3. Planning of activities considering the team's production capacity: creation of a backlog

(11.5 hours)

Summary

3.1 How to create a backlog (2.5 hours)

3.2 How to manage card granularity (2.5 hours)

3.3 How to manage capacity planning (2.5 hours)

3.4 Practical workshop – Agile adoption: building a board and writing cards (4 hours)

3.1 How to create a backlog (2.5 hours)

What is a backlog?

- A **prioritized list** of features, enhancements, bug fixes, tasks, etc.
- Owned and maintained by the **Product Owner**
- Evolving document – it is never complete
- Provides a single source of requirements for any changes to be made to the product

Components of a backlog item

- **Title** – Brief summary
- **Description** – Detailed explanation or user story
- **Acceptance Criteria** – Conditions to consider the story "done."
- **Priority** – Order of importance
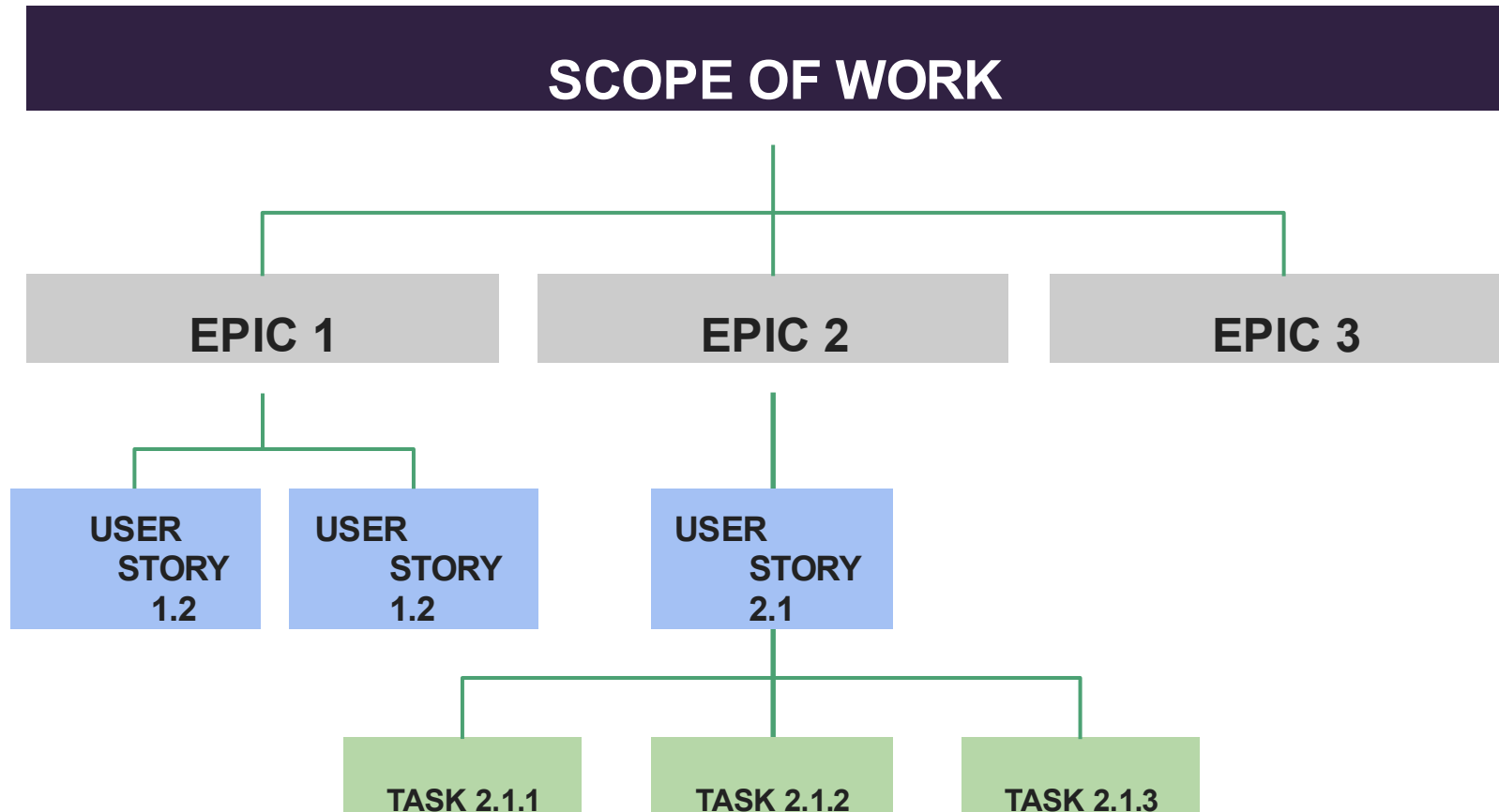- **Estimation** – Effort required (e.g., story points, hours)

How to create a backlog

TIP: You choose the granularity of the cards on your board. You can decide what is an 'Epic' and what is a 'Story'.

