Agile Project Management

ISEN Master 1 - 2025

Who am I?

- Juliette WATINE
  - Cursus universitaire
    - HEI – Promo 2005
    - Concordia University, Montréal, Canada
  - Jobs:
    - 2005-2007: ALTEN, Brussels, Belgium
    - 2007-2010: EUROCLEAR, Brussels, Belgium
    - 2010 -2015: ID KIDS GROUP, Roubaix, France
    - 2015-2018: CREDIT MUTUEL NORD EUROPE, Lille, France
    - 2018-2023: ONEY, Croix, France
    - Since 2023: DECATHLON, Croix France

To contact me: Juliette.watine@junia.com

Introduction to Agile project management and discovery of Scrum and Kanban frameworks

Module 1

Goals of Module 1

1. Understand the fundamentals of agility.
2. Learn about the roles, events, and artifacts of the Scrum framework.
3. Introduce Kanban as an alternative or complement to Scrum.
4. Learn how to write a Product Backlog and User Stories.

Agenda Theoretical Part (2h)

Introduction to agility:

- History and origin of agility.
- The principles of the Agile Manifesto.
- Comparison between traditional project management and agile methods.

Overview of the Scrum framework:

- Roles: Product Owner, Scrum Master, Development Team
- Events: Sprint, Daily Scrum, Sprint Review, Sprint Retrospective.
- Artifacts: Product Backlog, Sprint Backlog, Increment.

Introduction to Kanban:

- Origins and principles of Kanban.
- Differences and complementarities with Scrum.
- Key practices: Visualization of the workflow, Limitation of WIP (Work In Progress), Flow management.

User Stories and the Product Backlog:

- Definition of User Stories.
- Techniques for writing User Stories (INVEST, 3C).
- Prioritization of the Product Backlog.

Introduction to agility

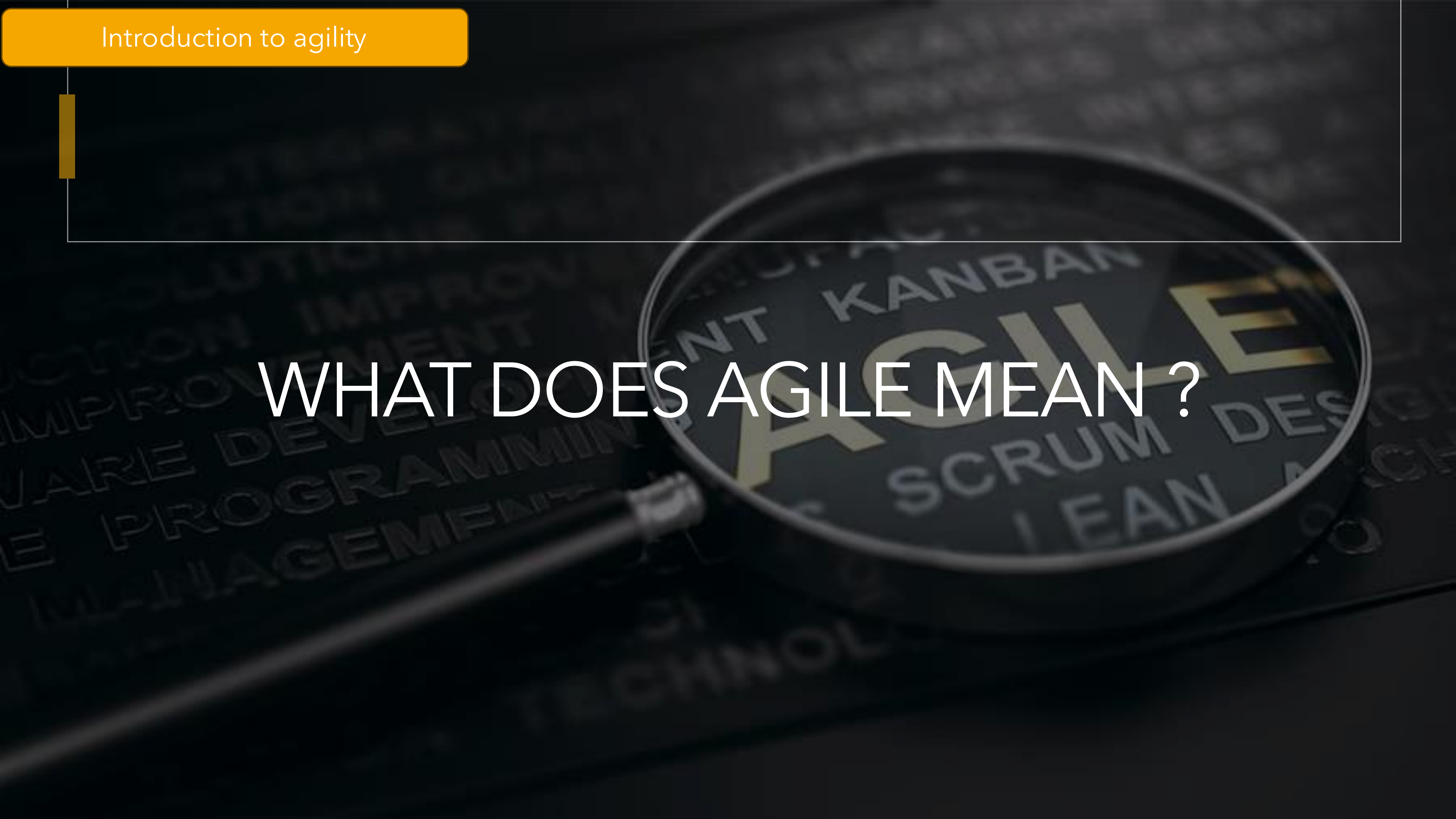
Part 1

Introduction to agility

History and origin of agility.

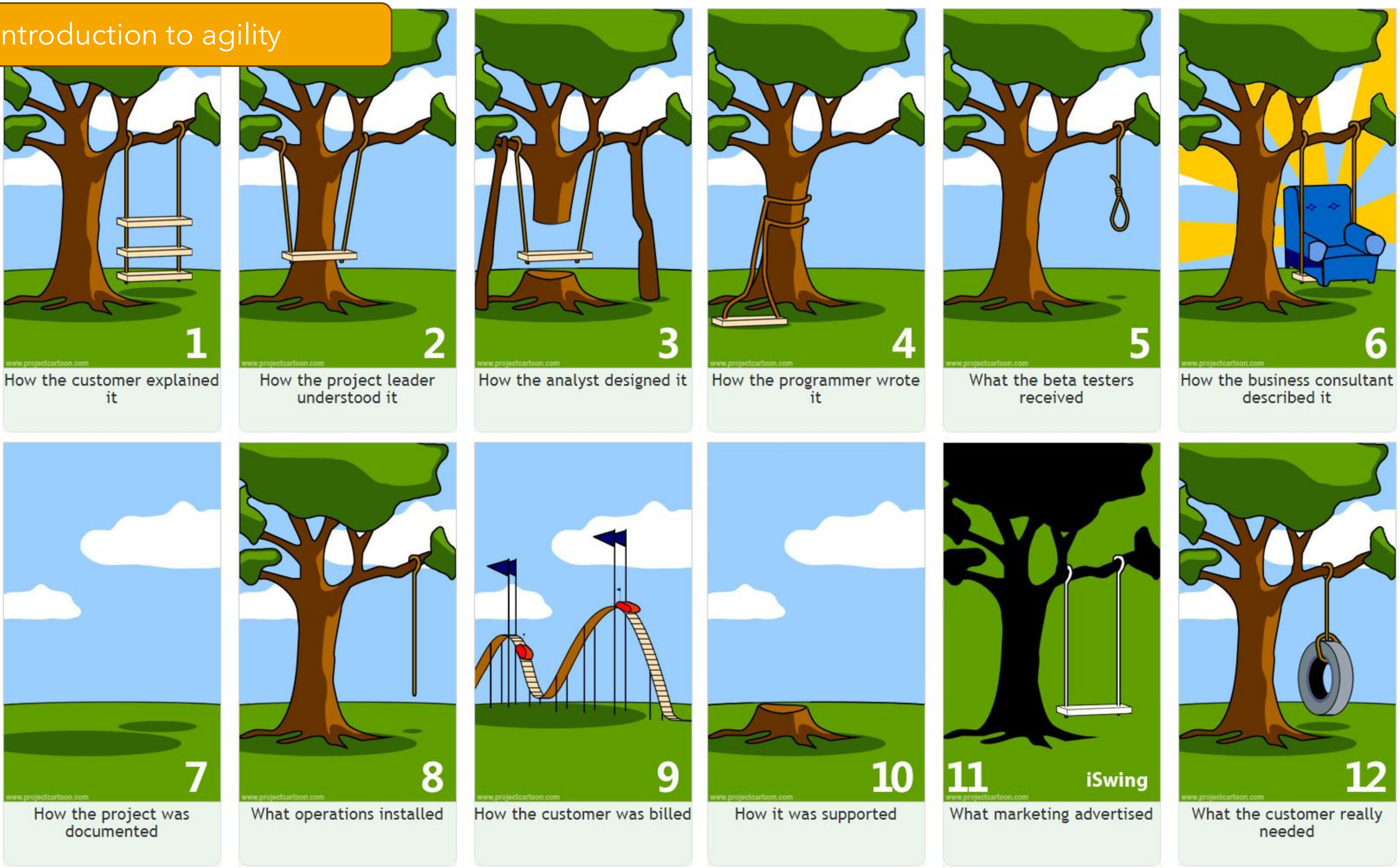
The principles of the Agile Manifesto.

Comparison between traditional project management and agile methods.

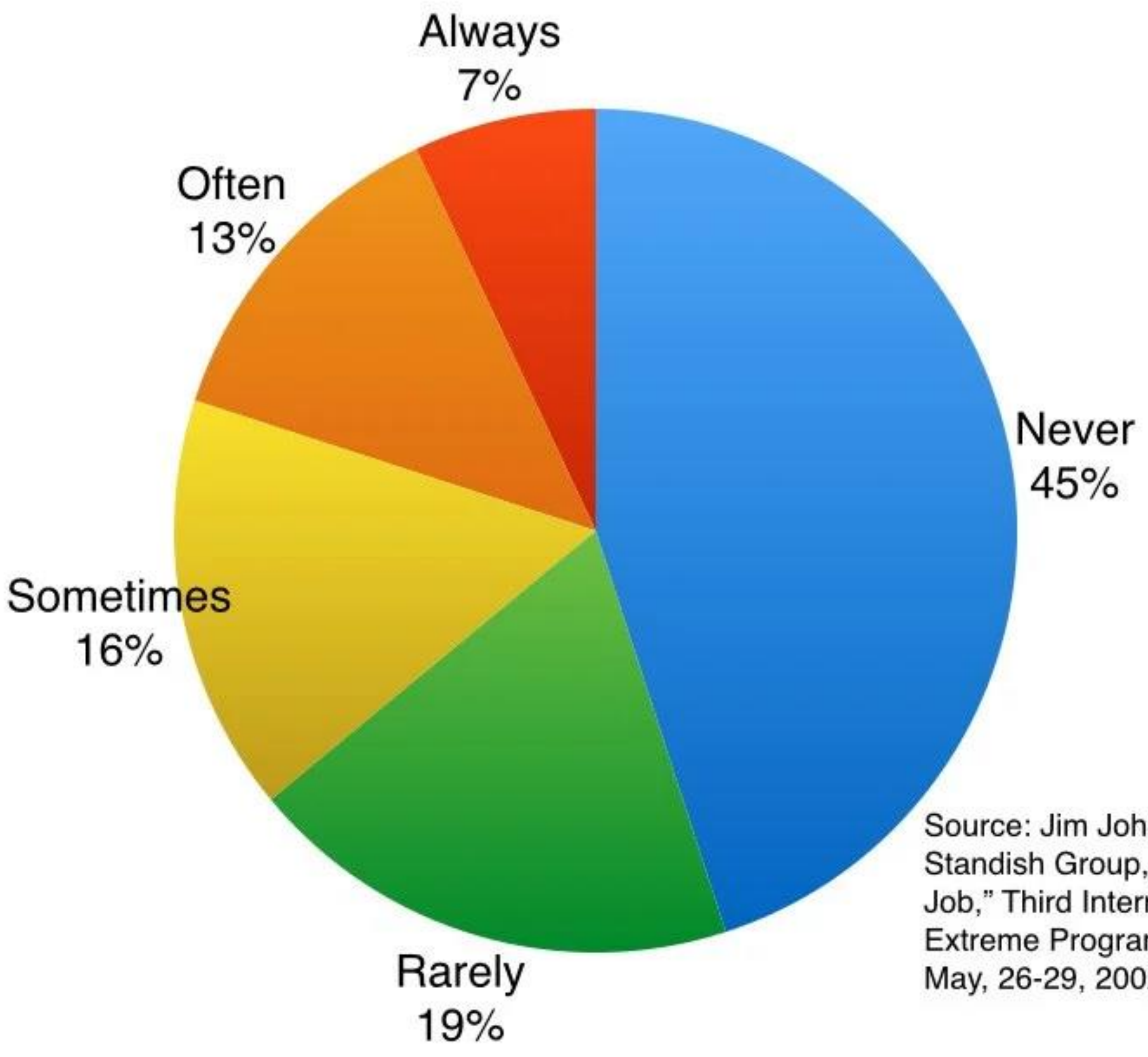


WHAT DOES AGILE MEAN ?

