so I can <REASON>.

- Description of a software feature written from the perspective of the end user
- It always tells us the story from User's point of view

# User Story – Example (1)

As a user I can login  
so that I can access  
my dashboard

- Username at least 6 characters
- Password must contain number and capital letter
- Lock out after 3 failed attempts



# User Story – Example (2)

## Add Prospect

As a property manager I want to add a new prospect to the lead management system so I can track my interactions with the prospect.

## Conditions of Satisfaction

Capture name, email, phone #, contact date, contact format, lease type, and move-in date

Verify prospect is associated with an existing campaign



# User Story – Example (3)

As a user, I can  
cancel a reservation.

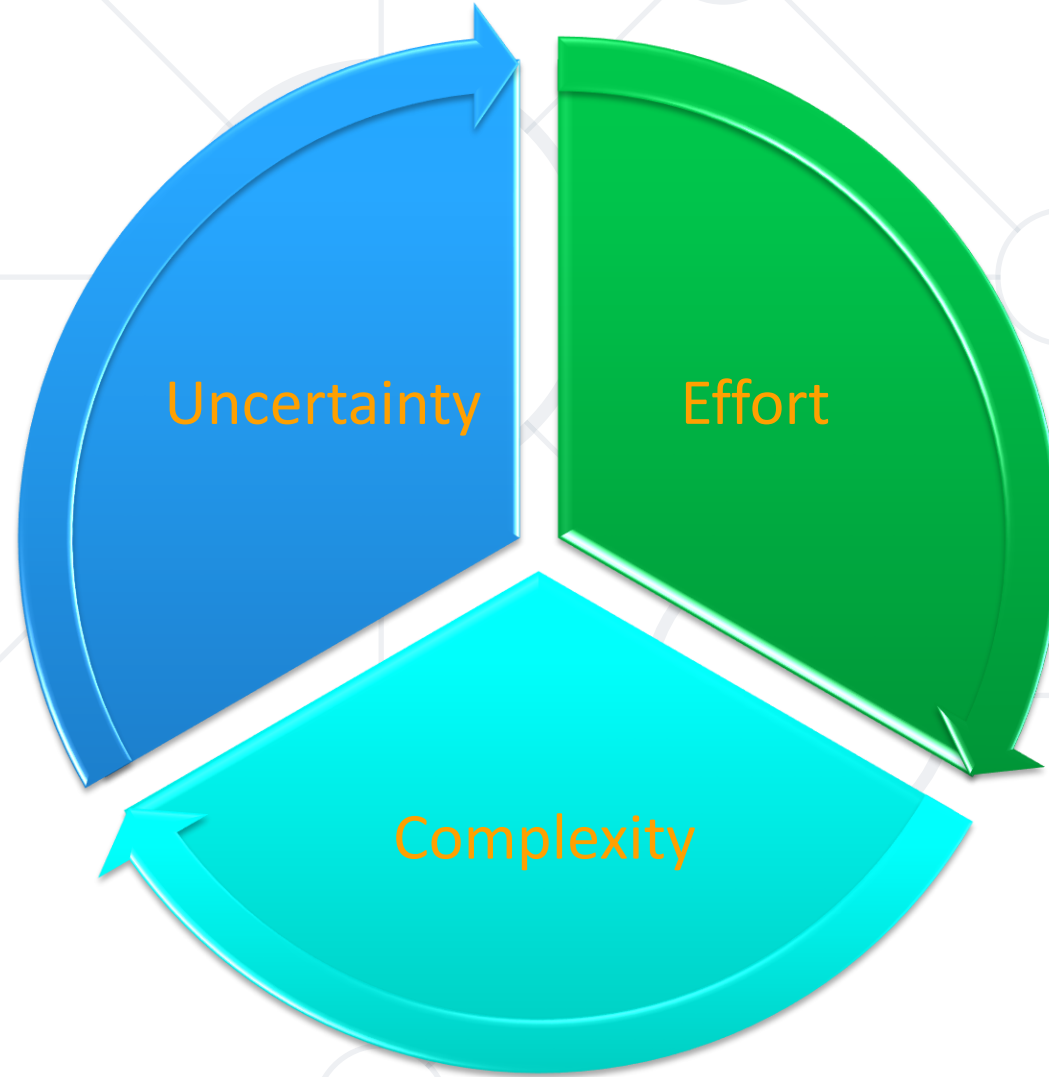
- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

- Verify that a premium member can cancel the same day without a fee.
- Verify that a non-premium member is charged 10% for a same-day cancellation.
- Verify that an email confirmation is sent.
- Verify that the hotel is notified of any cancellation.