Example:

- SCOPE OF WORK: tender
- EPIC: Technical documentation; Economic documentation; Administrative documentation
- STORY: Completion of documents; Drafting of estimates
- TASK: Completing Annex 1 and Annex 2, Creating economic simulation files

How to create a backlog

```
                    ┌─────────────────────────────────────────┐
                    │            SCOPE OF WORK                 │
                    └─────────────────────────────────────────┘
                                      │
         ┌────────────────────────────┼────────────────────────────┐
    ┌─────────┐                 ┌─────────┐                   ┌─────────┐
    │ EPIC 1  │                 │ EPIC 2  │                   │ EPIC 3  │
    └─────────┘                 └─────────┘                   └─────────┘
         │                           │
    ┌────┴────┐                      │
┌────────┐ ┌────────┐           ┌────────┐
│ USER   │ │ USER   │           │ USER   │
│ STORY  │ │ STORY  │           │ STORY  │
│ 1.2    │ │ 1.2    │           │ 2.1    │
└────────┘ └────────┘           └────────┘
                                     │
                       ┌─────────────┼─────────────┐
                  ┌──────────┐  ┌──────────┐  ┌──────────┐
                  │TASK 2.1.1│  │TASK 2.1.2│  │TASK 2.1.3│
                  └──────────┘  └──────────┘  └──────────┘
```

How to create a backlog

1. Start with high-level goals – What problems are we solving?
2. Write user stories – Use the format: "As a [user], I want [feature] so that [benefit]."
3. Break down epics into smaller stories.
4. Prioritize based on business value and urgency.
5. Estimate effort with the team.
6. Review and refine regularly (Backlog Grooming/Refinement).

How to create a backlog – Common mistakes to avoid

- Having vague or overly detailed stories
- Not involving the team in backlog refinement
- Ignoring technical debt or bugs
- Letting the backlog grow without pruning

Group Exercise – Create a mini product backlog
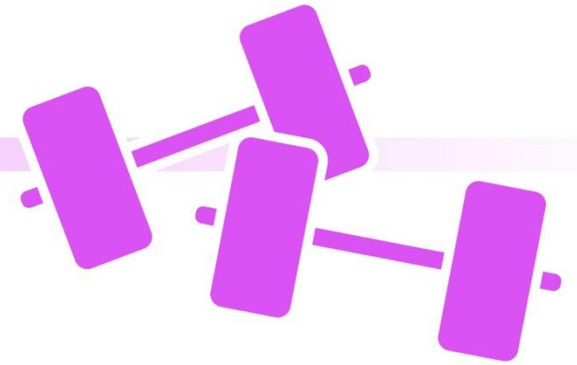
Scenario:

EduTrack is an initiative to create a web-based classroom management platform designed to simplify
administrative tasks for teachers, improve student engagement, and provide real-time insights into academic progress.

The platform includes features like attendance tracking, assignment submissions, grade management, and student-teacher communication.

The web solution must be intuitive, scalable, and secure, focusing on user-friendly design and continuous adaptability through iterative feedback from teachers and students.

Group Exercise – Create a mini product backlog
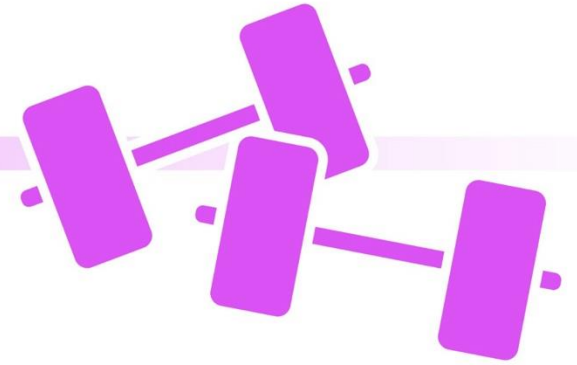
Exercise:
Think of 5 features or needs students/teachers might have.

Instructions:

- In small groups, write 5 user stories using the format:
  "As a [student, teachers], I want [feature] so that [benefit]."

- Assign a priority (1–5) and a rough estimate (e.g., 1–3 points).

- Present your mini backlog to the class.

Example:

As a student, I want to create a to-do list so that I can track my assignments.
Priority: 1 – Estimate: 2 points

How to create a backlog

At the beginning, building the backlog will be very difficult!

Sometimes you build cards that are too 'big' (you realise this when a card remains in the 'doing' state for too long)…

…other times you go into too much detail (you can tell if a card transits the board for too short a time)

Tip: divide the cards so that they never last more than 4 hours, i.e. half the working day. You will move more cards and be more motivated!

To what extent should the content of the cards be detailed?

