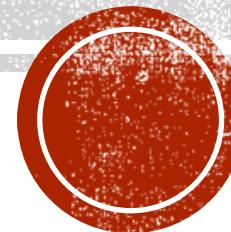


# SCRUM

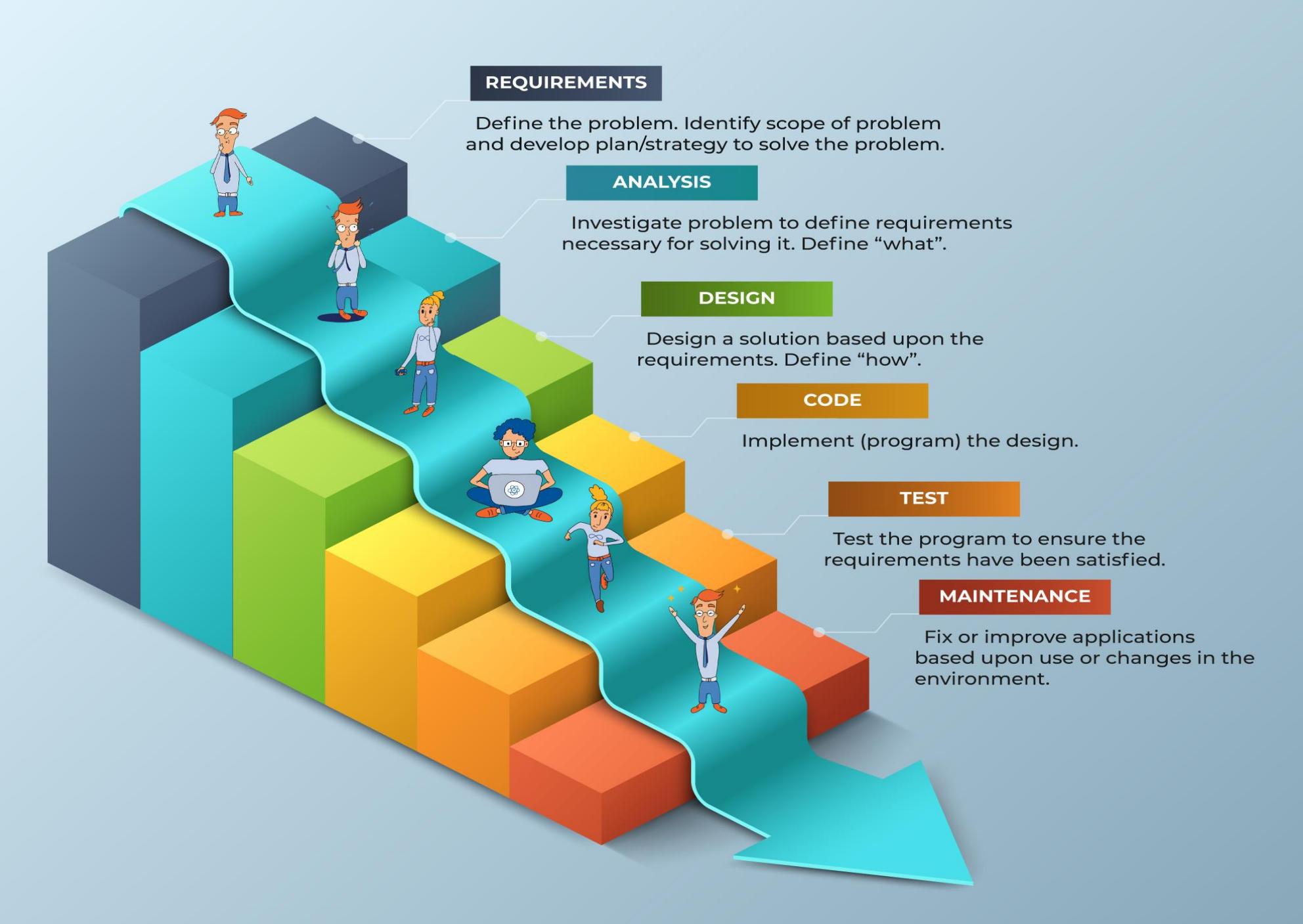


A mechanism to optimize the value delivery system.

A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.