- Relative sizing
- Story points – Fibonacci sequence – 1, 2, 3, 5, 8, 13, 21
- T-shirt sizes – XS, S, M, L, XL, XLL

# Story Points



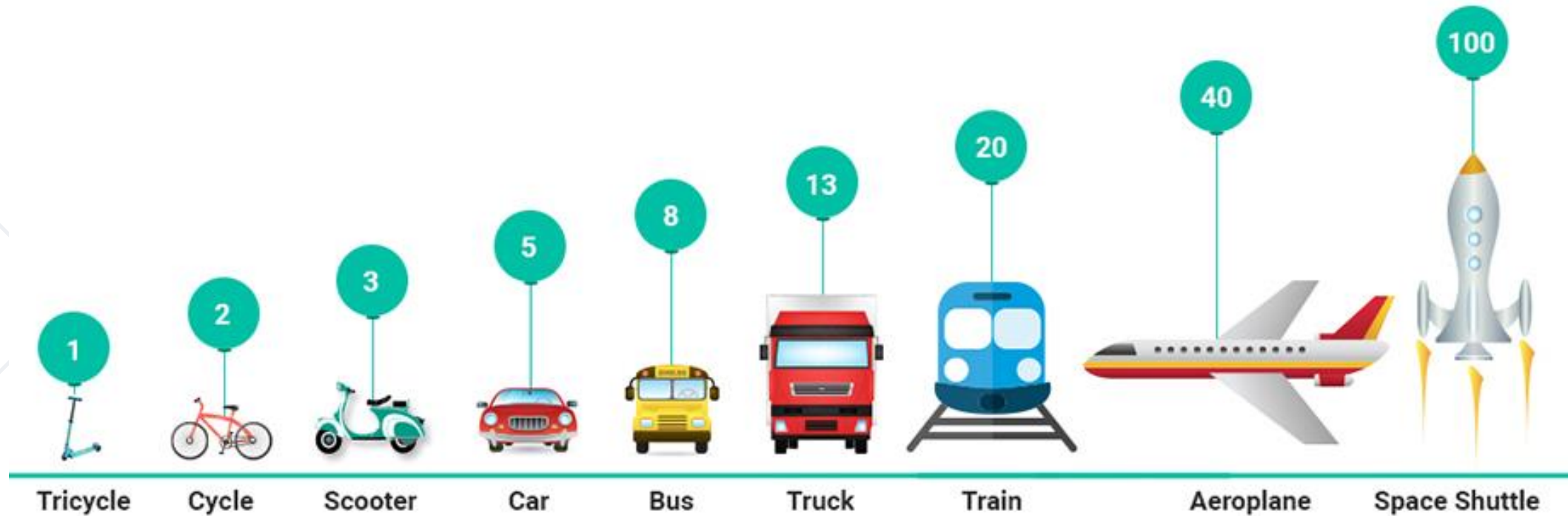
- What are "story points"?
  - In short, "story points" are the units of measure used in Scrum
  - They would express the overall effort required to complete any User Story
  - The Development Team decides on how big is the User Story
  - The sizing is done before the Sprint

# Story Points – Sizing (1)

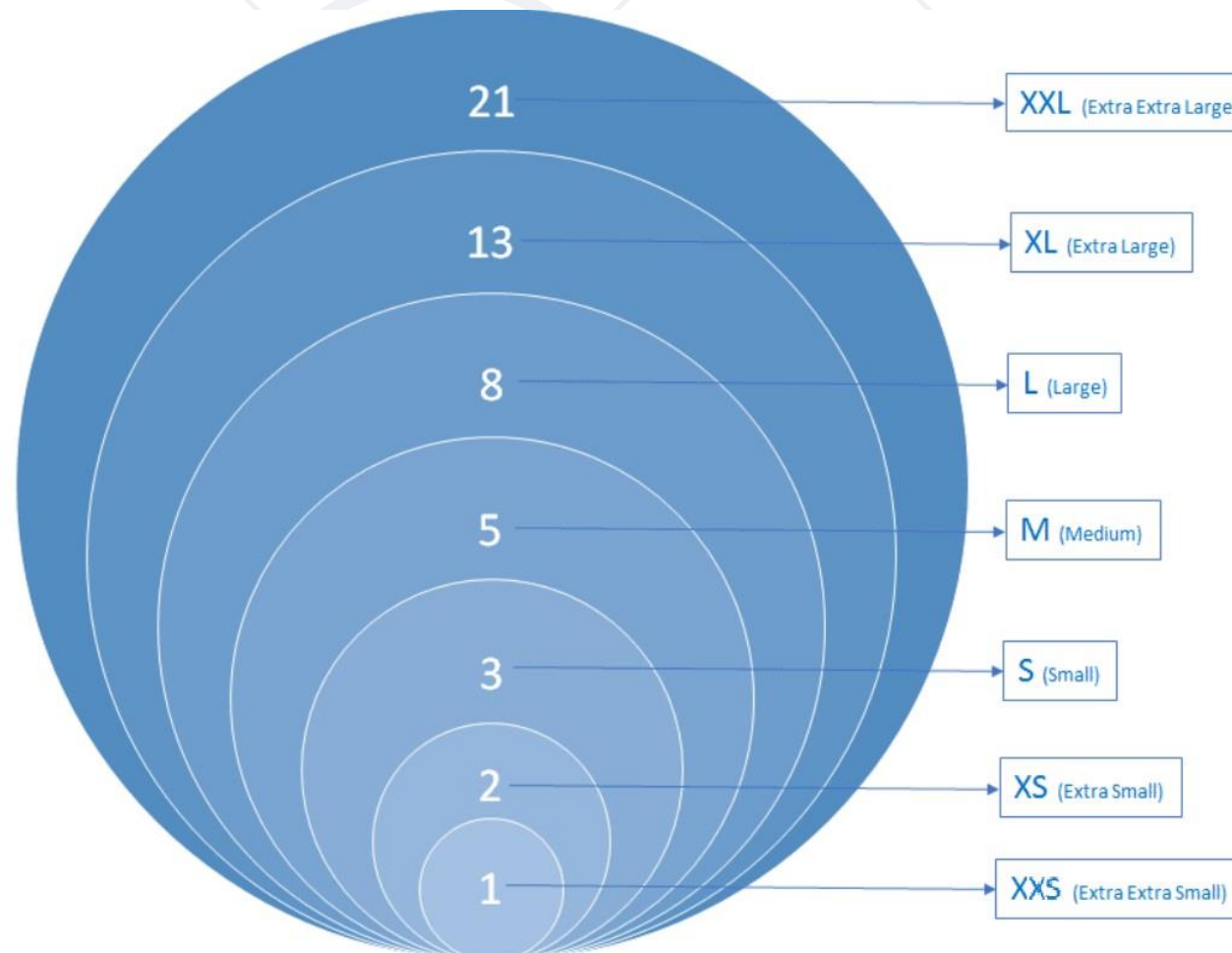
- Some best practices when creating the "story points"
  - Consider this to be a team exercise
  - Short, specific, important
  - This is an incremental work
  - Experience matters
- Planning Poker
  - One of the most recognized practices to create story points
  - Effort belongs to the Scrum Team
  - Experience matters
  - Role of Scrum Master is essential