3.2 How to manage card granularity (2.5 hours)

Different granularities

SCOPE OF WORK

SCOPE OF WORK

EPICS

USER STORIES

TASKS

USER STORIES

TASKS

WHAT ENDS UP IN THE 'BACKLOG'
COLUMN OF THE BOARD
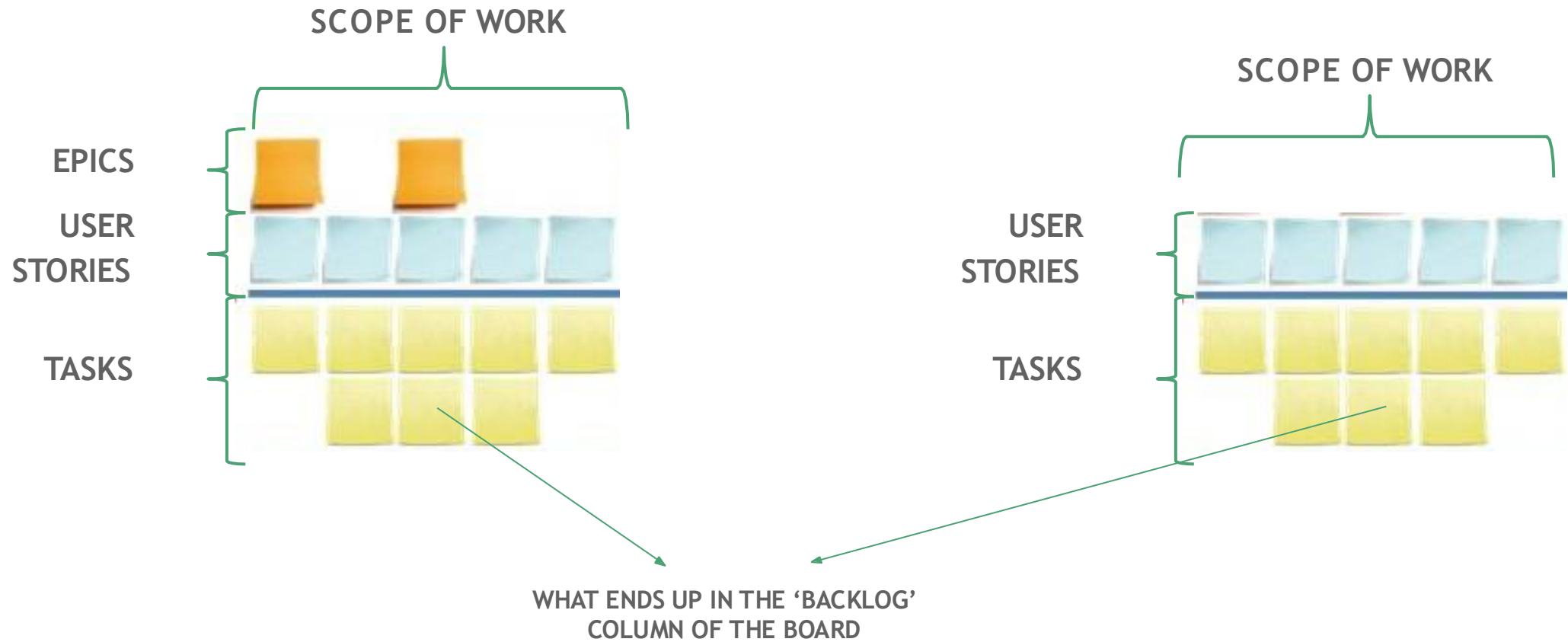
WBS vs RBS

**WORK BREAKDOWN STRUCTURE
(Waterfall)**

Breakdown of the project
from the WORK to be performed
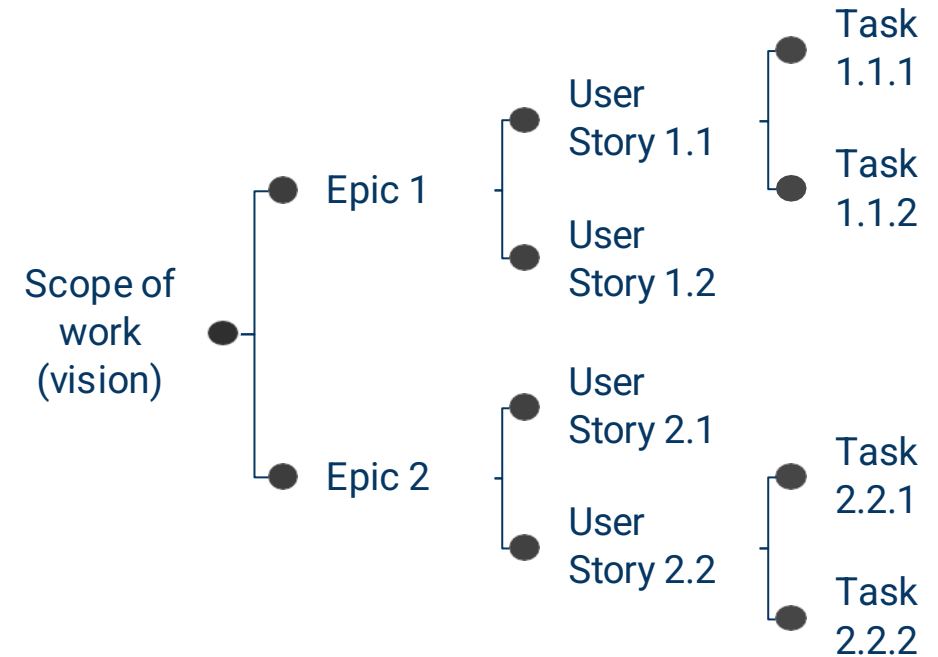
**REQUIREMENTS / FEATURE
BREAKDOWN STRUCTURE
(Agile)**