# WATERFALL



# History of Scrum

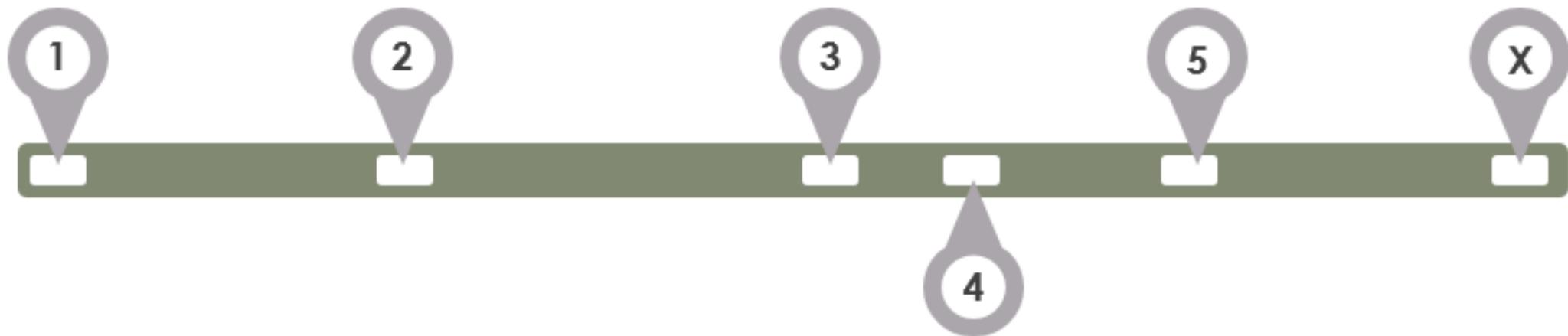
**1986** – Takeuchi & Nonaka coined Scrum product development in Harvard Business Review

**1993** – Jeff Sutherland created Scrum process

**2001** – Agile Manifesto

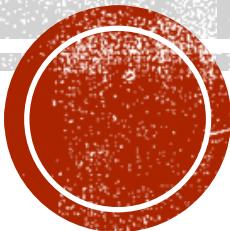
**2007** – Scaled Agile Framework (SAFe)

**2014** – Dr. Dave Cornelius “The Value of Scrum to Organizations”



2002 – Scrum Alliance Created

# **AGILE MANIFESTO, VALUES & PRINCIPLES**



# **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.

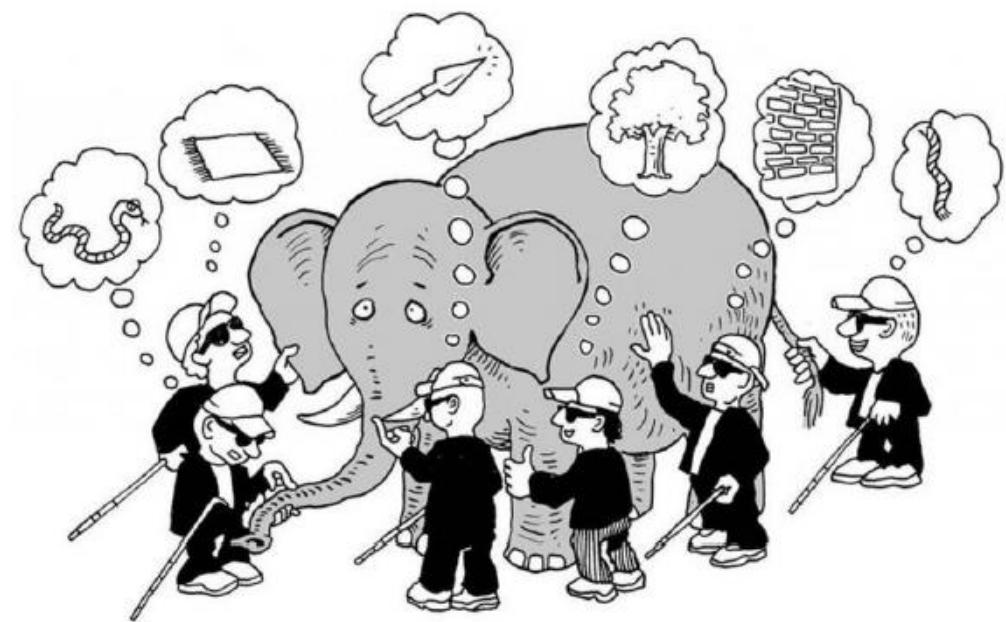
Individuals and interactions

over

Processes and tools



$$\begin{array}{r} 1 \\ + \\ 1 \end{array} = \begin{array}{r} 3 \\ \hline 1234 \end{array}$$