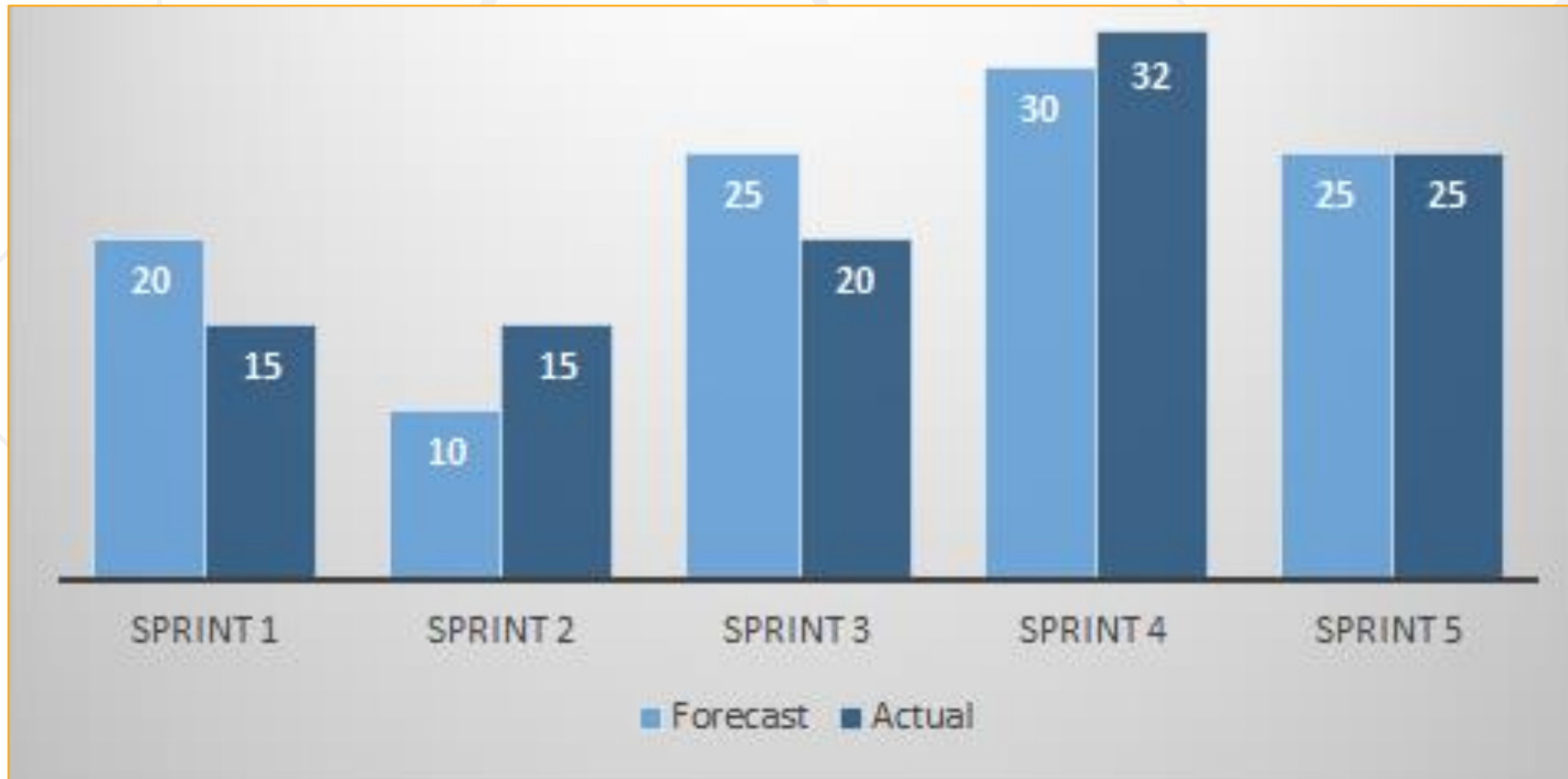
# Story Points – Sizing (2)