Introduction to agility

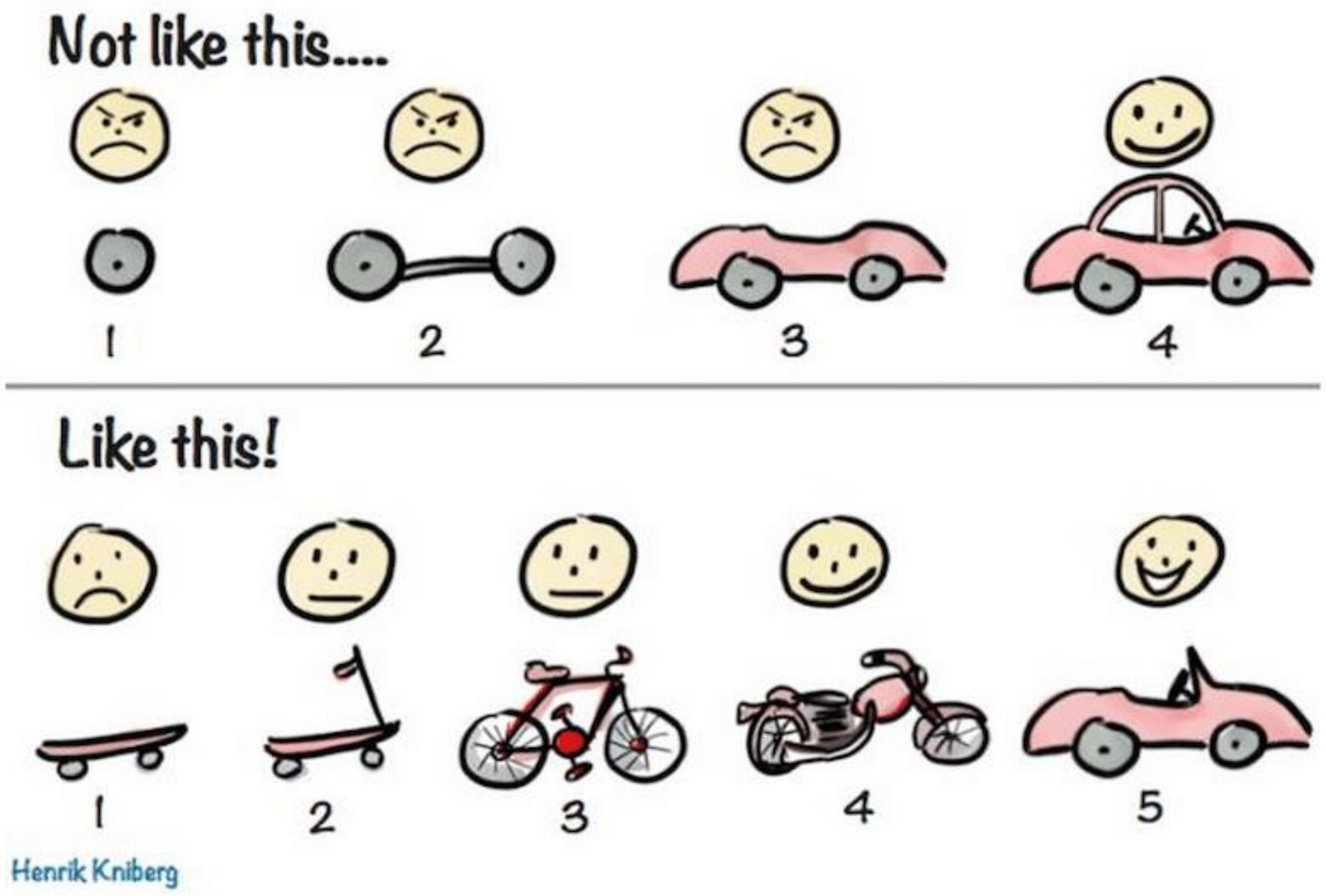

1
How the customer explained it
2
How the project leader understood it
3
How the analyst designed it
4
How the programmer wrote it
5
What the beta testers received
6
How the business consultant described it
7
How the project was documented
8
What operations installed
9
How the customer was billed
10
How it was supported
11
iSwing
What marketing advertised
12
What the customer really needed
www.projectcartoon.com

Feature Use in Four Internal-Use Products

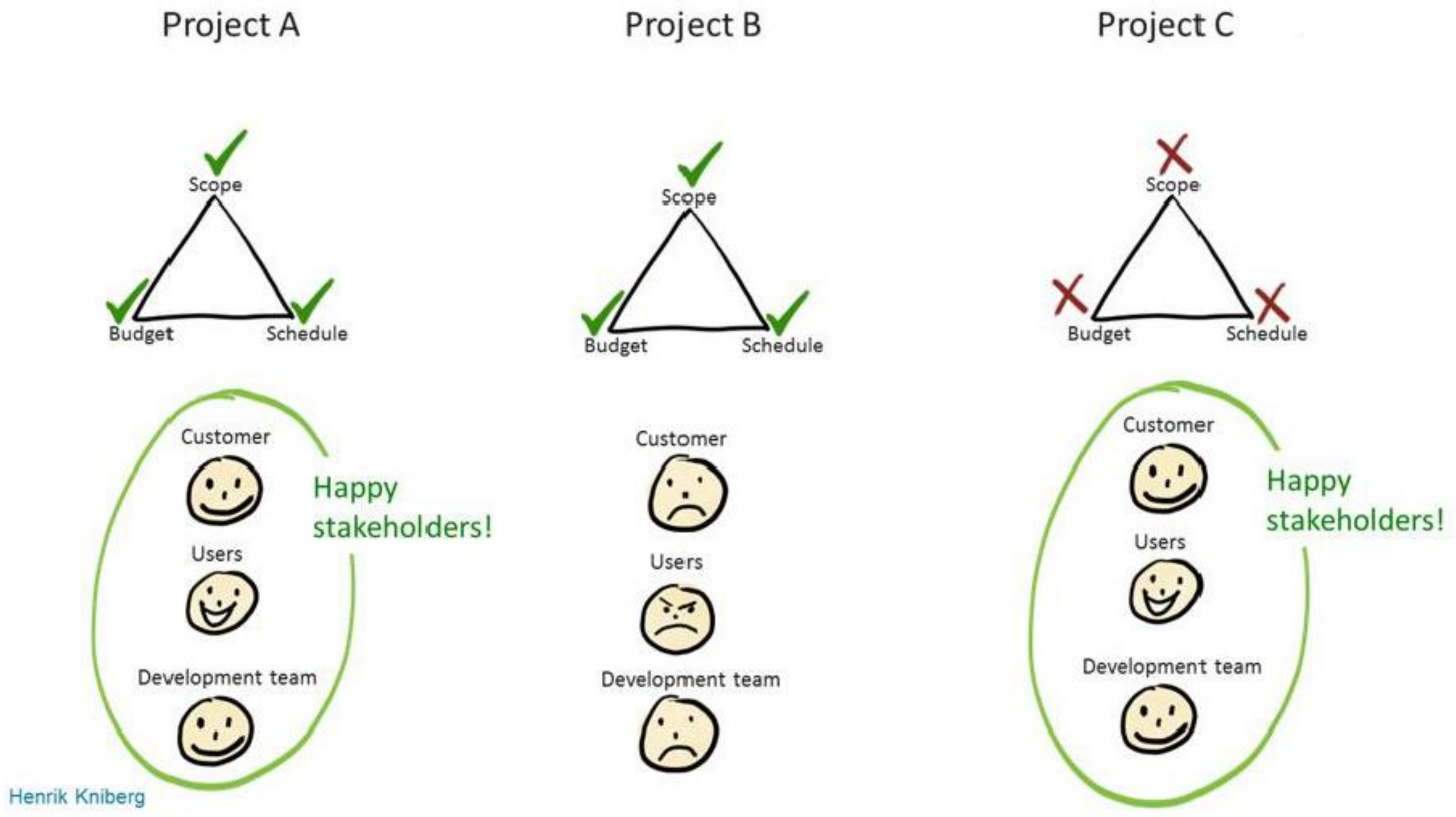


Source: Jim Johnson, Chairman of The Standish Group, Keynote "ROI, It's Your Job," Third International Conference on Extreme Programming, Alghero, Italy, May, 26-29, 2002.

Introduction to agility



Introduction to agility



So what is agile? And what is not agile?

It's not just for IT

It's not speed

It's not a miraculous method

It's not a solution when the objective is not clearly defined

It's not a big fiesta or the chaos

It's not just post-it

It's not a simple recipe

What is Agile project management?

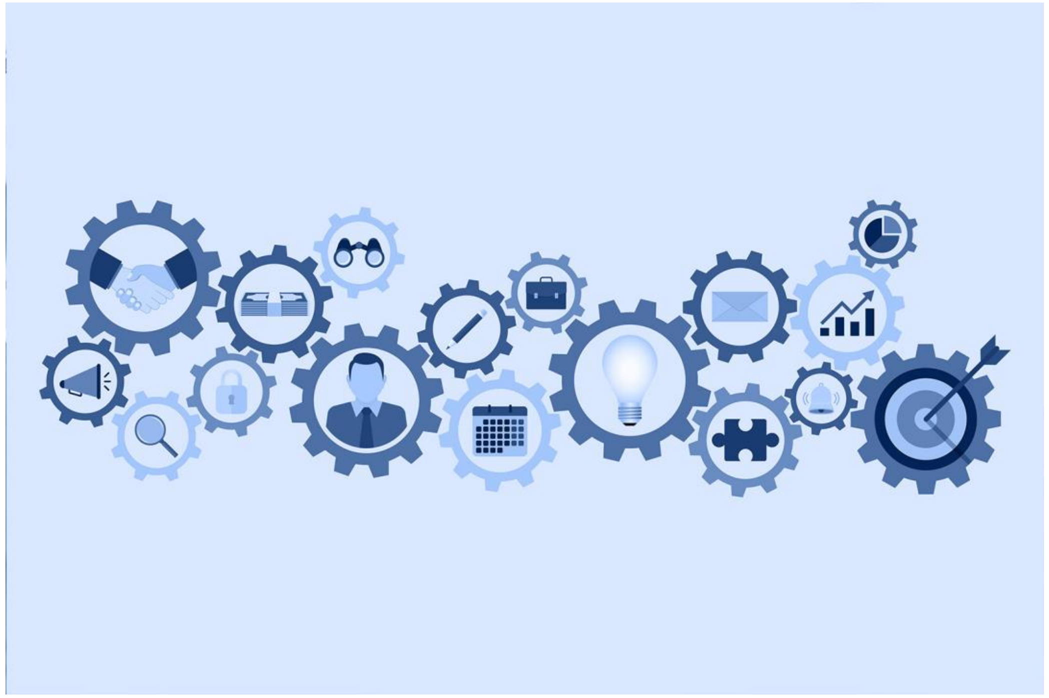
- Agile project management is an **iterative** approach to manage software development projects that focuses on **continuous releases** and incorporating **customer** feedback with every iteration.
- Software teams that embrace agile project management methodologies increase their development **speed**, expand **collaboration**, and foster the ability to better respond to market **trends**.

Example of project

- Make a cake
  - ⇒Waterfall Model
  - ⇒Agile Model

History

- 2001: Agile Manifesto written by 17 experts in the USA
- Based on RAD (Rapid Application, Development) , method defined in 1990s



Agile Manifesto

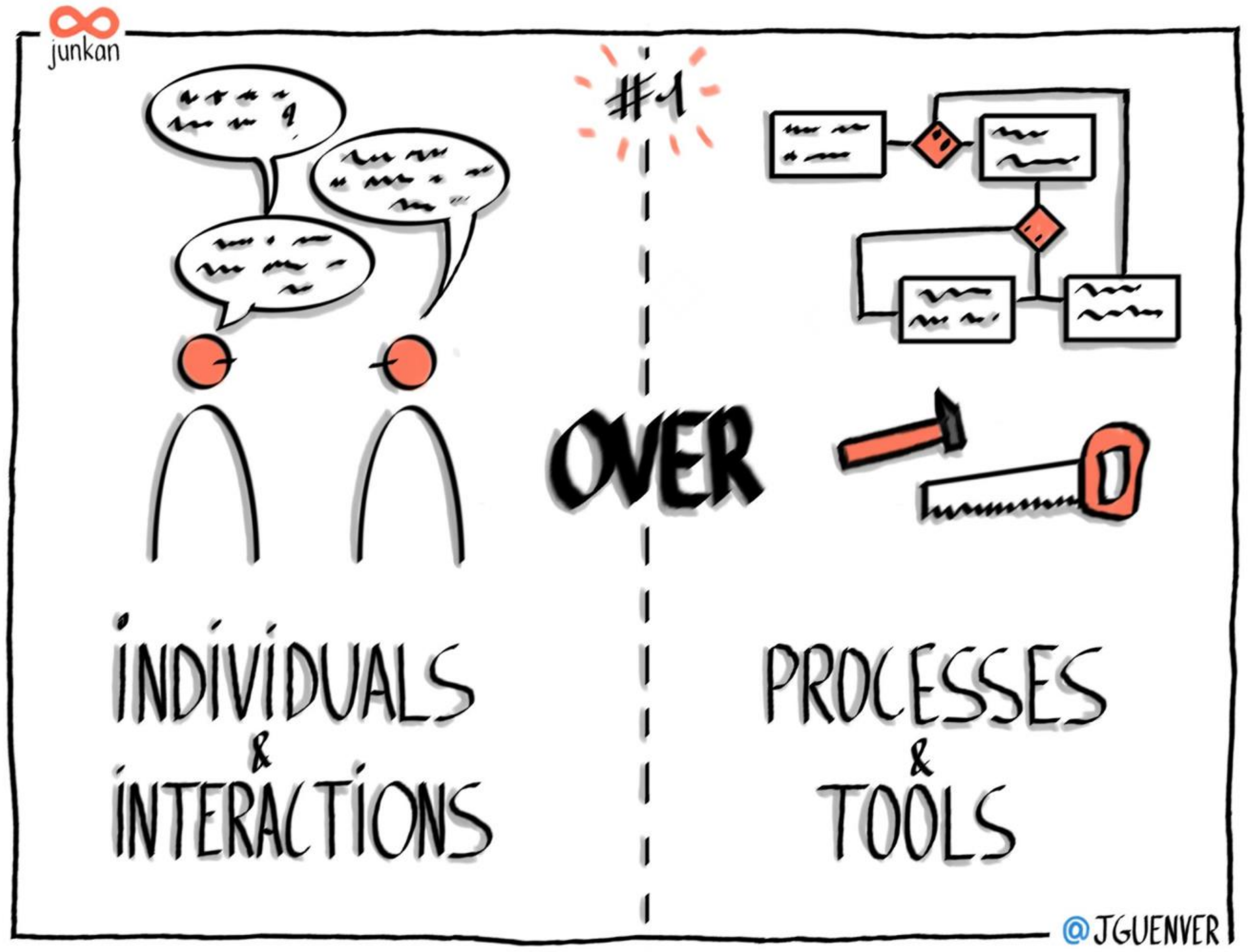
4 Values

12 Principles

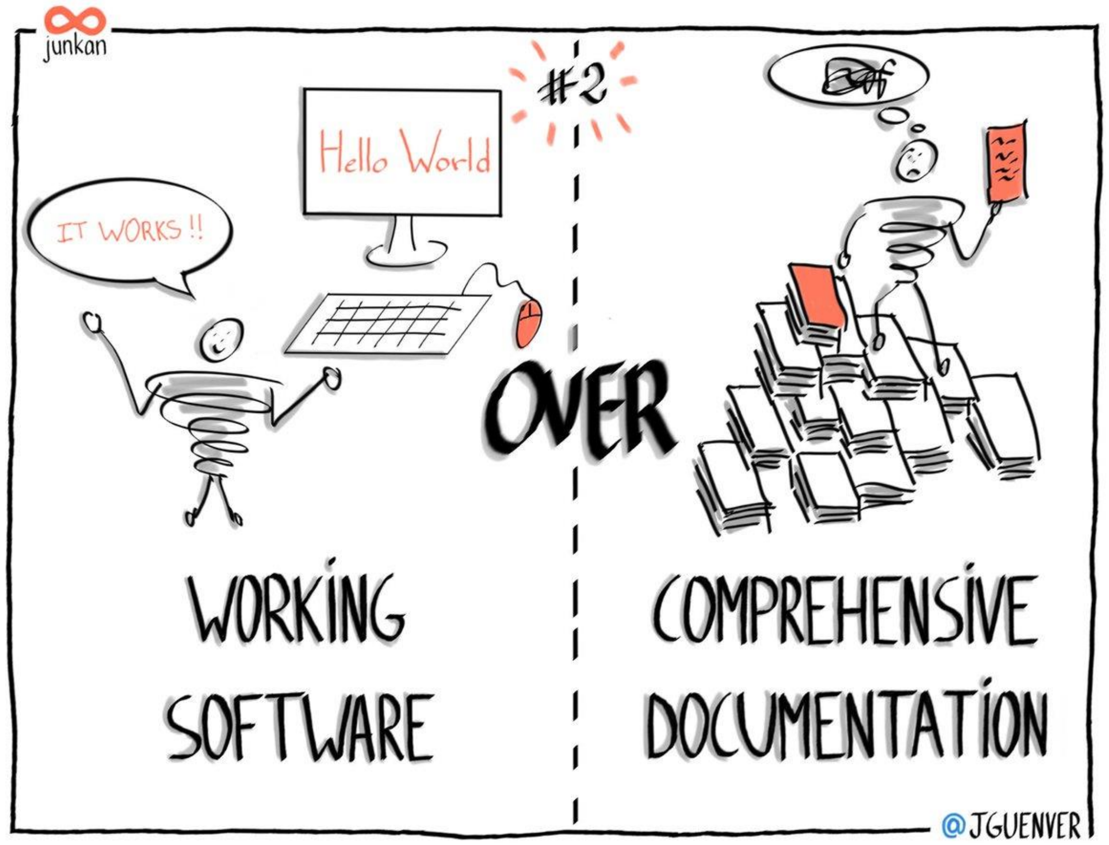
4 main Values

Individuals and interactions	Working software	Customer collaboration	Responding to change
OVER	OVER	OVER	OVER
Processes and tools	Comprehensive documentation	Contract negotiation	Following a plan