Breakdown of the project
from REQUIREMENTS /
FEATURES
of the OUTPUT

WBS vs RBS

**WORK BREAKDOWN STRUCTURE
(Waterfall)**

**REQUIREMENTS / FEATURE
BREAKDOWN STRUCTURE
(Agile)**

Scope of work
- Phase 1
  - Task 1.1
  - Task 1.2
- Phase 2
  - Task 2.1
  - Task 2.2
- Phase 3
  - Task 3.1
  - Task 3.2

Scope of work (vision)
- Epic 1
  - User Story 1.1
    - Task 1.1.1
    - Task 1.1.2
  - User Story 1.2
- Epic 2
  - User Story 2.1
  - User Story 2.2
    - Task 2.2.1
    - Task 2.2.2
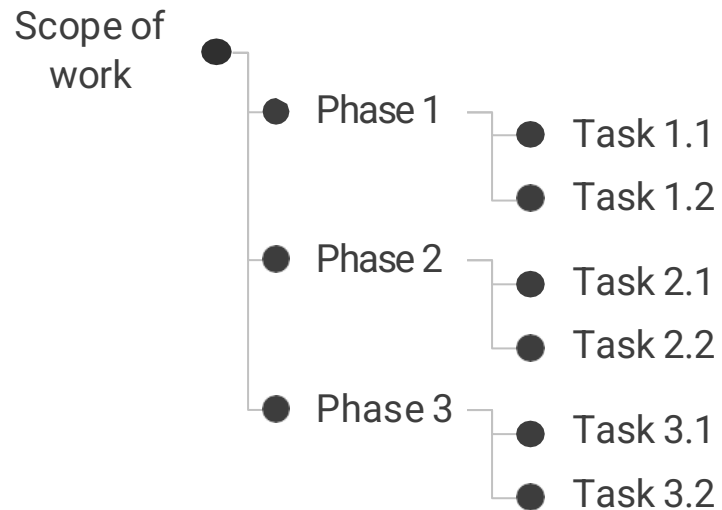
Feature vs Epic

- Feature: describes a solution and/or answer to a given requirement

- Epic: large body of work that can be broken down into a series of smaller stories

Epic → User Stories

Big, coarse-grained, sketchy

Small, detailed, specific
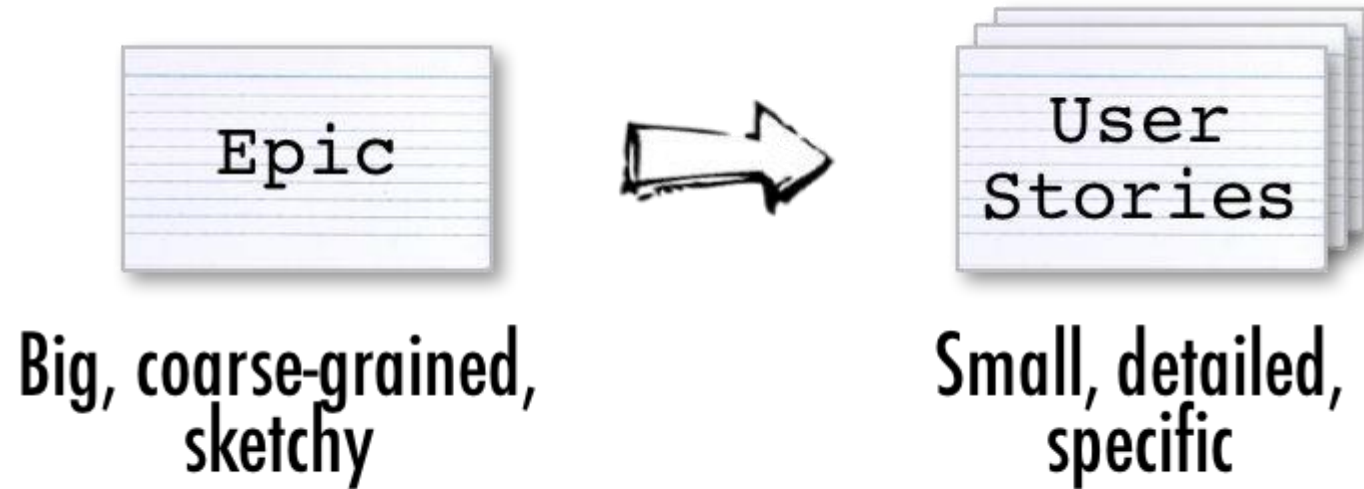
Epic vs User Story

Each Epic is an expression of a requirement and is very similar to a story (but bigger and more complex)...