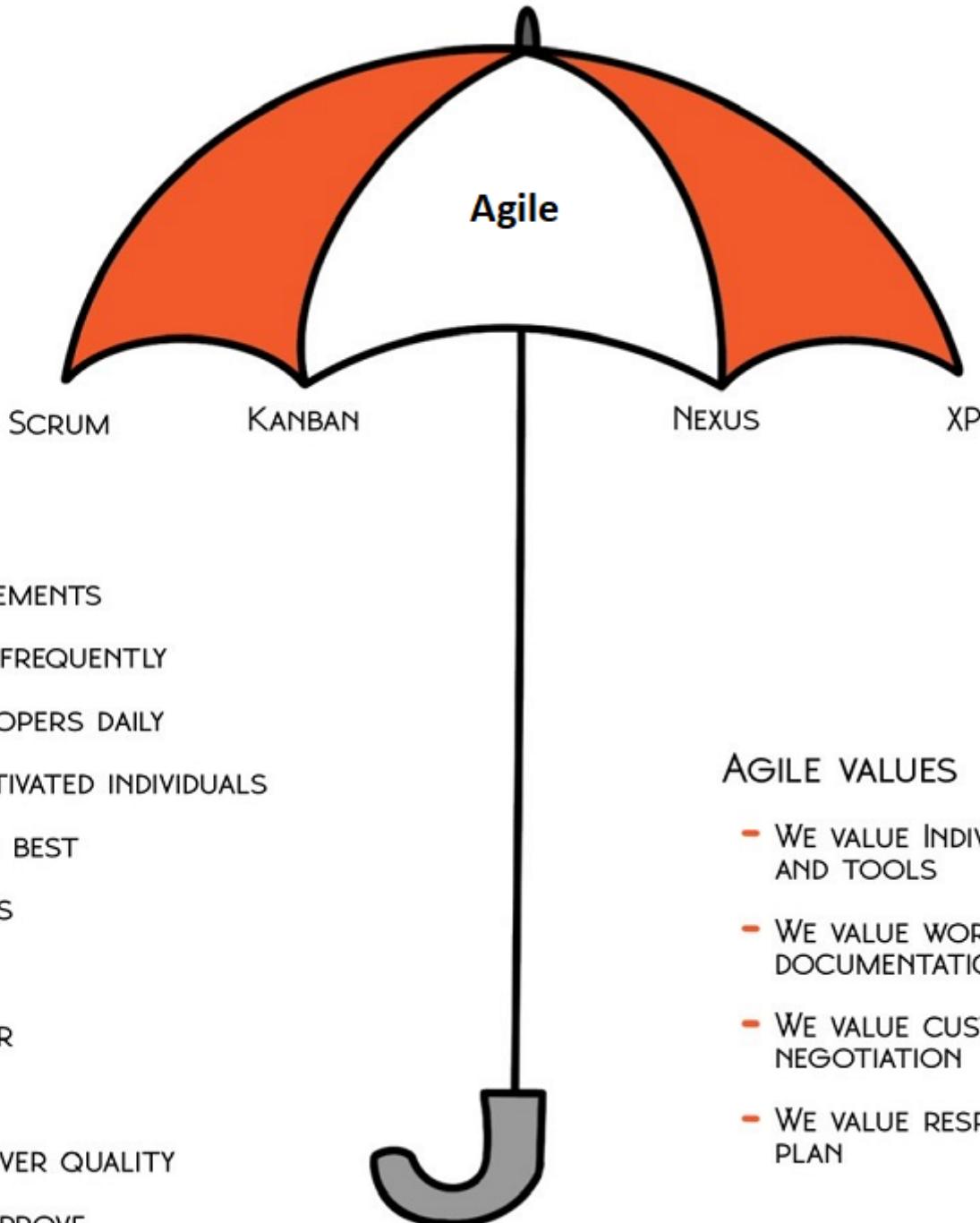


**That is, while there is value in the items on the right, we value the items on the left more.**

|                              |      |                             |
|------------------------------|------|-----------------------------|
| Individuals and interactions | over | Processes and tools         |
| Working product              | over | Comprehensive documentation |
| Customer collaboration       | over | Contract negotiation        |
| Responding to change         | over | Following a plan            |