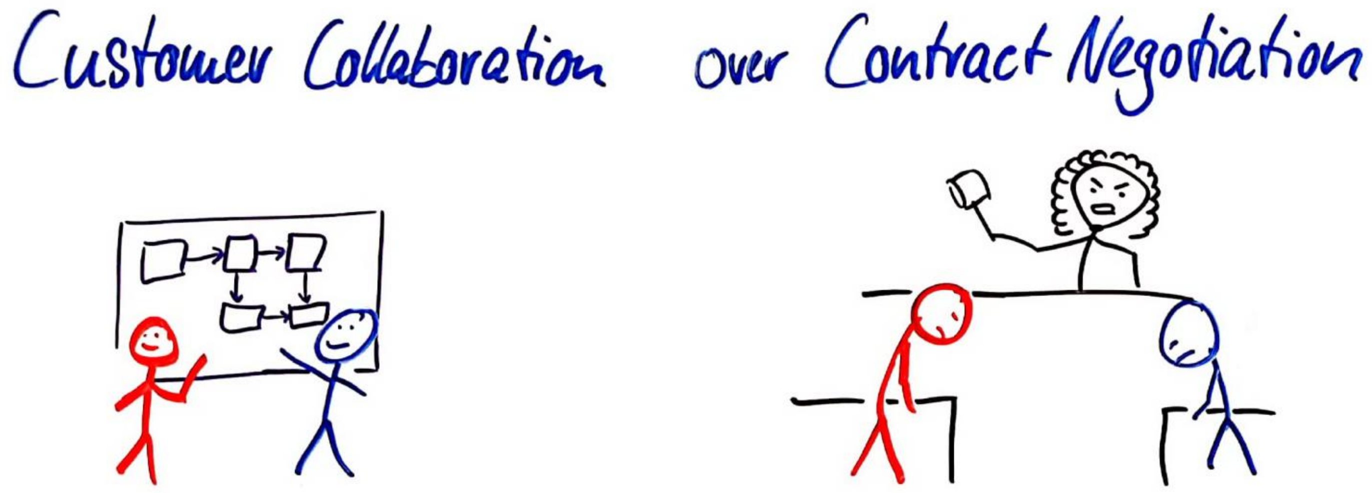
Individuals and interactions over processes and tools



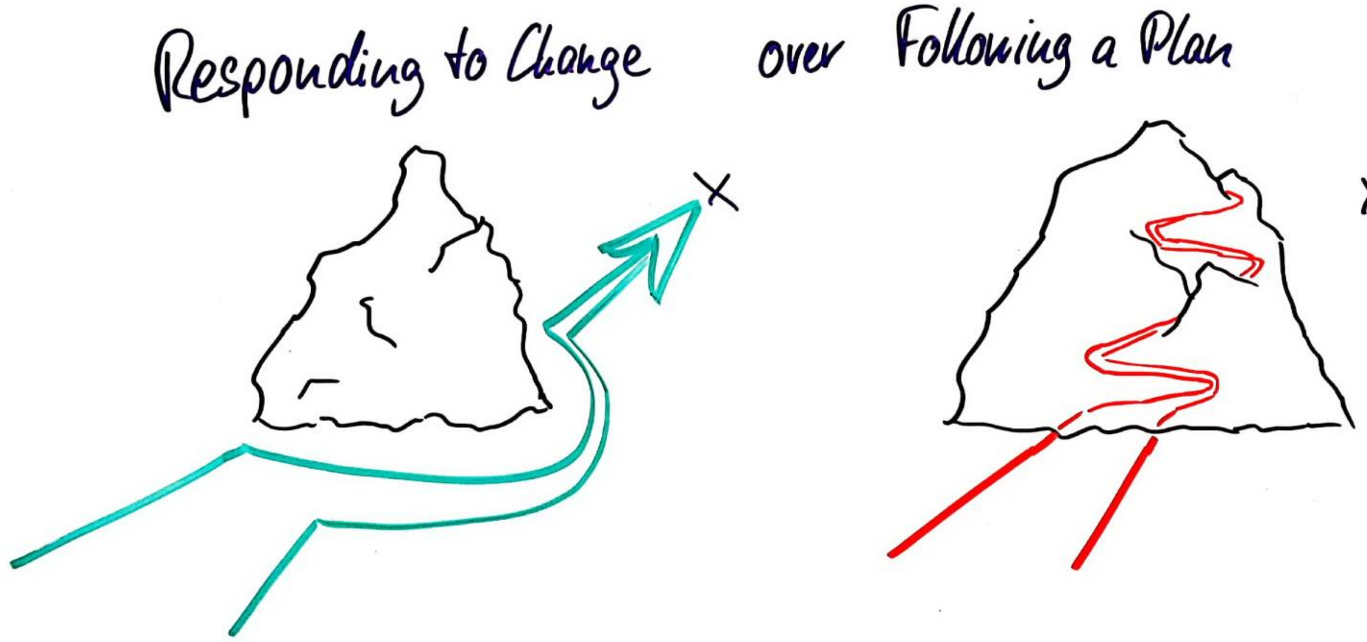
Working software over comprehensive documentation



Customer collaboration over contract negotiation



Responding to change over following a plan

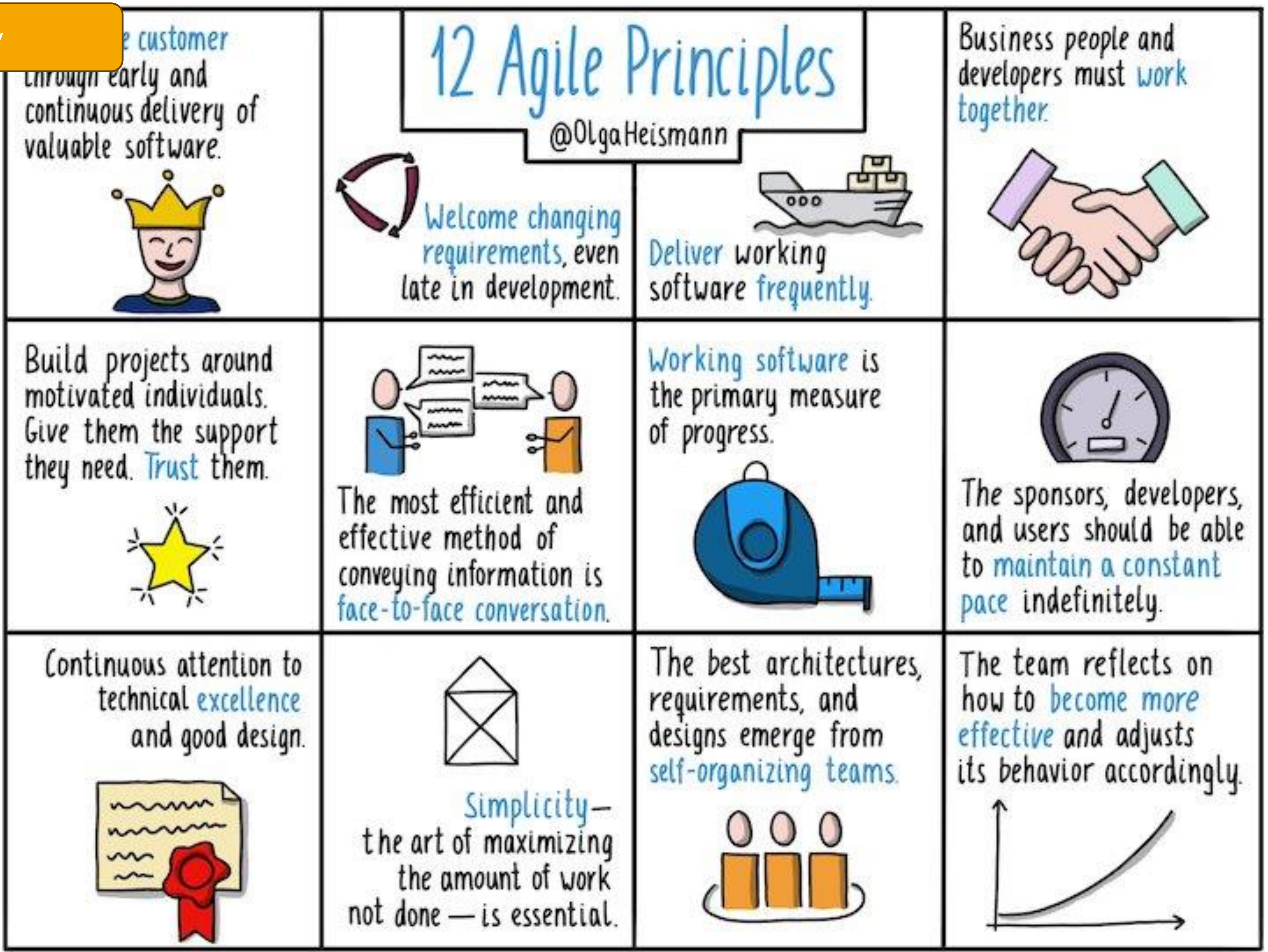


Values vs Rules

Don't forget that agile is a mindset, not a set of strict rules to follow.

12 Principles

Introduction to agility



12 Agile Principles

@OlgaHeismann

e customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development.

Deliver working software frequently.

Business people and developers must work together.

Build projects around motivated individuals. Give them the support they need. Trust them.

The most efficient and effective method of conveying information is face-to-face conversation.

Working software is the primary measure of progress.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design.

Simplicity— the art of maximizing the amount of work not done — is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

The team reflects on how to become more effective and adjusts its behavior accordingly.

Satisfy the customer !

Our highest priority is to satisfy the **customer** through early and continuous delivery of valuable software.

Welcome changing requirements!

Welcome **changing** requirements, even late in development.

Agile processes harness change for the customer's competitive advantage.

AGILE METHODS WELCOME CHANGING REQUIREMENTS, EVEN LATE IN DEVELOPMENT
CRAP!
THEY WANT YOU TO USE THIS NEW ENGINE!

Deliver working software frequently!

Deliver working software **frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale..

Business people and developers must work together

Business people and developers must **work together** daily throughout the project.

Motivation and Trust

Build projects around **motivated** individuals.

Give them the environment and support they need and **trust** them to get the job done.

Build projects around motivated individuals
Agile Manifesto, principle 5

Face-to-face conversation

The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation.**

Working software

Working software is the primary measure of progress

Maintain a constant pace

Agile processes promote sustainable development.

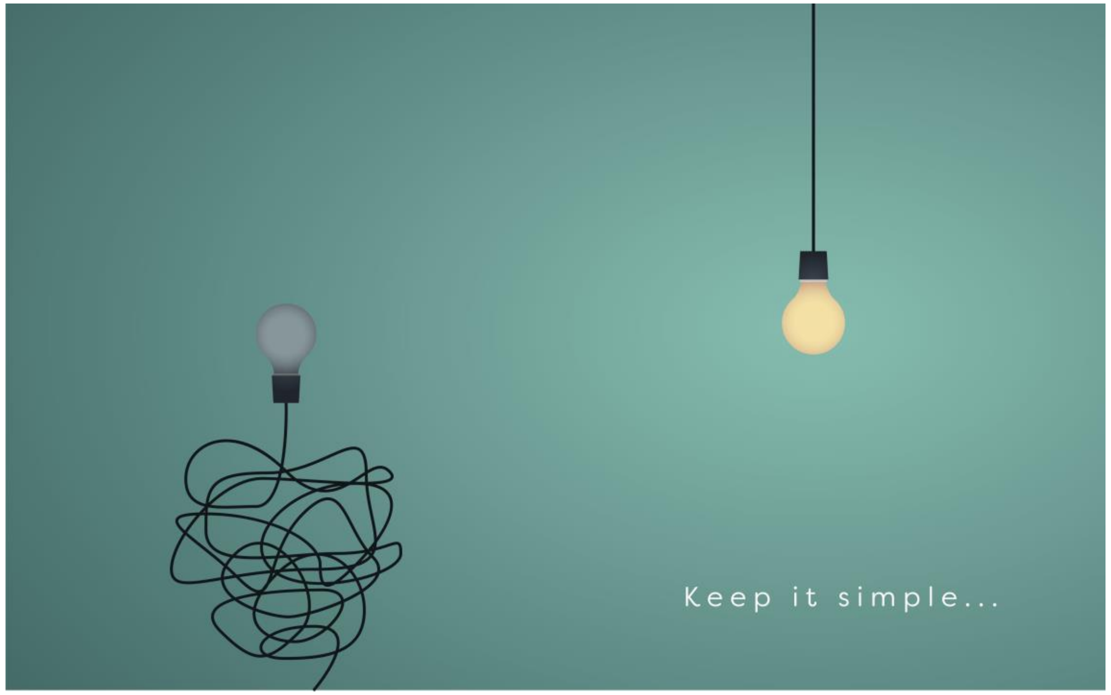
The sponsors, developers, and users should be able to maintain a **constant pace** indefinitely.

Excellence and good design

Continuous attention to **technical excellence** and **good design** enhances agility.

Simplicity

Simplicity-the art of maximizing the amount of work not done-is essential.