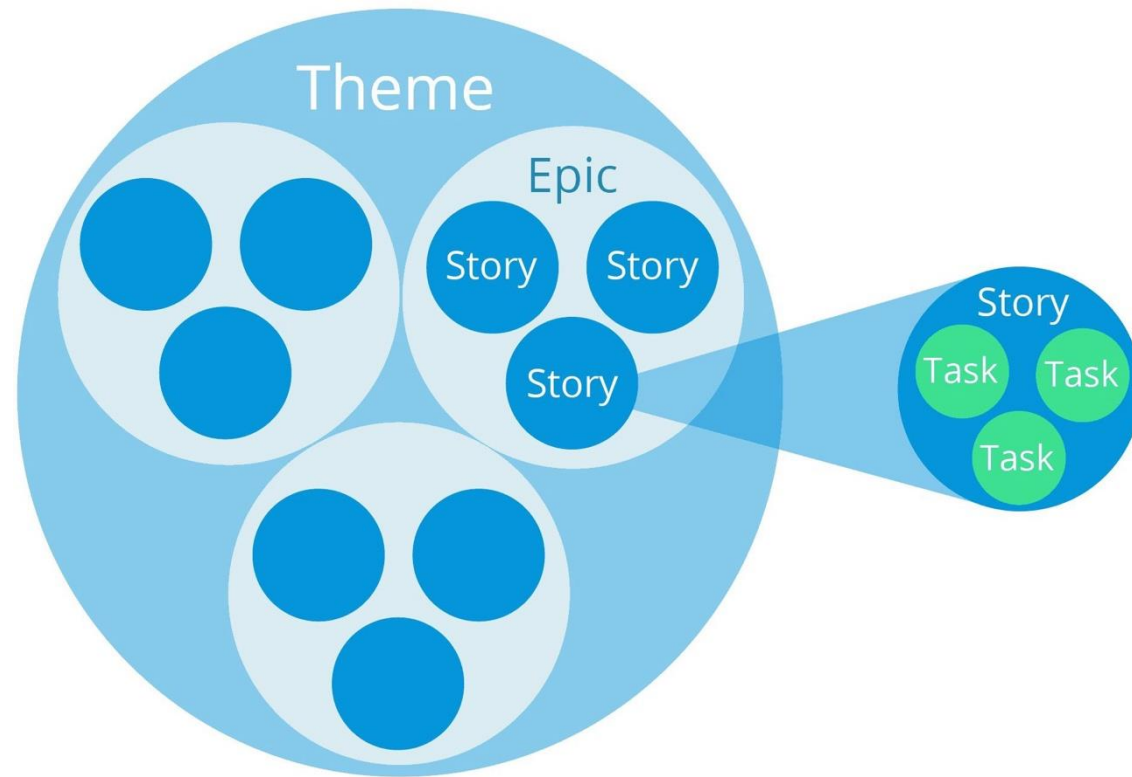...and serves to structure and group requirements to better manage the backlog.

User Story: the most efficient way of defining expectations about new functionalities and/or improvements regarding a certain output

Task: a unit of work needed to realise a user story
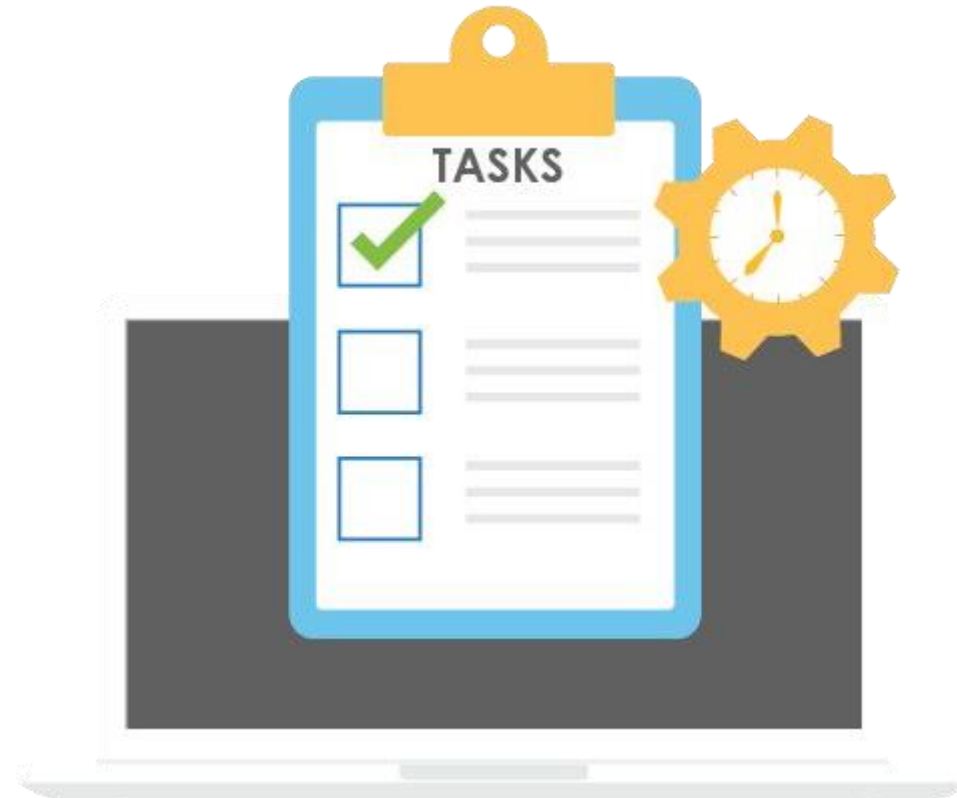
Relations between different cards

Three good reasons to use RBS

1) Allows the team and the client to communicate using mutually understandable terms

2) Allows prioritisation of work based on the customer's perceived value

3) Keeps track of work against actual value produced

Moscow Method

M MUST HAVE
Project cannot do without them

S SHOULD HAVE
Must-haves over the long run

C COULD HAVE
Low-cost tweaking

W WON'T HAVE
Get back to them at better days

How to create a backlog – Tips

When creating backlogs, start with the largest features and then break them down over time

This allows the customer not to go into detail until it is really necessary

Distinguishing cards

The simplest and most direct method is to create and use 'labels' to distinguish cards already in the creation phase.