## 12 PRINCIPLES

- SATISFY THE CUSTOMER
- WELCOME CHANGING REQUIREMENTS
- DELIVER WORKING SOFTWARE FREQUENTLY
- BUSINESS WORKS WITH DEVELOPERS DAILY
- BUILD PROJECTS AROUND MOTIVATED INDIVIDUALS
- FACE-TO-FACE CONVERSATION BEST
- WORKING SOFTWARE MATTERS
- CONSTANT PACE
- HIGH QUALITY CODE IS FASTER
- MAXIMIZE WORK NOT DONE
- SELF-ORGANIZING TEAMS DELIVER QUALITY
- FEEDBACK LOOPS USED TO IMPROVE

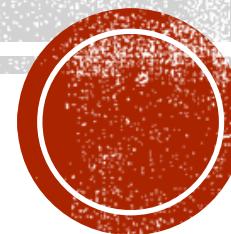


## AGILE VALUES

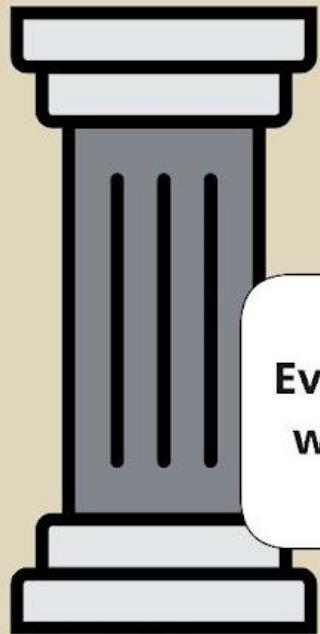
- WE VALUE INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WE VALUE WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- WE VALUE CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- WE VALUE RESPONDING TO CHANGE OVER FOLLOWING A PLAN



# **SCRUM PILLARS & VALUES**

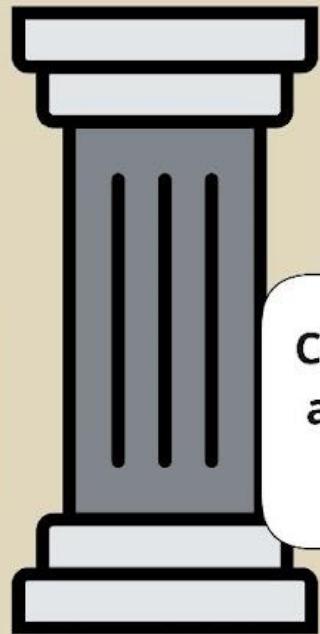


# Scrum - Empiricism



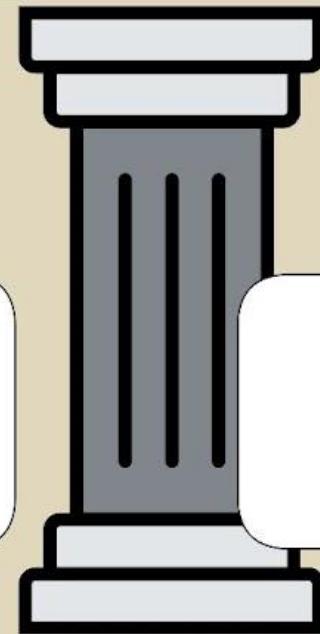
Transparency

Everybody knows  
what's going on



Inspection

Check your work,  
as you are doing  
it.