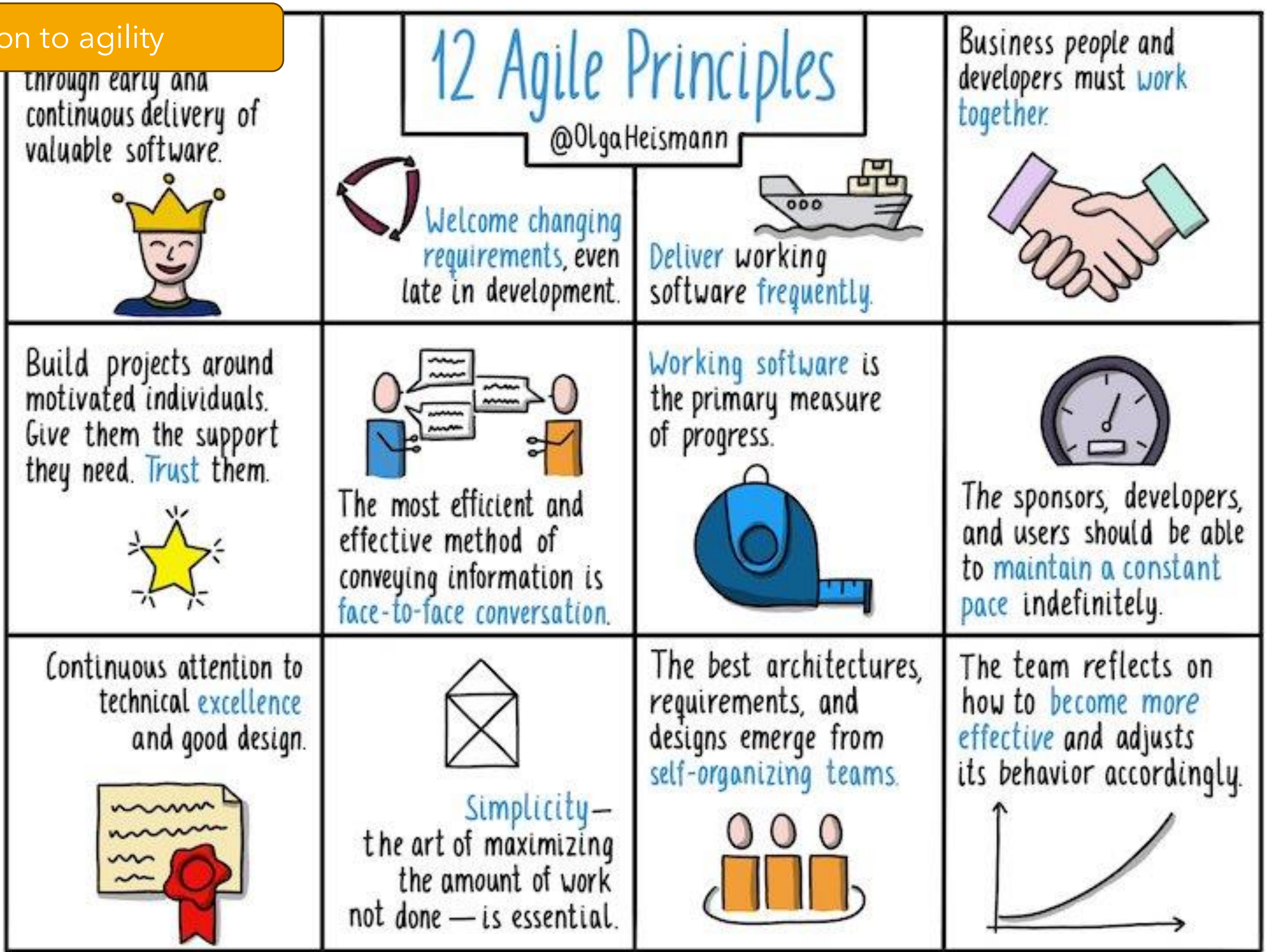
Self-organizing teams

The best architectures, requirements, and designs emerge from **self-organizing teams.**

Become more effective

At regular intervals, the team reflects on how to become more **effective**, then tunes and adjusts its behavior accordingly.

Introduction to agility



12 Agile Principles

@OlgaHeismann

through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development.

Deliver working software frequently.

Business people and developers must work together.

Build projects around motivated individuals. Give them the support they need. Trust them.

The most efficient and effective method of conveying information is face-to-face conversation.

Working software is the primary measure of progress.

The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

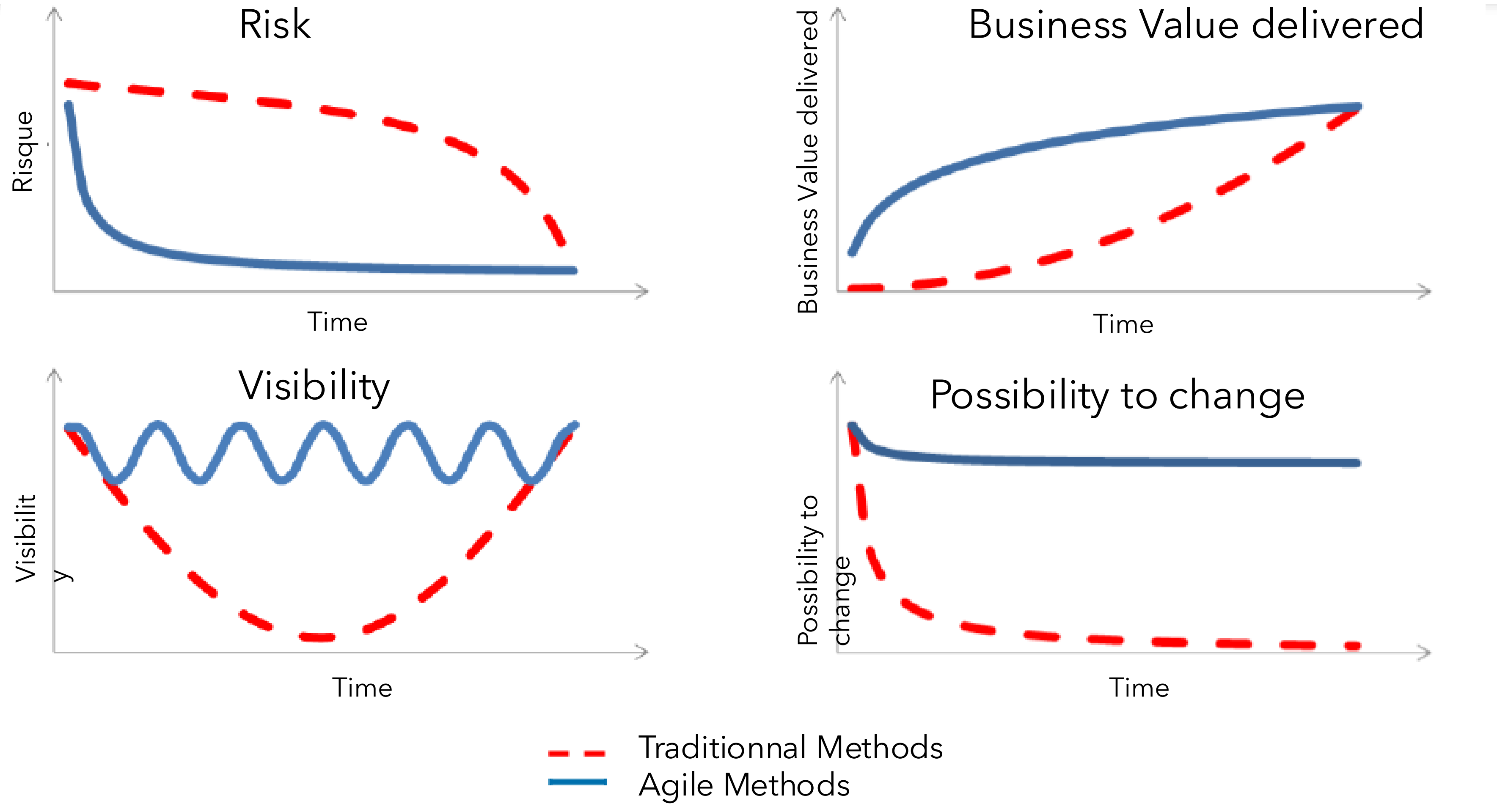
Continuous attention to technical excellence and good design.

Simplicity— the art of maximizing the amount of work not done — is essential.

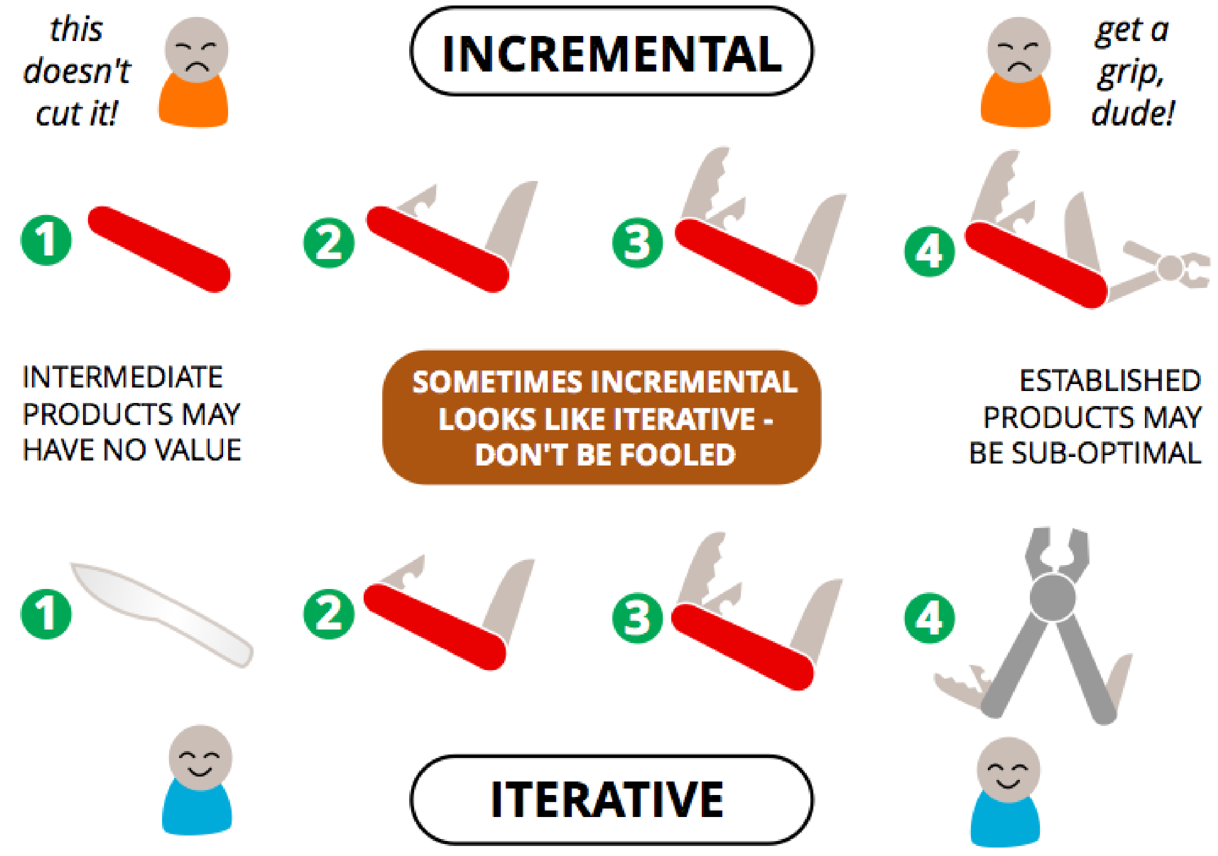
The best architectures, requirements, and designs emerge from self-organizing teams.

The team reflects on how to become more effective and adjusts its behavior accordingly.

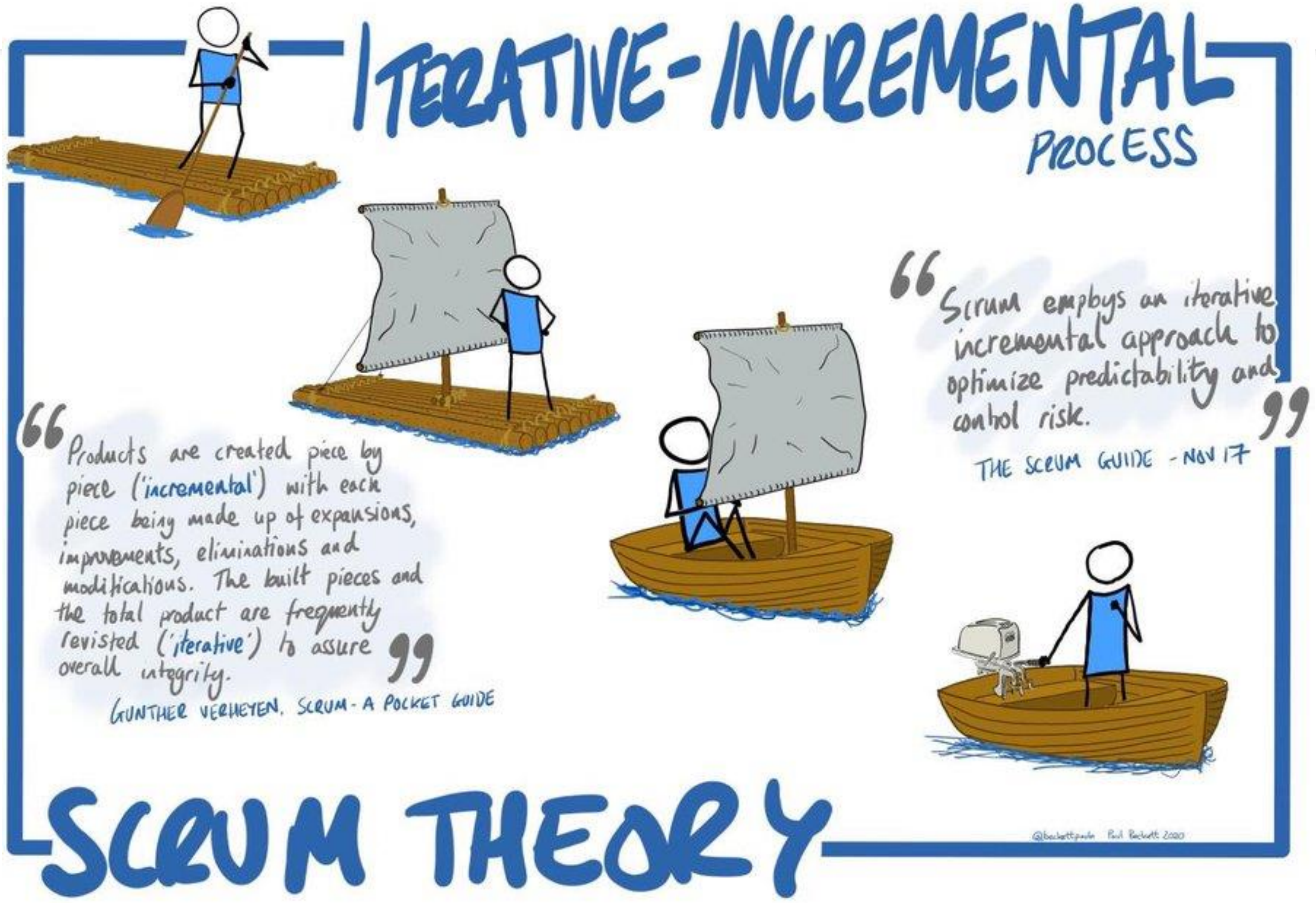
What will Agile bring?