There are also power-apps useful for this purpose. One of these is Screenful for Trello. How does it work?

Screenful's Epic allows you to:

1) Indicate which lists contain epic

    E.g.: we might decide that in the backlog we put all cards of type "Epic"

2) Assigning tasks to each epic
    NB: First I have to create the task (e.g. in To Do) and then assign it to the Epic in the backlog

3) Defining a syntetich version of the RBS

This makes it possible to observe the number of tasks per epic (≈ RBS) and monitor progress for each epic.

3.3 How to manage capacity planning (2.5 hours)

Managing capacity planning

Our normal customer demands the maximum in the shortest possible time, with the twofold result

- Capacity saturation

- Complex releases

Factors that very often result in:

- Frustrated teams

- Increasing time to market

So what to do?

1. Giving weight to outcomes
2. Define a release plan

Giving weight to outcomes

- How important are features (CARDs) to each other?
- Which of them are most likely to achieve the desired OUTCOME value?

Highest Value

High Value

Page 1

Valuable
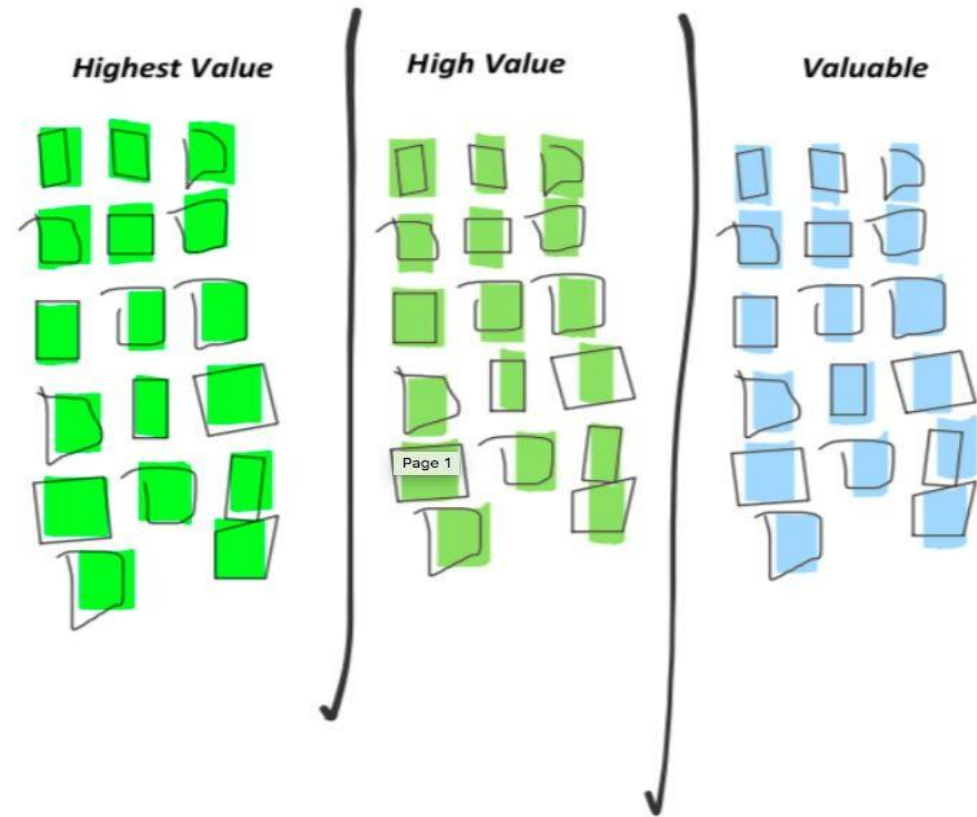
Giving weight to outcomes

- How much do the most important features weigh individually?
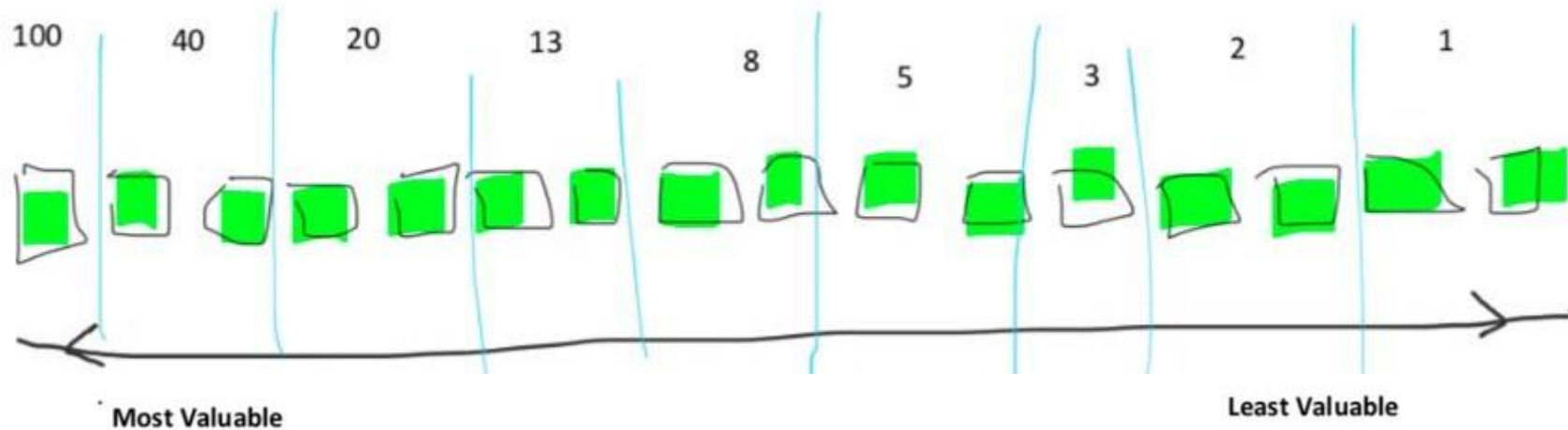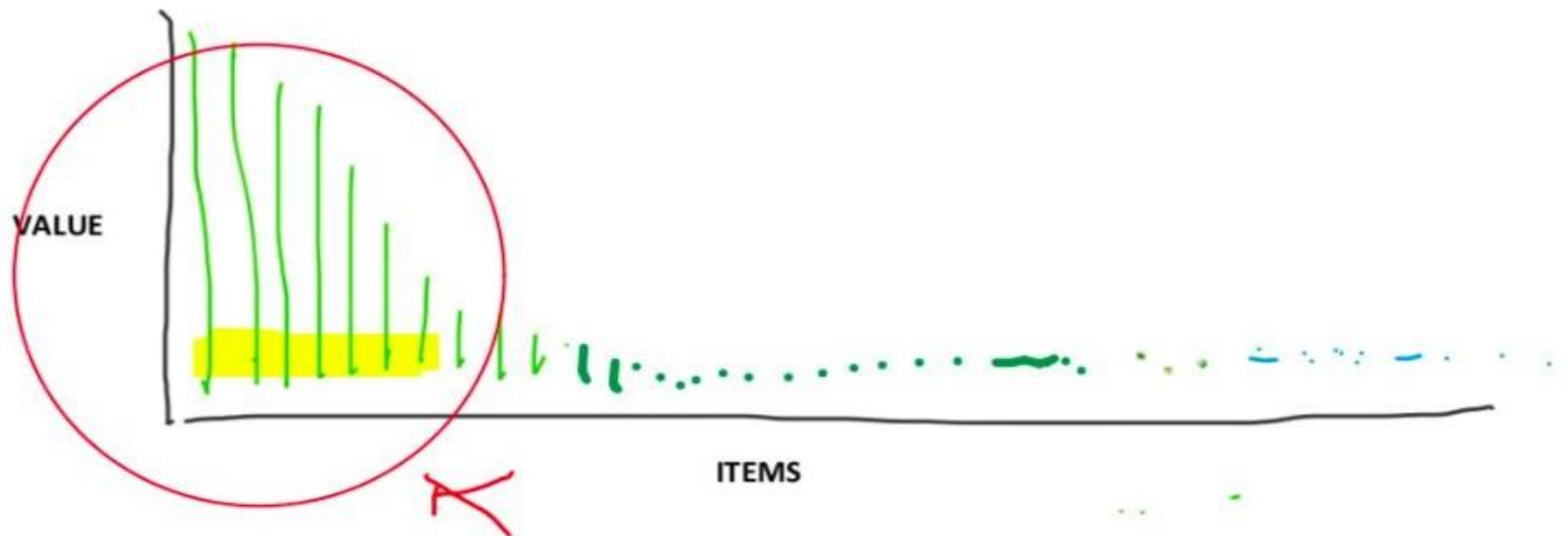

Note: let the team make the estimates!

Giving weight to outcomes

• What is the actual production capacity available?
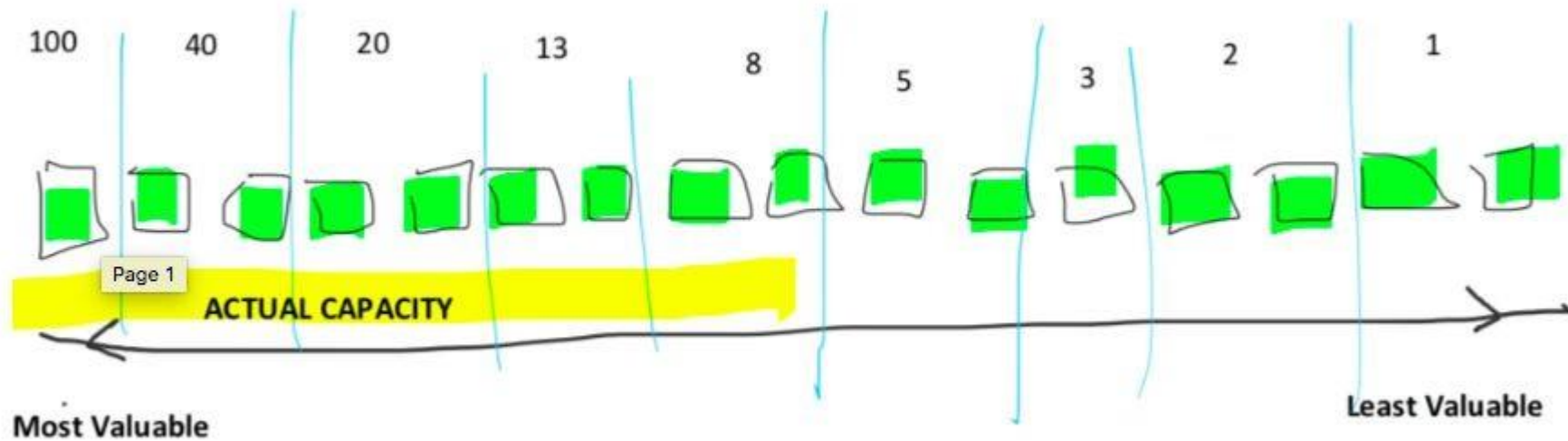
Giving weight to outcomes