# Story Points – Sizing (3)



- Team's capacity for an iteration – measured in story points



- Servant Leadership
- Timeboxing
- Retrospectives
- Prioritized backlog
- Daily meetings
- Empirical and value-based measurement



# The D.E.E.P Product Backlog (*emerging*)

- A product backlog should evolve, adding new items as we learn more about the problem space
- New Stories can be added or removed based on new reality or client wishes
- SCRUM is based on Iterations (Sprints)

- "Why" prioritize?
  - It is a way to show which User Stories are with greatest importance
  - Help the team organize the Product Backlog
  - It could represent a financial value, a value for the customer, or purely technical criteria to be considered (you cannot assemble the cake before the foundations are done and cooled)

- "How" to prioritize?
  - **Kano** model: defines the **satisfiers**, the **excitement**, **indifferent** attributes, **reverse** attributes
  - **MoSCoW** model: **M**ust have, **S**hould Have, **C**ould Have, **W**on't Have
  - **Cost of Delay**: prioritization based on pure financial criteria
  - **100 Dollar Test**: similar to Monopoly game, giving a value to each US

# Sprint Backlog (1)



# The Sprint Backlog (2)

- It covers only the current Sprint
- It reflects the US-s which are selected for the Sprint only
- It is sole managed by the Development Team
- While the Product Backlog lists ALL User Stories required for the project, the Sprint Backlog would reflect only the User Stories for the current Sprint
- The Priorities, the Estimates and the number of User Stories to be included in the Sprint (in the Sprint Backlog) should be rightfully reflected

# The Sprint Backlog (3)

## 'Product Backlog' vs 'Sprint Backlog'

*The key differences between two common artifacts.*



Anything that needed to accomplish the project vision



Product Owner owns the Product backlog



Contains requirements, defects, tasks.



Everyone contributes to the product backlog



Product backlog refinement meeting is to refine the product backlog

Anything that is needed to fulfil the sprint goal.



Development team owns the sprint backlog



Subset of product backlog items defined as priority by product owner.



Only development team contributes to the sprint backlog



Sprint Planning meeting is to refine the sprint backlog items

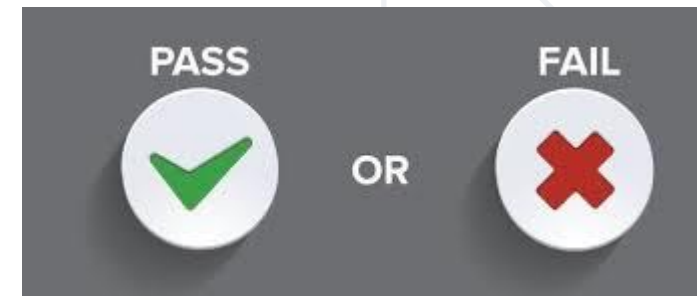


# The Sprint Backlog (4)

- Done during "**Sprint Planning**" meeting
  - The entire Scrum team participates
  - 4 hours = 2 weeks sprint
  - The meeting is held before each Sprint
- The Product Owner presents the top priority User Stories
- The Dev Team asks questions / seeks clarification
- Based on priorities, the Sprint is planned up
- The Sprint Backlog takes an amount of User Stories to utilize the upcoming Sprint

# Acceptance Criteria (1)

- Set of criteria which a User Story (US) should meet in order to be completed
- There is no set rule of **who** is defining the Acceptance Criteria
- In general, the Acceptance Criteria are initiated by the Product Owner, but a result of the work of the entire (Scrum) team
  - Why do you think is that?
- A good practice is that each US has at least one Acceptance Criteria (AC)
- There is no maximum cap of AC-s





- Another perspective – you can tell a better story when using clear set of AC-s
- Helps avoid misconception, misleading and confusion
  - An example of Acceptance Criteria

**User story:** *As a user, I want to be able to register online, so that I can start shopping online.*

**Acceptance criteria:**

- ☐ User can only submit a form by filling in all required fields
- ☐ The email user provided must not be a free email
- ☐ Submission from same IP can only be made three times within 30 minutes
- ☐ User can only submit a form by filling in all required fields
- ☐ User will receive a notification email after successfully registration