Iterative or Incremental


this doesn't cut it!
INCREMENTAL
get a grip, dude!
1
2
3
4
INTERMEDIATE PRODUCTS MAY HAVE NO VALUE
SOMETIMES INCREMENTAL LOOKS LIKE ITERATIVE - DON'T BE FOOLED
ESTABLISHED PRODUCTS MAY BE SUB-OPTIMAL
1
2
3
4
ITERATIVE

Iterative or Incremental



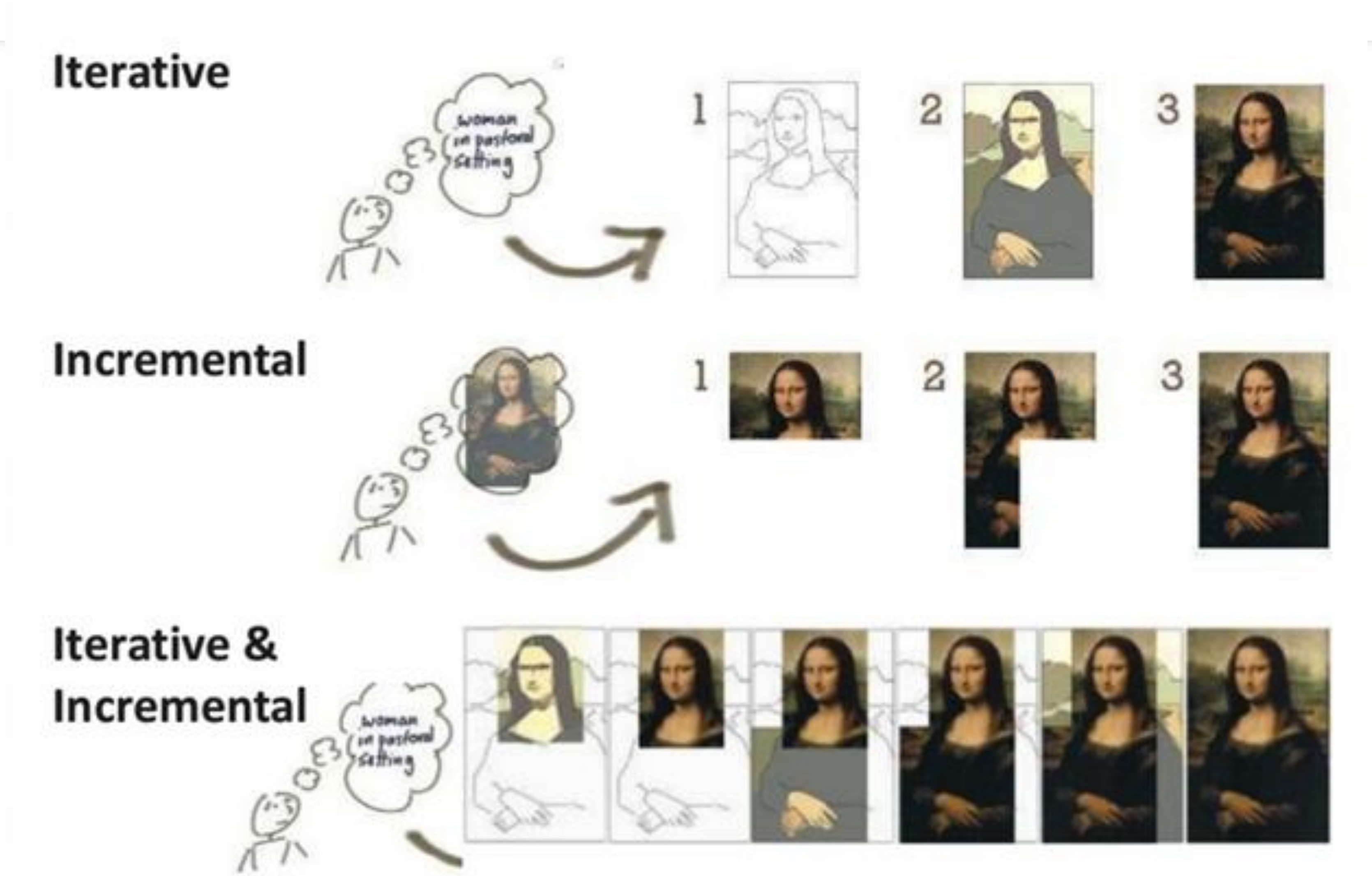
"Products are created piece by piece ('incremental') with each piece being made up of expansions, improvements, eliminations and modifications. The built pieces and the total product are frequently revisited ('iterative') to assure overall integrity."

GUNTHER VERHEYEN. SCRUM - A POCKET GUIDE

"Scrum employs an iterative incremental approach to optimize predictability and control risk."

THE SCRUM GUIDE - NOV 17

Iterative or Incremental

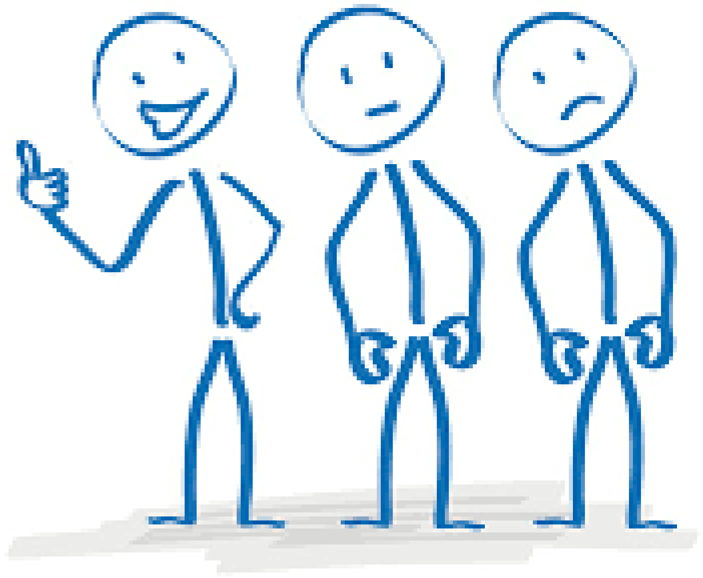


Overview of SCRUM FRAMEWORK

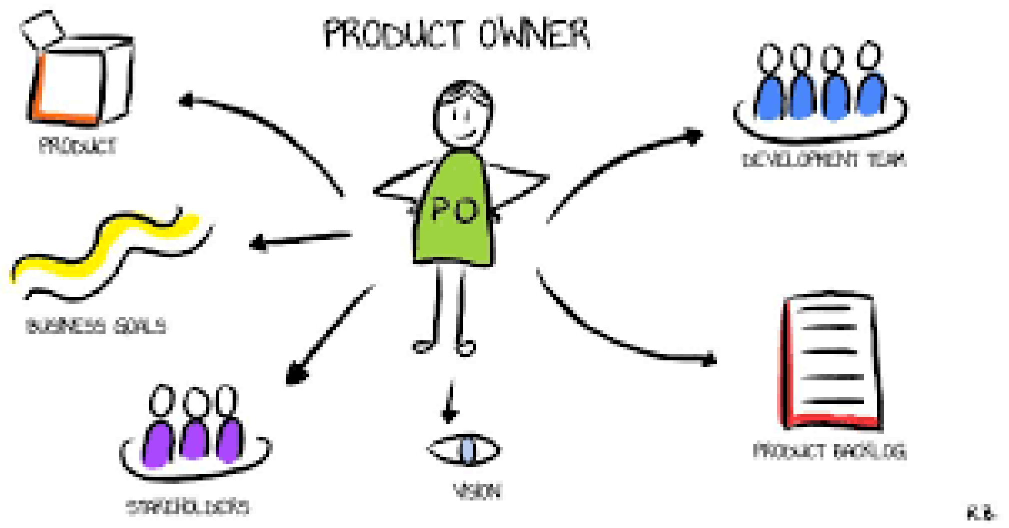
Part 2

3 roles

- Product Owner (PO)
- Scrum Master (SM)
- Developper



What does the PO do?



- The Product Owner
  - Is solely responsible for updating the **Product Backlog**
  - Prioritizes the User Stories formulated in the **Product Backlog.**
  - Monitors the budget and the calendar through the **Product Backlog**.

Developers

- Developers are in charge of "project operations."
- They deliver the functionalities to on **a regular basis.**

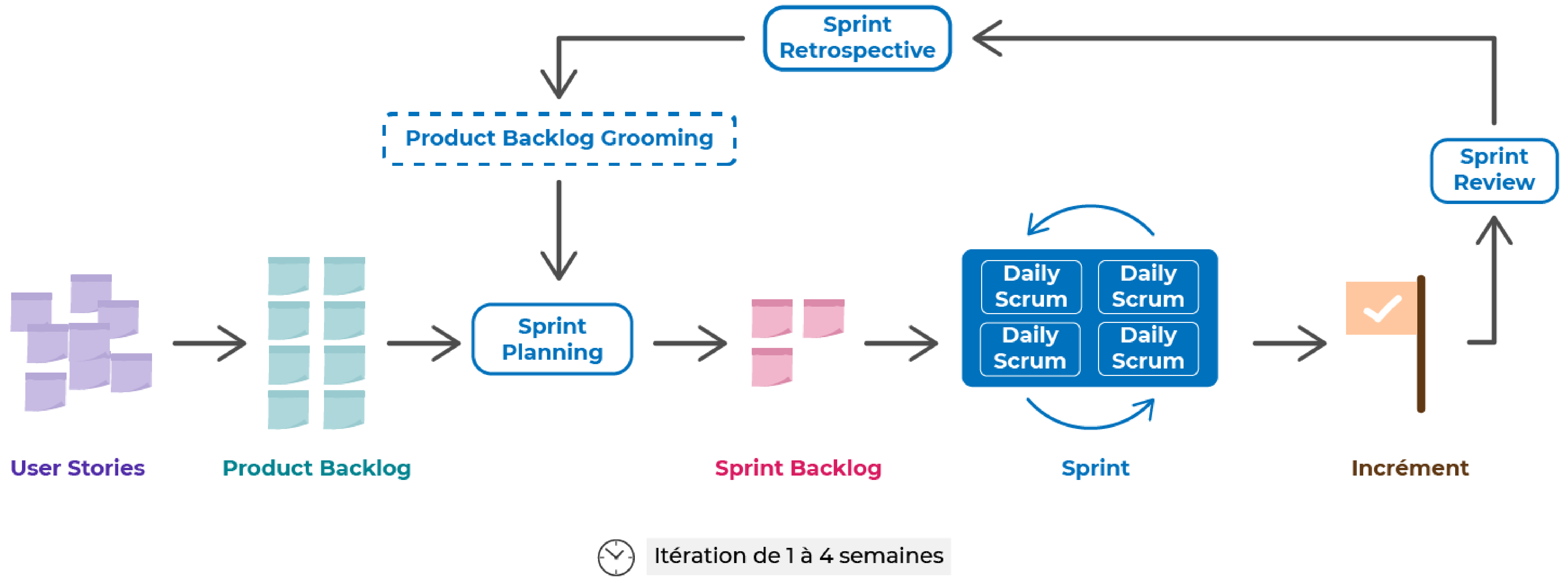
Scrum master

scrum police
hero
scribe
secretary
SCRUM MASTER ?
coffee clerk
chairman
admin
JIRA
team boss

- Responsible for understanding, respecting and **implementing the Scrum model.**
- He is **at the service** of the developers, the Product Owner and the project.
- **The scrum master is not a boss!**

Source: lesvoixdelaveille

Scrum

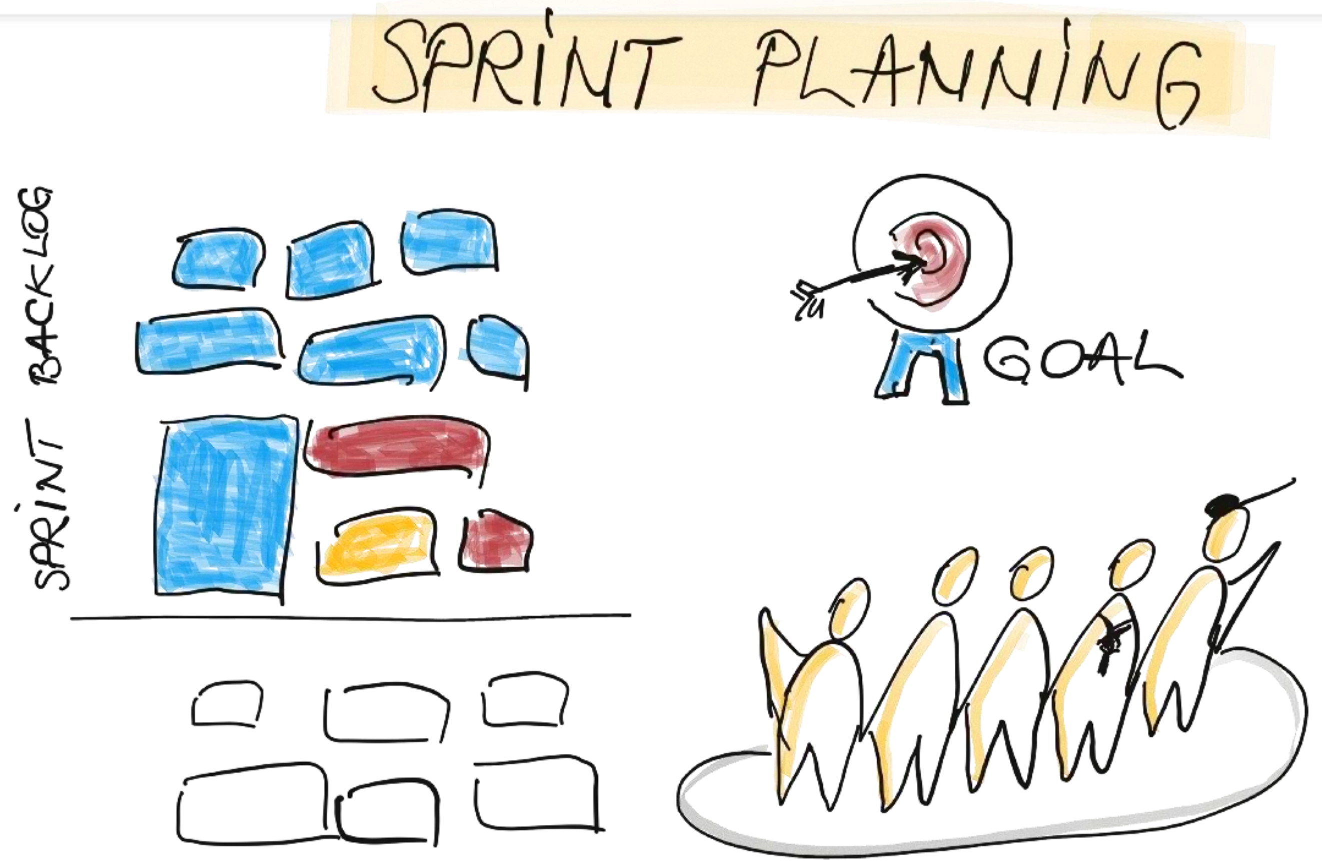

Sprint Retrospective
Product Backlog Grooming
Sprint Review
Daily Scrum
Daily Scrum
Daily Scrum
Daily Scrum
Sprint Planning
User Stories
Product Backlog
Sprint Backlog
Sprint
Incrément
Itération de 1 à 4 semaines

Let’s start a sprint!