- And with a Pareto diagram [representation of the degree of influence of certain variables on a phenomenon] we get that...

Establishing a release plan

Release Plan: set of activities aimed at transforming requests into tested and functioning outputs

How to proportion the backlog according to the team's production capacity?

Capacity planning: optimising productivity by managing available resources and skills needed for the Sprint

Example: a team of 6 people is asked to complete a series of activities estimated at 200 hours in 5 days

1. Activity weight: 200h
2. Sprint duration: 5 days
3. Duration of working day: 8h
4. Team size: 6 persons

Total working capacity: 5*8*6 = 240h

But why then do we often end up in overload?

Some of the factors limiting capacity:

1. Uneven teams (different skills)
2. Several projects at the same time
3. Bottlenecks (e.g. validation)
4. Emerging activities (tickets)

So what to do?

Some of the factors limiting capacity:

1. Uneven teams (different skills)
2. Several projects at the same time
3. Bottlenecks (e.g. validation)
4. Emerging activities (tickets)

So what to do?

1. Define the team configuration in advance
2. Limiting the number od open contexts
3. Foresee validation times
4. Establishing a Focus Factor

Focus Factor (FF): estimation of how focused the team is.

Average = 70%

A lower FF may mean that the team anticipates many disturbing factors or assumes that they have made over-optimistic estimates.

How to calculate a reasonable Focus Factor

The best way is based on the performance of the last sprint (or the average of the most recent sprints)?

Calculation of the Focus Factor

FFof the last sprint = Effective speed/Available man days

Effective speed: Sum of the weight of all stories completed in the last sprint.

Once the weight of the cards and the focus factor have been defined, we can define the Release Plan.

Exercise

- Activity weight: 200h
- Sprint duration: 5 days
- Duration of working day: 8h
- Team size: 6 persons
- Focus Factor: 0.7

What is the **effective working capacity?**
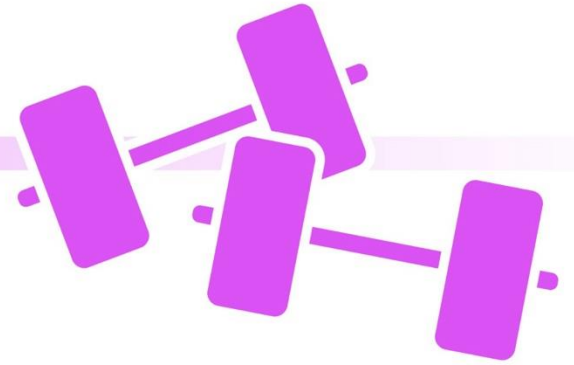
What have we learnt?

1. Renegotiating EFFORT based on the SHARED UNDERSTANDING of VALUE (outcome-oriented) and according to the PRODUCTIVITY OF THE TEAM

2. Assess the EFFECTIVE IMPACT of any ADDITIONAL RESOURCES

3. Give PRIORITY to the backlog through a COHERENT RELEASE PLAN

4. Understand if and when to add any EMERGING FEATURES to the backlog

1. Renegotiating EFFORT based on the SHARED UNDERSTANDING of VALUE (outcome-oriented) and according to the PRODUCTIVITY OF THE TEAM

2. Assess the EFFECTIVE IMPACT of any ADDITIONAL RESOURCES

3. Give PRIORITY to the backlog through a COHERENT RELEASE PLAN

4. Understand if and when to add any EMERGING FEATURES to the backlog

2.5 Practical workshop – Agile adoption: building a board and writing cards (4 hours)

Practical workshop

Scenario:

EduTrack is an initiative to create a web-based classroom management platform designed to simplify
administrative tasks for teachers, improve student engagement, and provide real-time insights into academic progress.

The platform includes features like attendance tracking, assignment submissions, grade management, and student-teacher communication.

The web solution must be intuitive, scalable, and secure, focusing on user-friendly design and continuous adaptability through iterative feedback from teachers and students.

Exercise:
Now you're ready to try to set up an Agile board!

Based on the EduTrack scenario:

• Define workflows

• Write user stories and tasks on cards

• Manage work effectively

NOTE: you can do this either in physical form, using paper sheets and markers, or in digital form (recommended option), using the Miro board found **here**.