- The Definition of Done (DoD) is based on the agreed criteria that must be met for a team to consider an aspect of the product shippable or complete
- It established a shared understanding across the team what must be done for a user story to be considered finished

## DONE

adjective

1. Unit tests passed
2. Code reviewed
3. Acceptance criteria met
4. Functional Tests passed
5. Non-Functional requirements met
6. Product Owner accepts the User Story

# Definition of Done – Example

**DEFINITION OF DONE**

Team: \_\_\_\_\_  
Product: \_\_\_\_\_

As a team, before saying that an item of the sprint backlog is Done, we agree that it will meet the following:

- ☒ Code is complete and according to development team standards.
- ☒ Code refactored.
- ☒ Meet acceptance criteria.
- ☐ Code checked-in to the repository.
- ☒ Unit test written and green.
- ☐ Test coverage: \_\_ %.
- ☐ Pair programming.
- ☒ Peer review.
- ☐ Code merge and tagged.
- ☒ Deployed to the development environment.



Product Backlog



Sprint Backlog



Product Increment

The 3<sup>rd</sup> artefact

- Also referred as "potentially releasable product"
- Why is considered as "product increment"?
  - After each sprint, the product can function on its own
  - The next version is an improvement of the previous one
  - Almost a never-ending cycle

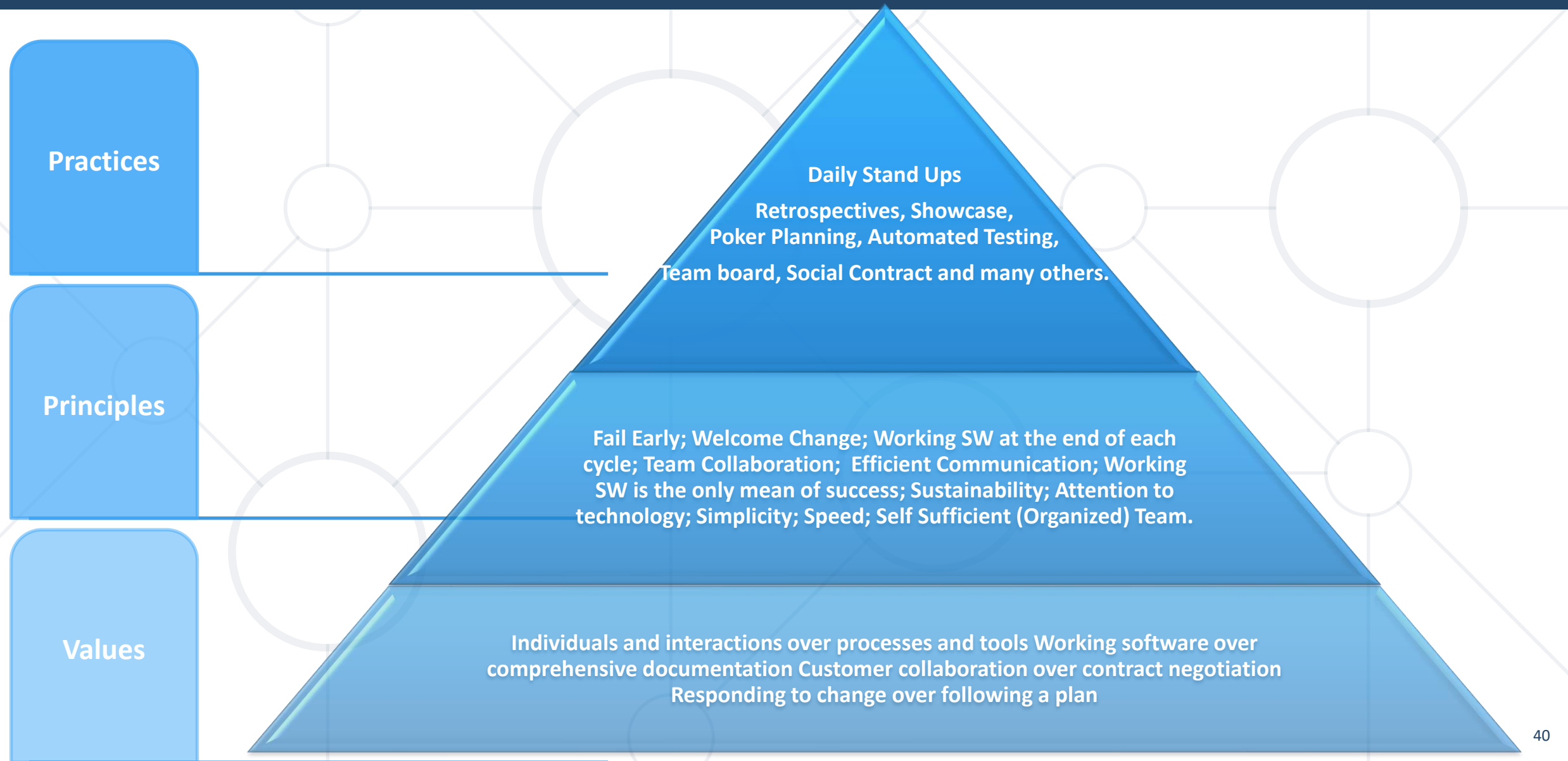
# Product Increment (2)