- Box time from 1 to 4 weeks (maximum).
    - This is the period during which all product features will be developed and incremented.
- The duration of the sprints is **the same** throughout the duration of the project.
- The number of sprints depends on the team's ability to cover needs.
- A new sprint begins as soon as the previous one is completed.
- The decision to cancel a sprint or end it prematurely can only be made by the Product Owner.

Sprint Planning

- Start each sprint with a ritual event (or ceremony) intended to define the sprint's objective and its content.
- Decision with the Scrum team which User Stories will be developed during the sprint



Source: Ancaonuta

Daily Scrum Meeting

- 1/ day
- **15 minutes max**
- Fixed time
- **Stand-up**
- All team members
- No problem solving
- Graph update

DAILY SCRUM
sprint backlog
sprint goal
TODO
DOING
DONE
definit
of dor
The Liberators

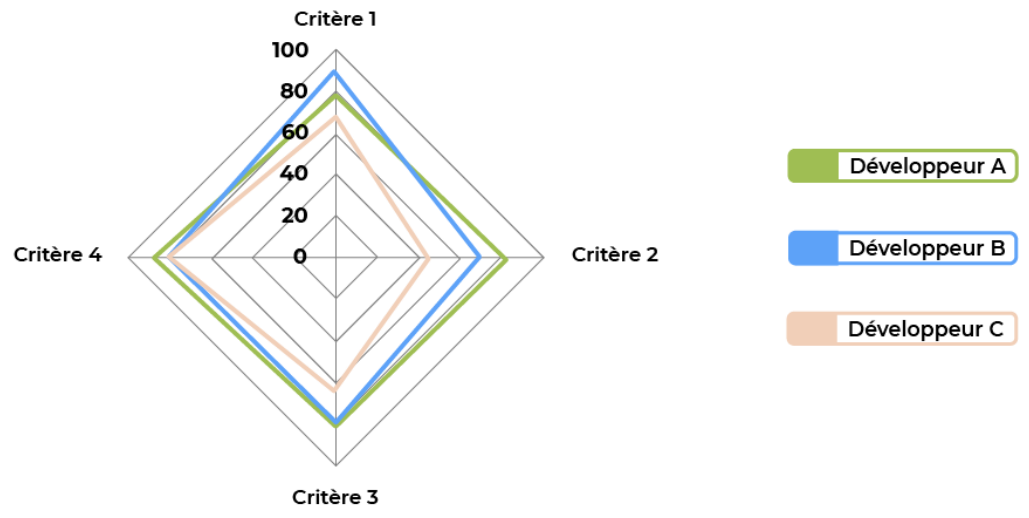
Sprint review

- The Sprint Review is intended to validate the Sprint Increment with the Scrum Team and project stakeholders.
- The team demonstrates completed features directly on the application.
- Revision of the Product Backlog and in particular the complexity of the User Stories with regard to what is observed during the sprint.

Sprint retrospective

- Your sprint retrospective is intended to make an appraisal of the sprint.

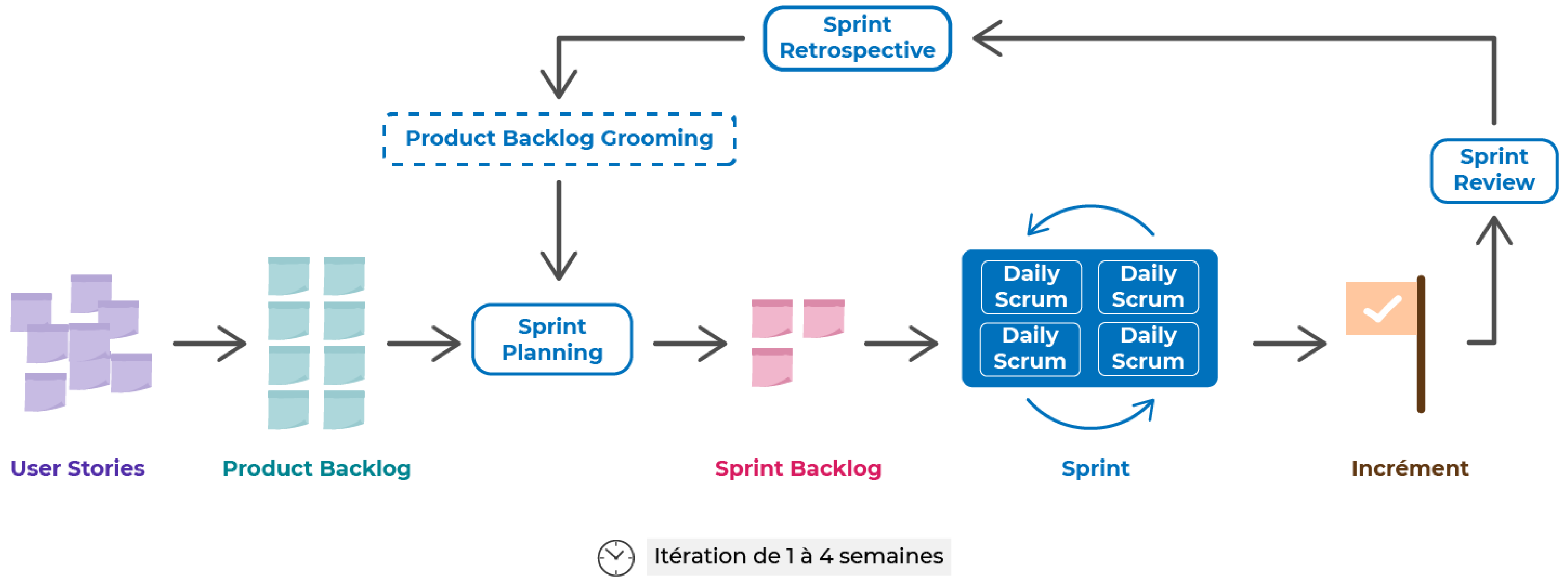
- Satisfaction radar + action plan



Product Backlog Grooming

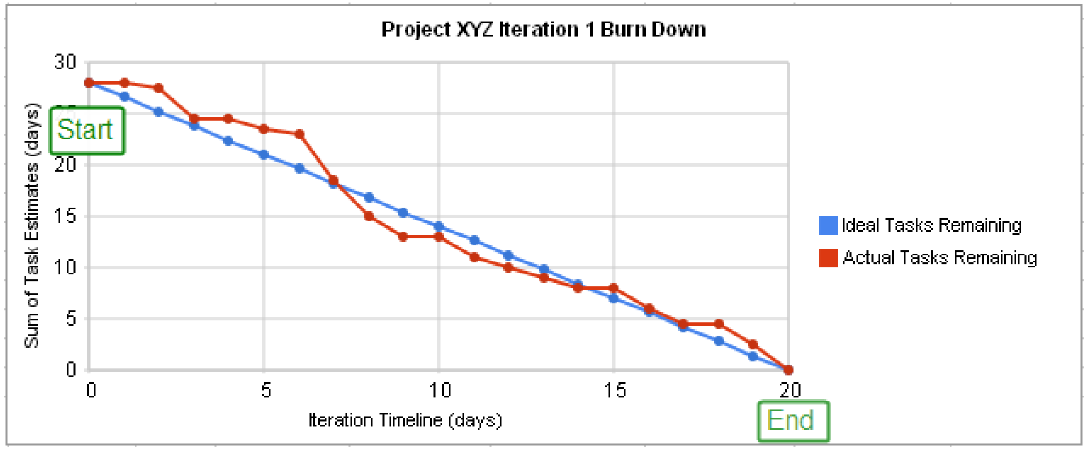
- Detect User Stories that no longer make sense for the project.
- Formulate and estimate new User Stories if needed.
- Reevaluate the order of priority of User Stories in the Product Backlog.
- Cut User Stories that could apply to future sprints.

Scrum

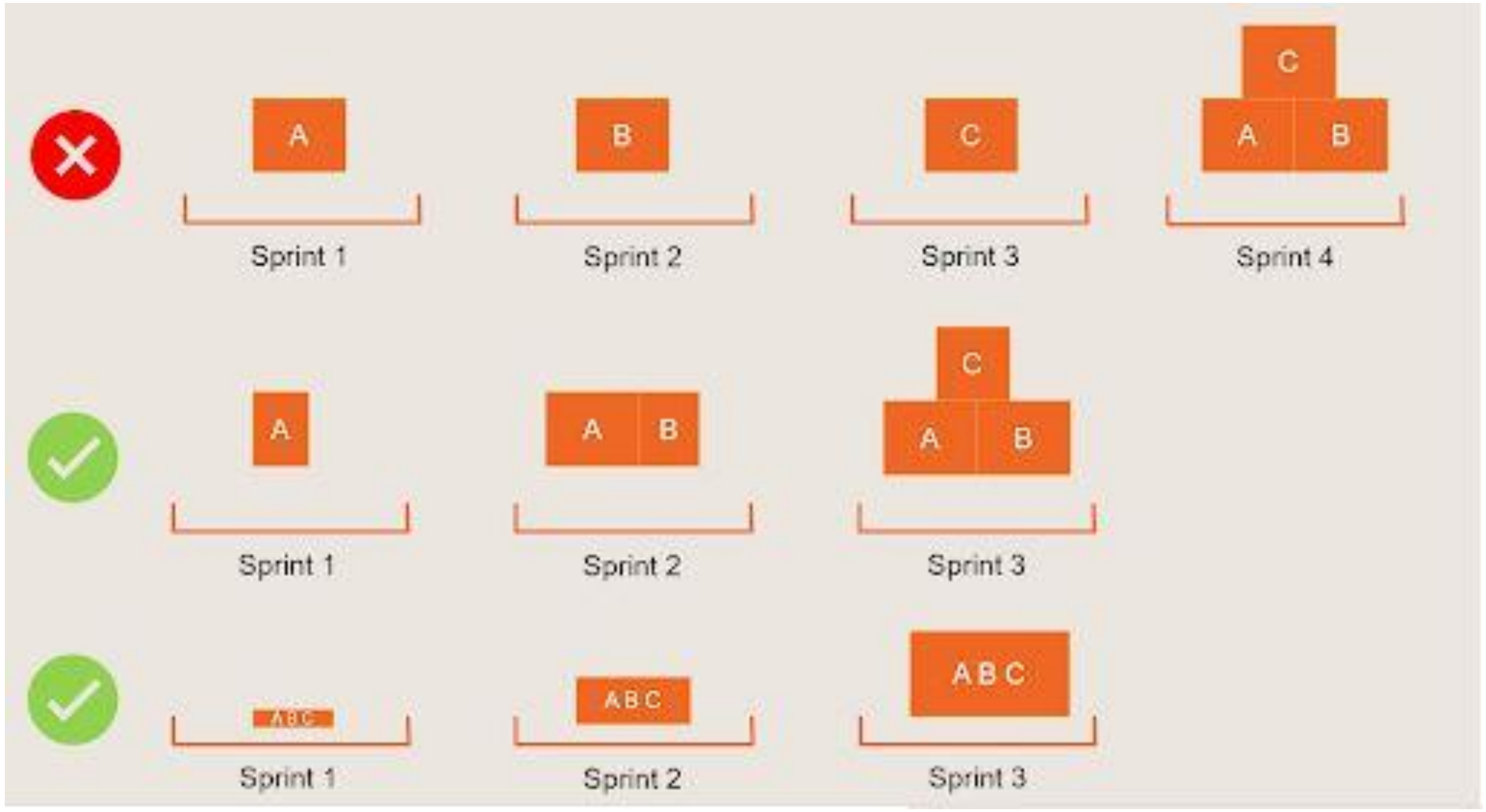

Sprint Retrospective
Product Backlog Grooming
Sprint Review
Daily Scrum
Daily Scrum
Daily Scrum
Daily Scrum
Sprint Planning
User Stories
Product Backlog
Sprint Backlog
Sprint
Incrément
Itération de 1 à 4 semaines

Sprint Backlog: Burndown Chart

- The vertical axis represents the amount of work left to do (in Story Points).
- The horizontal axis corresponds to the duration of the sprint (in days).

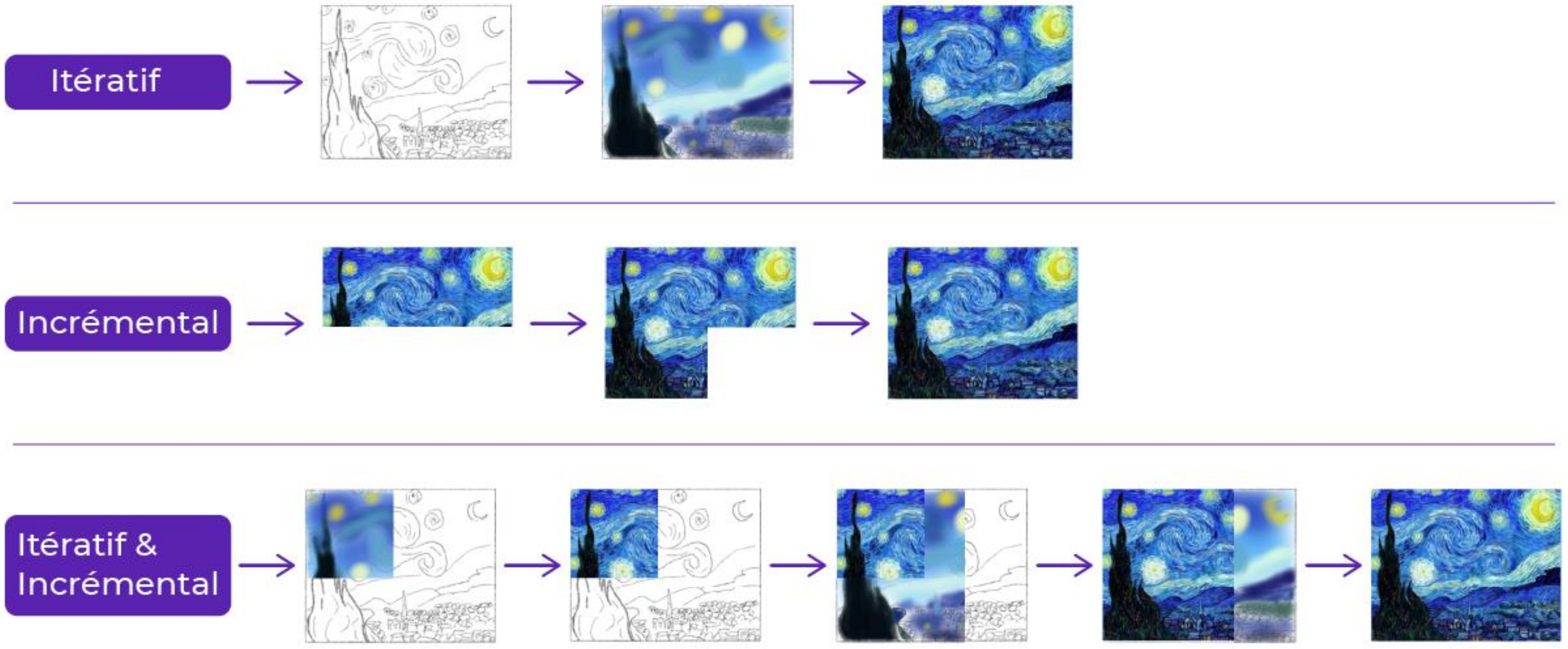


The increment



Source: DantotsuPM.co

Overview of the Scrum framework



Source: OpenClassroom

Introduction to kanban

What is Kanban?