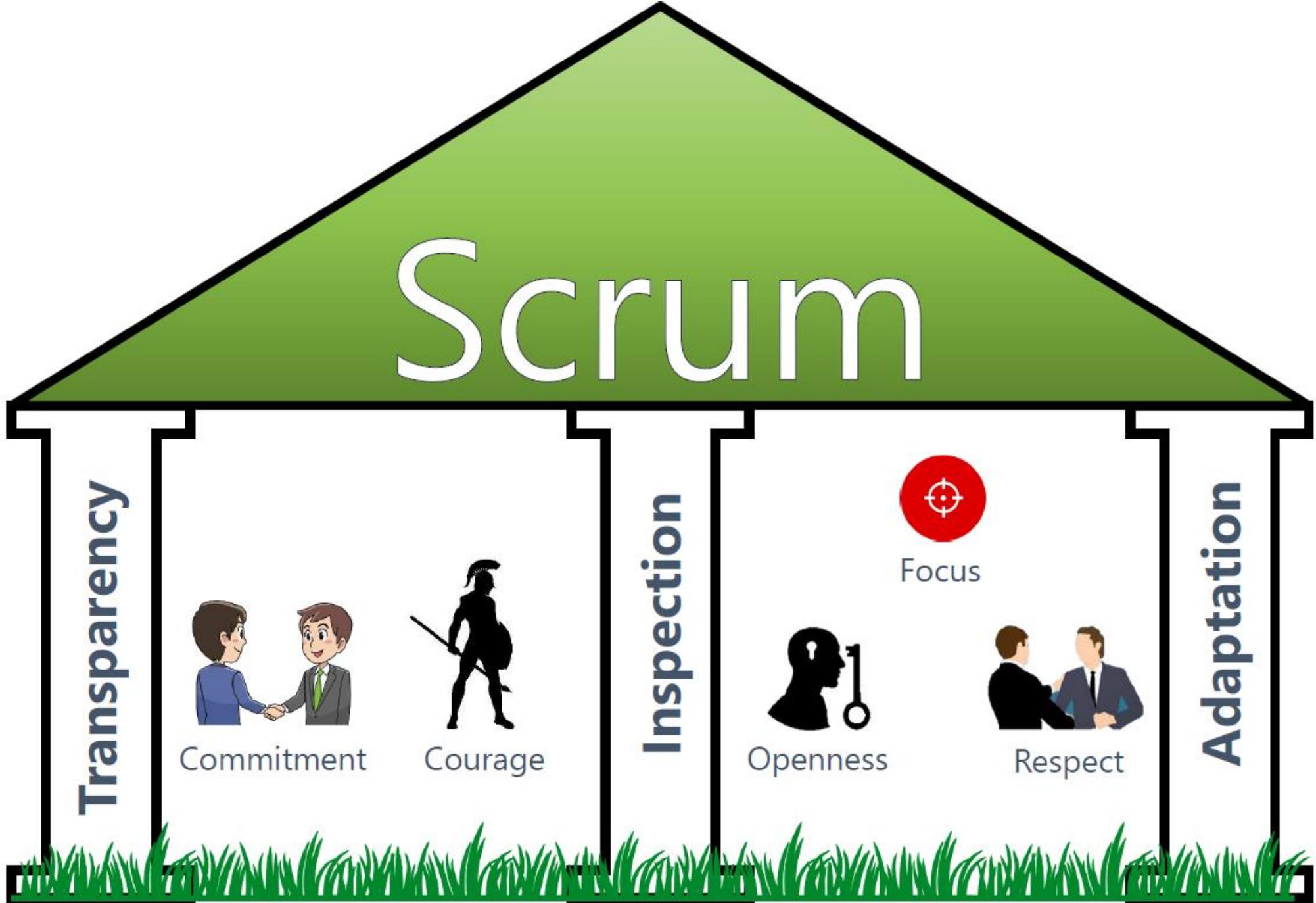


Adaptation

Adjust the  
direction.