- Also referred as "*potentially releasable product*"
- Why is considered as "product increment"?
- The key to remember – the "product increment" must function on its own
- It is not the final version (alpha, beta, delta....).
- This makes SCRUM extremely useful for SW development

# Abbreviations

- **SM** = Scrum Master
- **PO** = Product Owner
- **Dev Team** = Development Team
- **US** = User Story
- **AC** = Acceptance Criteria
- **DoD** = Definition of Done
- Daily Scrum = Daily Stand Up = Stand Up
- Kanban Board = Wall of Work
- Iteration = Sprint = Cycle

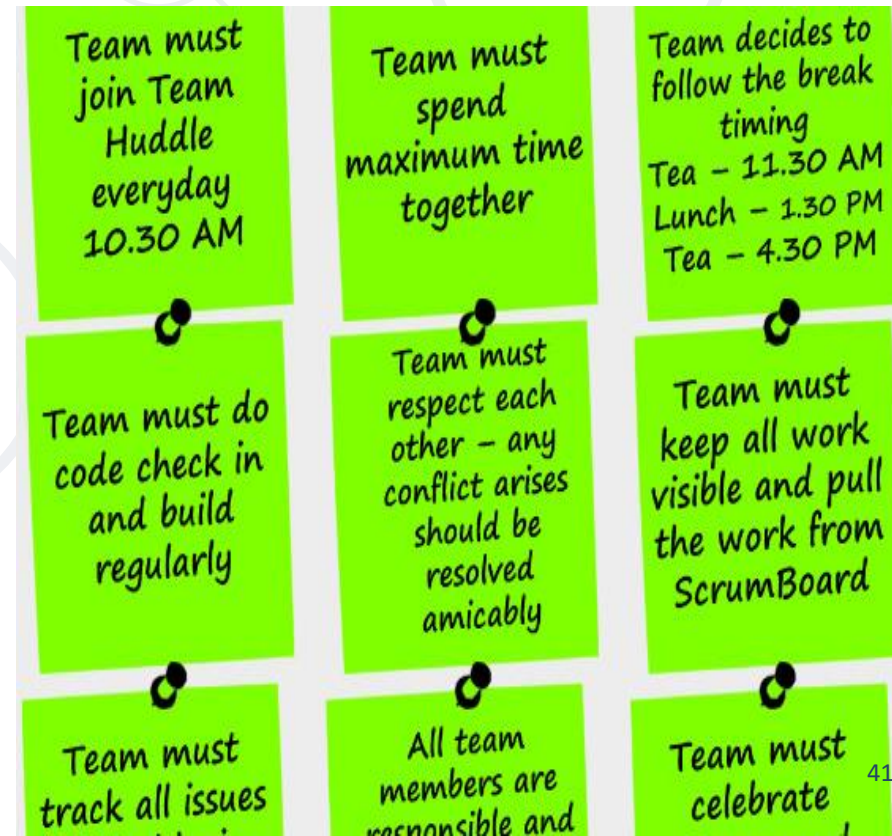
# How to Be Self-Organized?





# Social Contract (1)

- A **Social Contract** is a set of agreements that a team makes to describe how the team members will behave and work together
- It is done by the team to respect the team's needs
- Excellent practice for new teams
- Everyone contributes, SM facilitates
- It is a living document



# Social Contract (2)

- Save the environment
- Make it visible
- Follow up
- Make sure everyone contributes
- There is not one standard
- Think about what is important for the team
- Do not strive for perfection
- What about consequences?

# Social Contract – Example

## Be Available

- Work hours
- Leave
- Time away
- Sick leave
- Work from home



[www.growingagile.co.za](http://www.growingagile.co.za)



## Be On Time

- Meeting attendance
- Meeting scheduling
- Timeboxing



[www.growingagile.co.za](http://www.growingagile.co.za)



## Be Kind

- Avoid gossip
- No blame
- Have one conversation
- Honour confidentiality
- Respect culture
- Be aware of language



[www.growingagile.co.za](http://www.growingagile.co.za)



## Be Involved

- Take accountability
- Pair work
- Prefer face to face
- Store work centrally
- Share knowledge
- Have fun



[www.growingagile.co.za](http://www.growingagile.co.za)