Kanban is a simple method for **visualizing work** and thus managing it better.

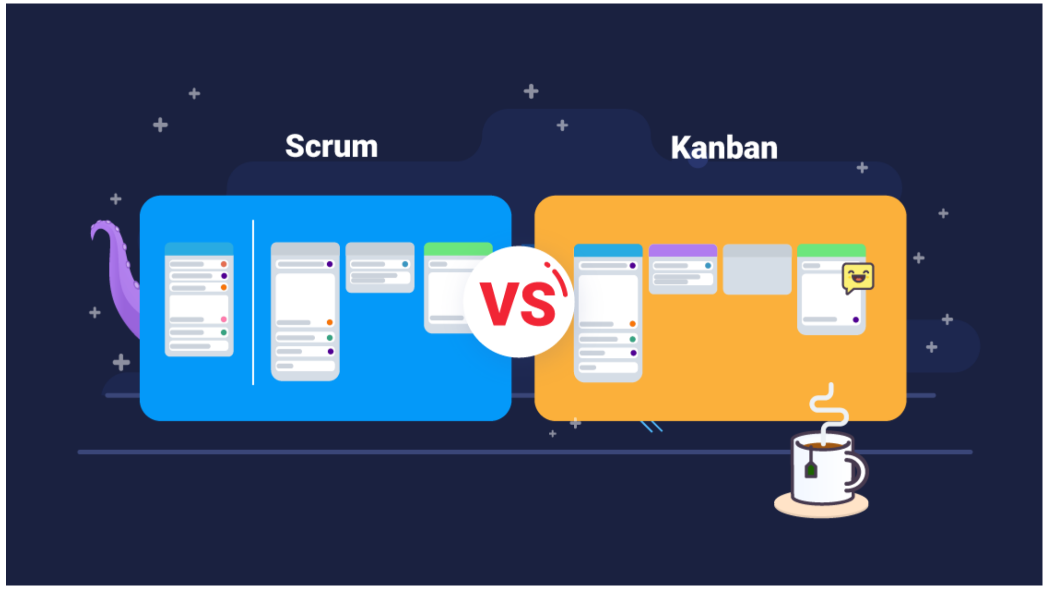
History of Kanban

- Developed by Toyota under the leadership of Taiichi Ohno, known as the father of the Toyota Production System.

- Kanban means Card in japanese

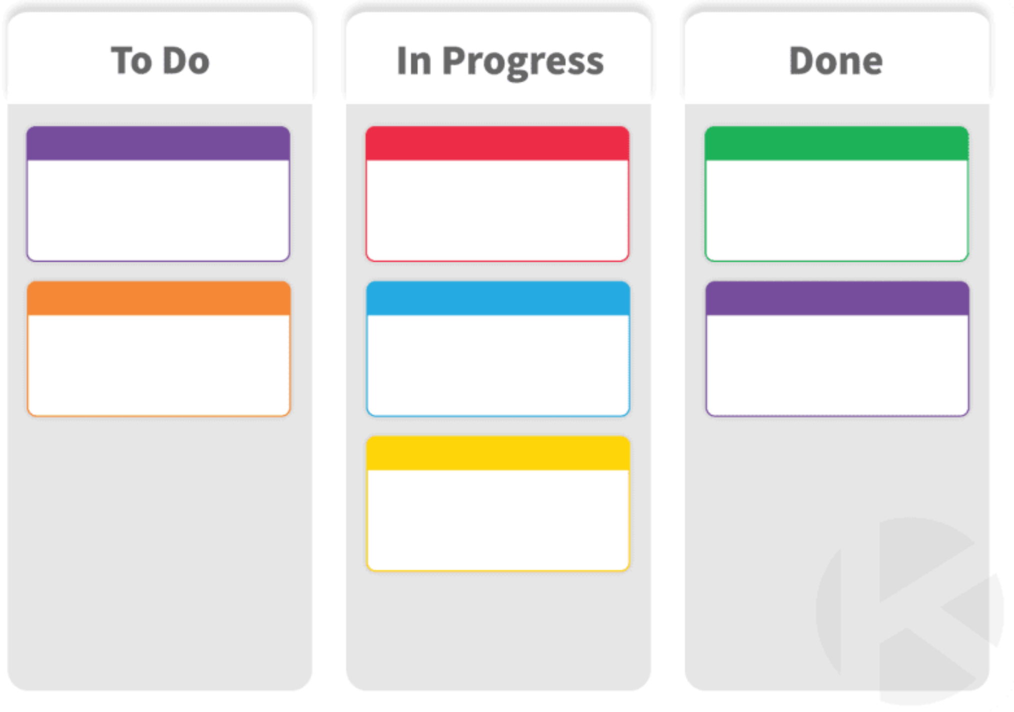
Scrum & Kanban

- Both Scrum and Kanban aim to improve productivity and ensure efficient delivery of high-quality products, but they do so through different approaches.
- Essentially, **Kanban** emphasizes **flexibility and continuous delivery**, making it suitable for continuous processes and teams focused on optimizing workflows.
- On the other hand, **Scrum** offers a structured approach with **time-limited iterations**, making it ideal for projects that require regular feedback and iterative development..



Kanban Board

- **Visual representation** of workflow, typically divided into columns that represent different stages of a process, such as “To Do,” “In Progress,” and “Done.”
- Clear and concise overview of **work progress,** making it easier for teams to track progress, **identify bottlenecks,** and ensure smooth workflow.
- Columns can be **customized** based on specific project needs, allowing for flexible and adaptable workflow visualization.



Kanban Cards

- Kanban cards represent **individual tasks** or work items that go through the different stages depicted on the Kanban board.
  - Each card contains **essential information** about the task, such as a description, responsible person, due date, and any relevant details or attachments.
  - These cards can be color-coded or labeled to indicate different work types, priorities or categories, improving clarity and organization of tasks.
- As work progresses, Kanban cards are moved across the columns of the board, providing a tangible and immediate visual indication of task status.
-

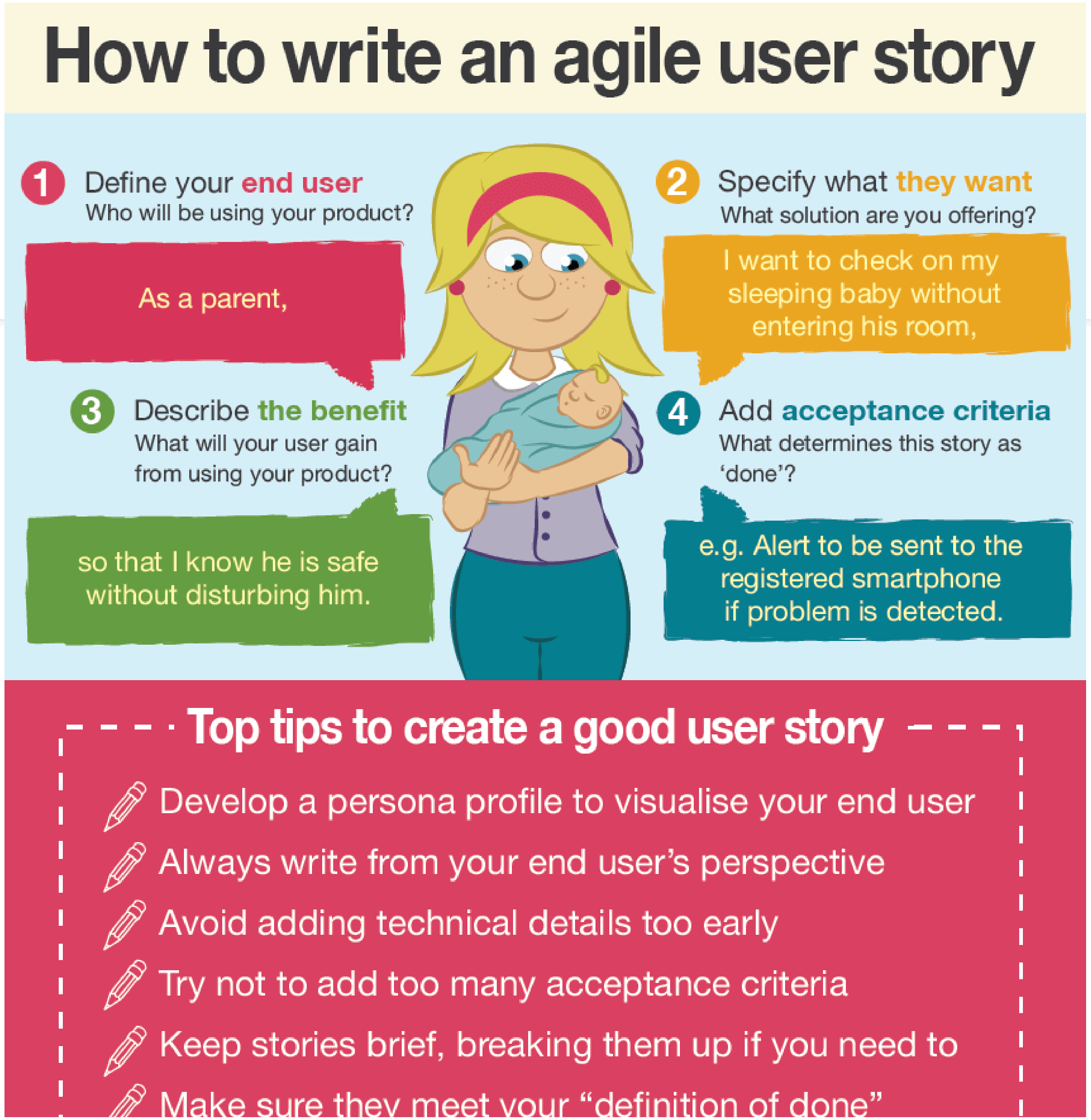
User stories and Product backlog

What is and Why a user story (US)?

- Simple sentences written in common language.
- Enable the team to precisely describe all the functionalities of the project.

- The user story is a great way to define clearly your product
- It helps you articulate your product's features using simple vocabulary<u>, without technical details</u>
- It enables to highlight <u>potential questions or disagreements</u>
- They help to clarify the teams on the "what" to build, for "who", "why" and "when", but also to the communication of the project outside the team.
- The user story encourages the participation of non-technical people in the project

User Story: 1 Sentence



How to write an agile user story

1 Define your **end user**
Who will be using your product?

As a parent,

2 Specify what **they want**
What solution are you offering?

I want to check on my sleeping baby without entering his room,

3 Describe **the benefit**
What will your user gain from using your product?

so that I know he is safe without disturbing him.

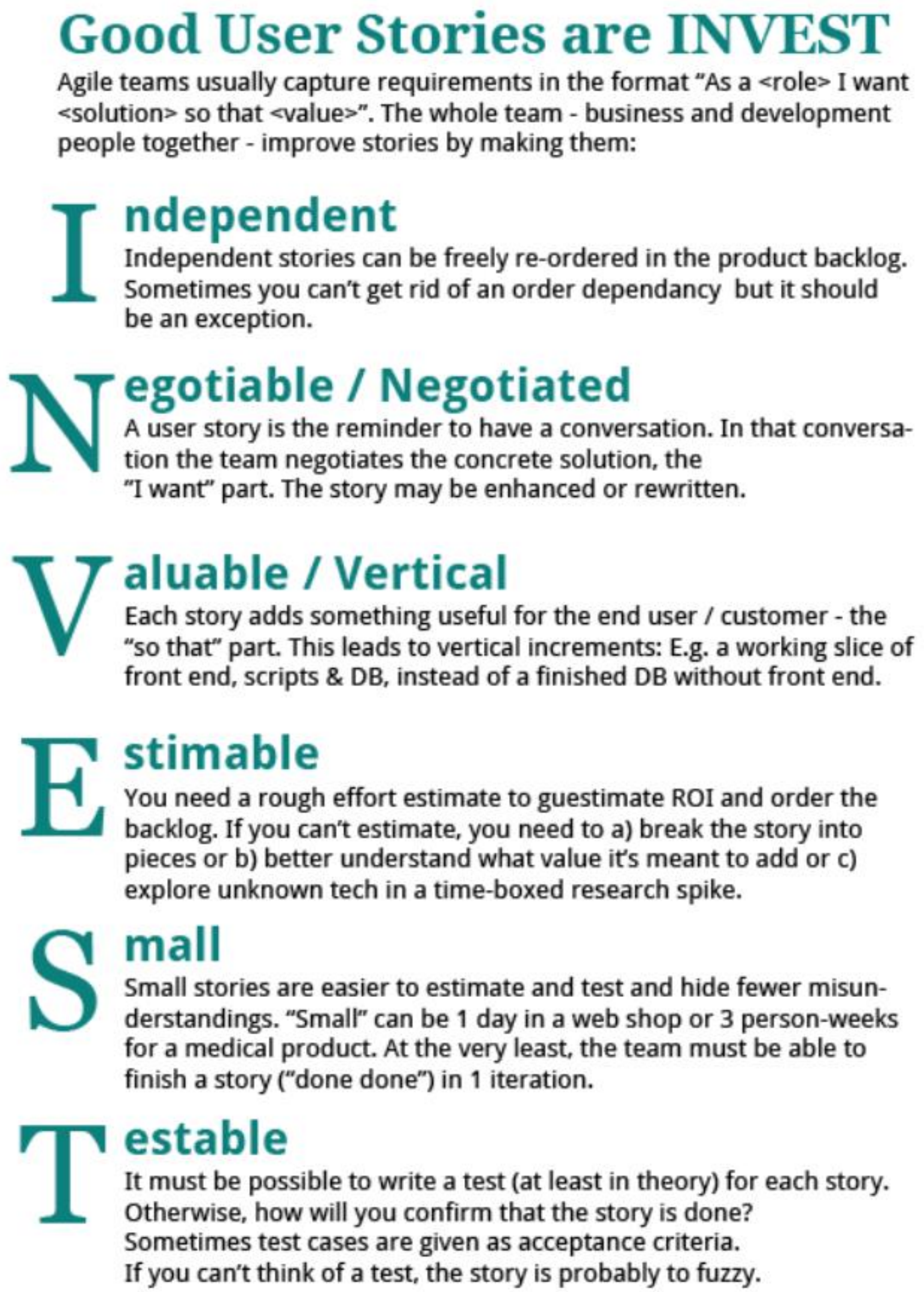
4 Add **acceptance criteria**
What determines this story as 'done'?

e.g. Alert to be sent to the registered smartphone if problem is detected.