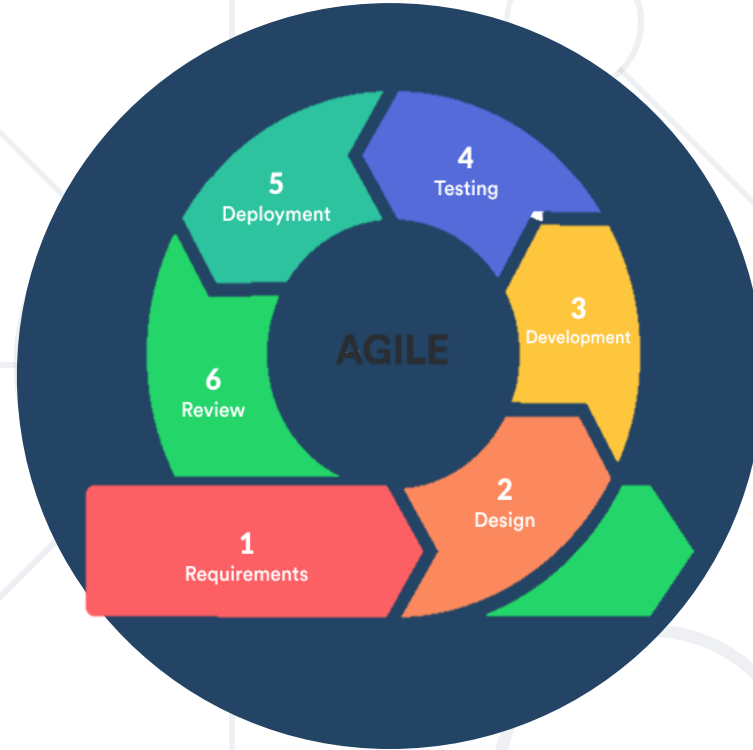
## Be Proactive

- Learn continuously
- Ask for help
- Take risks
- Respect WIP



[www.growingagile.co.za](http://www.growingagile.co.za)





# Phases in Scrum

# Scrum – What Did We Learn So Far? (1)

- Based on **Sprints/Iteration**
- The team carries a new, improved version of the product through each iteration
- It is a **lightweight framework** that helps organizations to generate value through adaptive solutions for **complex problems**
- Scrum handles three basic roles
  - The **Product Owner** (owns the business vision)
  - The **Scrum Master** (facilitator and Agile promoter)
  - The **Dev Team** (people who get things done)



# Scrum – What Did We Learn So Far? (2)

- Scrum handles three basic roles

- The Product Owner

- Holds the relationship with the customer
    - Holds the overall vision of the project
    - The commercial mind in the project

- The Scrum Master

- Main facilitator and protector of the team
    - Make certain the team delivers, using Agile methods and practices
    - Servant Leader

- The Dev Team

- The people who get the work done
    - The technical experts



Product Owner



Development Team



Scrum Master

# Scrum – What Did We Learn So Far? (3)

- Scrum uses three artifacts to help manage work

- Product Backlog

- Created and Owned by the PO
    - Available to all Stakeholders
    - Keeps the overall project objective/s
    - D.E.E.P

- Sprint Backlog

- Defines the items to be completed during the particular Sprint
    - Owned and managed by the Dev Team

- Product Increment

- Working version of the product available at the end of each iteration
    - A new, improved version of the product is available at the end of the next iteration ...and then again



Product Backlog



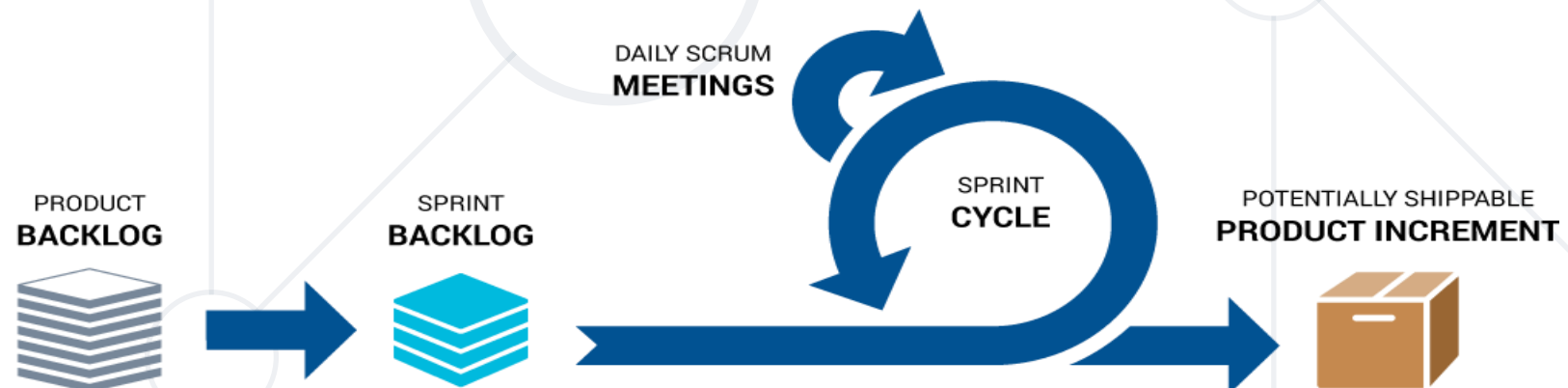
Sprint Backlog



Product Increment

# Scrum – How Does It Work?

- The **Scrum Artefacts** would give us an answer to "HOW" to build a successful project
- The **Scrum Actors** are telling us "WHO" would build it
  - Product Owner – the business mind
  - Scrum Master – the facilitator
  - The Dev Team – those who gets the things done
- The **Scrum Events (also referred to as ceremonies)** are giving us the flow of the events
  - The **Planning** phase
  - The **Delivery** phase
  - The **Closing** phase
  - ...and do it again