Top tips to create a good user story

- Develop a persona profile to visualise your end user
- Always write from your end user's perspective
- Avoid adding technical details too early
- Try not to add too many acceptance criteria
- Keep stories brief, breaking them up if you need to
- Make sure they meet your "definition of done"

User stories & Product backlog



Good User Stories are INVEST

Agile teams usually capture requirements in the format "As a <role> I want <solution> so that <value>". The whole team - business and development people together - improve stories by making them:

I ndependent
Independent stories can be freely re-ordered in the product backlog. Sometimes you can't get rid of an order dependancy but it should be an exception.

N egotiable / Negotiated
A user story is the reminder to have a conversation. In that conversation the team negotiates the concrete solution, the "I want" part. The story may be enhanced or rewritten.

V aluable / Vertical
Each story adds something useful for the end user / customer - the "so that" part. This leads to vertical increments: E.g. a working slice of front end, scripts & DB, instead of a finished DB without front end.

E stimable
You need a rough effort estimate to guestimate ROI and order the backlog. If you can't estimate, you need to a) break the story into pieces or b) better understand what value it's meant to add or c) explore unknown tech in a time-boxed research spike.

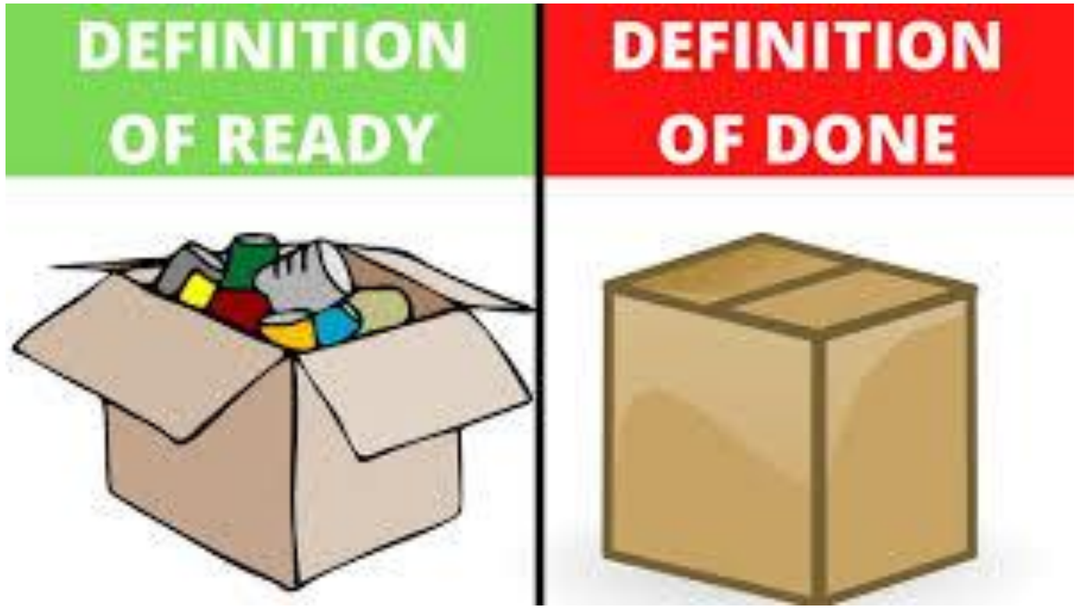
S mall
Small stories are easier to estimate and test and hide fewer misunderstandings. "Small" can be 1 day in a web shop or 3 person-weeks for a medical product. At the very least, the team must be able to finish a story ("done done") in 1 iteration.

T estable
It must be possible to write a test (at least in theory) for each story. Otherwise, how will you confirm that the story is done? Sometimes test cases are given as acceptance criteria. If you can't think of a test, the story is probably to fuzzy.

Source: Wall Skills

« Definition of ready » (DOR)

- Promote a good understanding
- Check the conformity of User Stories with the objectives of the project.
- **Explain the acceptance criteria for each User Story.**
- Validate the skills required within the Scrum team.
- Eliminate external dependencies



« Definition of Done » (DOD)

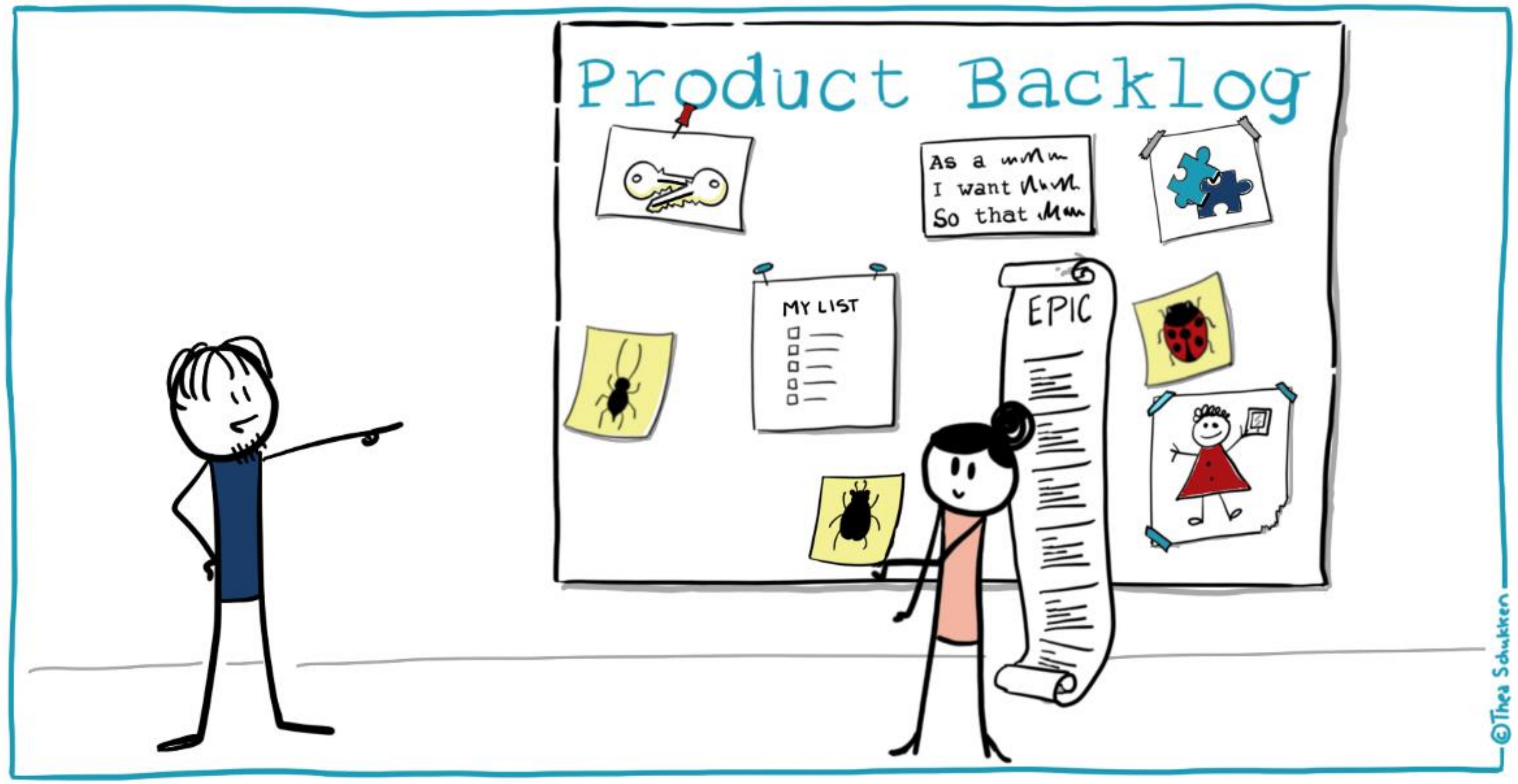
⇒works like a checklist

- All the actors of the project consult this list of criteria to follow the estimation or the realization of a functionality.

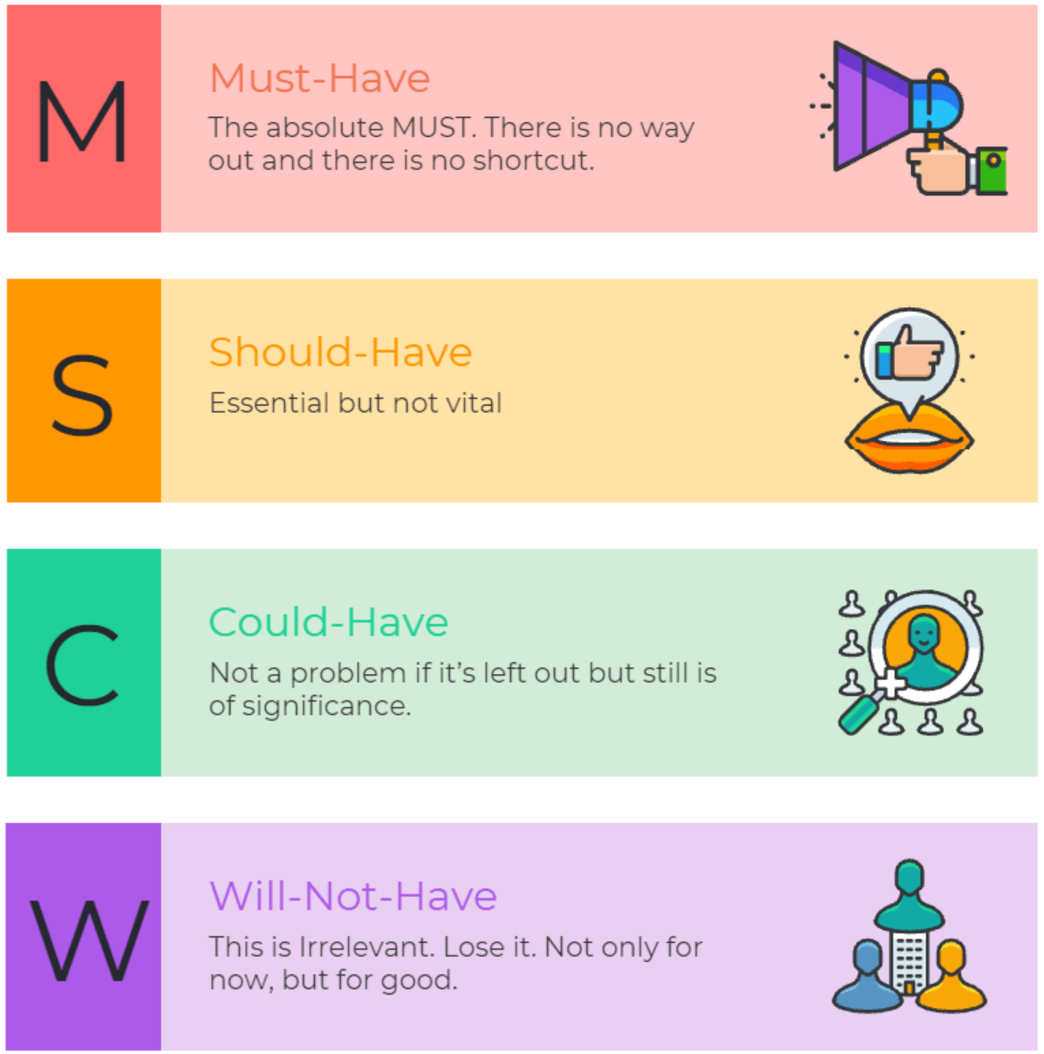
⇒**Given <a context>, when <the user performs some action> then <you see such consequences>.**

⇒Ex: Given app login page, when user enters correct username and password, then homepage appears.

Backlog Produit



MoSCoW



M **Must-Have**
The absolute MUST. There is no way out and there is no shortcut.

S **Should-Have**
Essential but not vital

C **Could-Have**
Not a problem if it's left out but still is of significance.

W **Will-Not-Have**
This is Irrelevant. Lose it. Not only for now, but for good.

Source: Project Cubicle

Agenda Practical case(2h)

- **Practical exercise:**
  - User Stories writing workshop and creation of a Kanban board:
    - Formation of **pairs**
    - Creation of User Stories for a fictitious project
    - Presentation of the User Stories created.
    - Creation of a Kanban board to visualize these User Stories.

Practical Case

Development of a mobile task management application

- Context: You are part of the development team of a technology start-up called "TaskMaster".
- Your team's mission is to develop a new mobile task management application intended for professionals and students.
- The goal is to create an intuitive application that allows you to efficiently manage daily tasks, plan projects, and collaborate with other users.
- The product should be simple to use, yet powerful enough to accommodate the needs of advanced users.

Practical Case

- **Main features of the application:**
  - Task management:
    - Creating, modifying and deleting tasks.
    - Assignment of priorities (low, medium, high).
    - Setting due dates.
    - Ability to add subtasks.
  - Project planning:
    - Organization of tasks into projects.
    - Visualization of project progress (e.g. simple Gantt chart).
    - Distribution of tasks between team members.
  - Collaboration:
    - Sharing projects with other users.
    - Assigning specific tasks to other users.
    - Discussions/comments on tasks.
  - Notifications:
    - Push notifications for imminent or overdue tasks.
    - Customizable reminders.
  - Cross-platform synchronization:
    - Real-time synchronization between devices (smartphones, tablets, computers).

- **Additional requirements:**
- The application must be available on iOS and Android.
- The user interface should be simple, modern and intuitive.
- A web version of the application must be provided to allow access via browser.
- The final product must meet data security standards, particularly for personal information and communications between users.

What is expected from you?

- User Stories writing workshop and creation of a Kanban board:
  - Formation of pairs
  - Creation of User Stories for the fictitious project
    - Write User Stories for the main functionalities of the application (task management, project planning, collaboration, etc.).
    - Prioritize User Stories in the Product Backlog.
    - Make sure User Stories are well defined and follow the INVEST model
  - Creation of a Kanban board to visualize these User Stories.
- Explanations on the difficulties encountered during the exercise.
- **PDF document sent by Google Form before today at 01:00 PM in French or in English**