# SCRUM Roles and Responsibilities Game

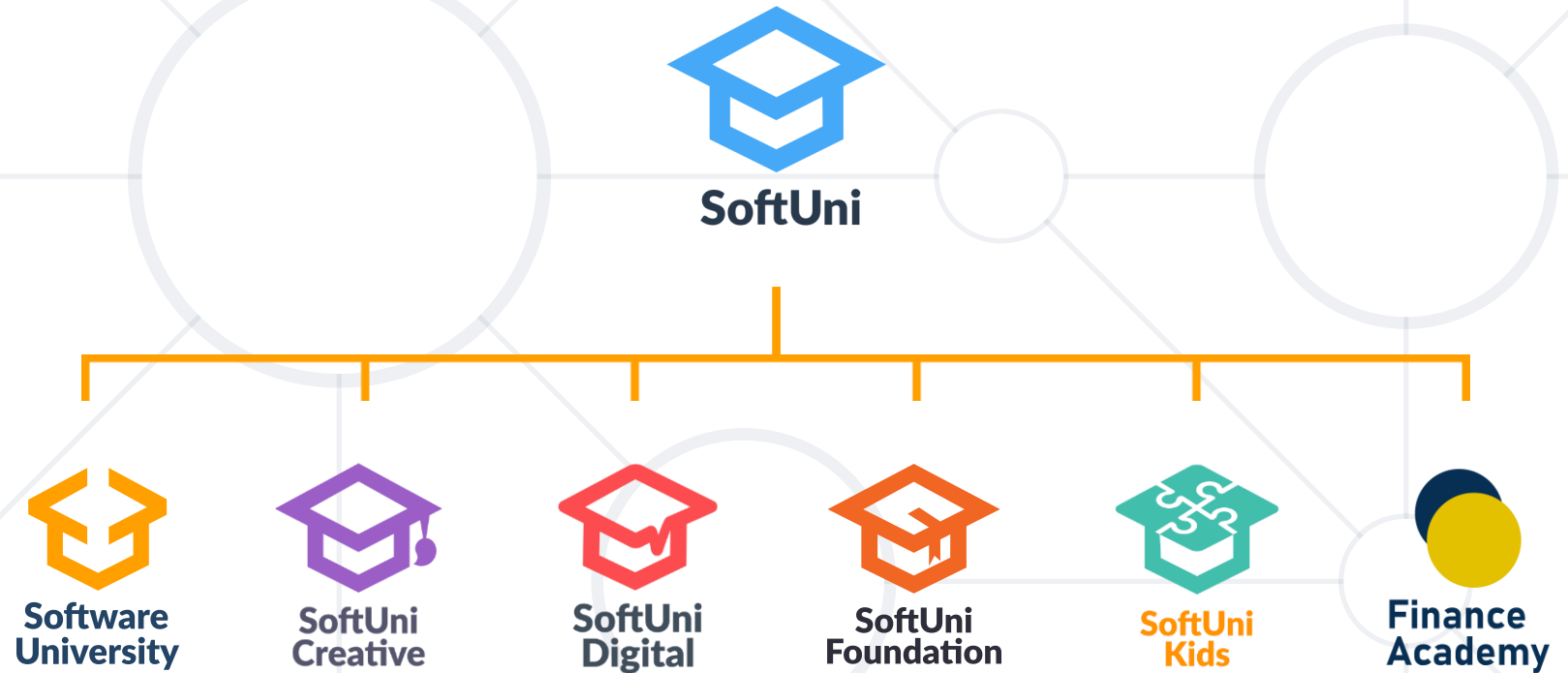


- Social contracts for agile teams
- Story points and estimation
- Product Backlog
- Everything you need to know about acceptance criteria
- Discussion topic - Can agile co-exist with waterfall?

- Scrum is a light-weight framework
- The entire team is responsible for the proper implementation of best practices
- Each Event is an irreplicable part of the process
- Skipping an Event, Phase, or an Artefact is moving away from pure Scrum



# Questions?



# SoftUni Diamond Partners

**SUPER  
HOSTING  
.BG**



**Coca-Cola HBC  
Bulgaria**

 **Flutter**<sup>TM</sup>  
International

**INDEAVR**  
Serving the high achievers



**AMBITIONED**

 **DRAFT  
KINGS**



**SOFTWARE  
GROUP**



**BOSCH**



**Postbank**

*Решения за твоето утре*

 **PHAR  
VISION**



**SmartIT**

**DXC**  
TECHNOLOGY

**createX**

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)



Software University