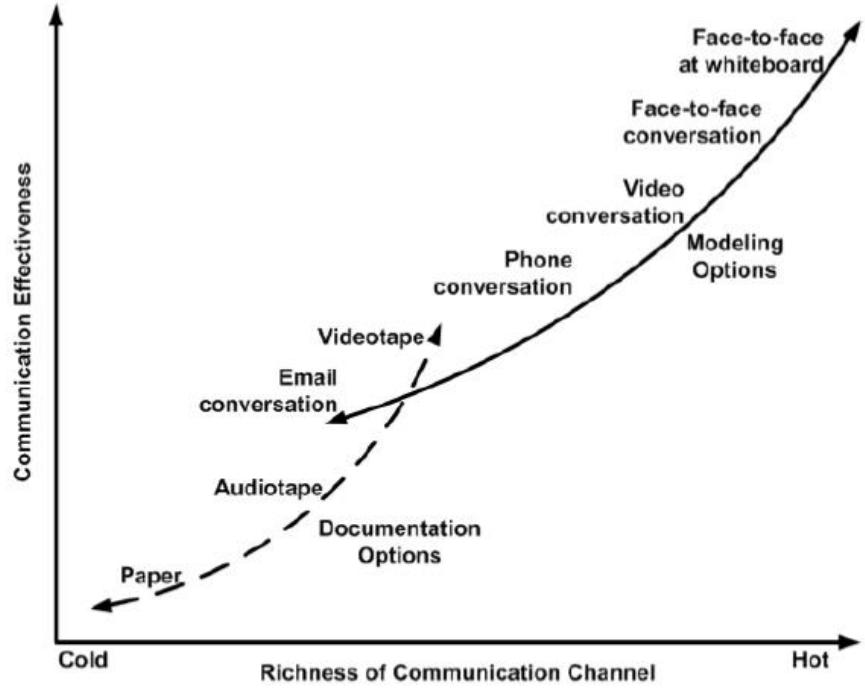
# Scrum



# Simplicity

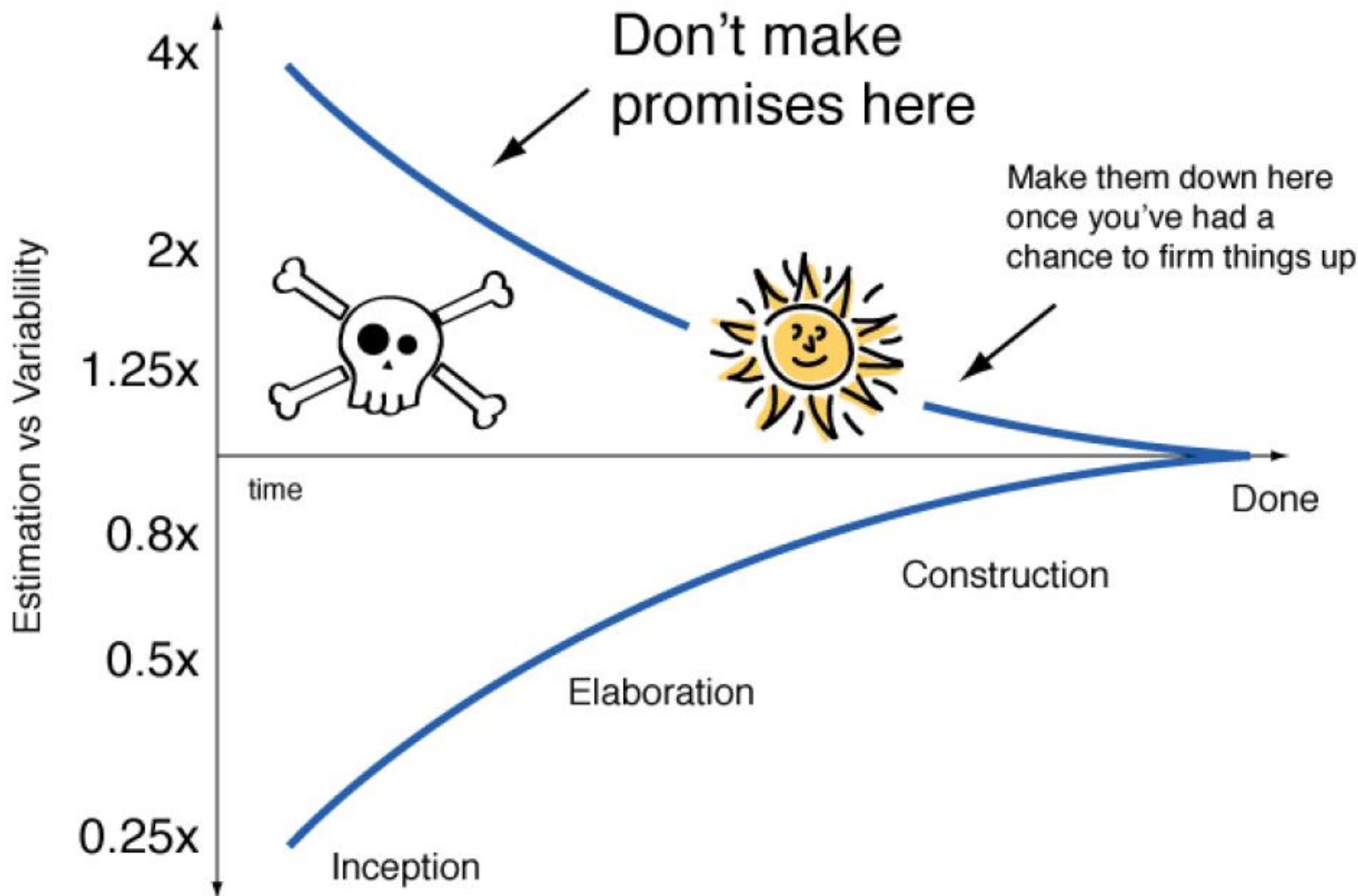


# Face-to-face communication



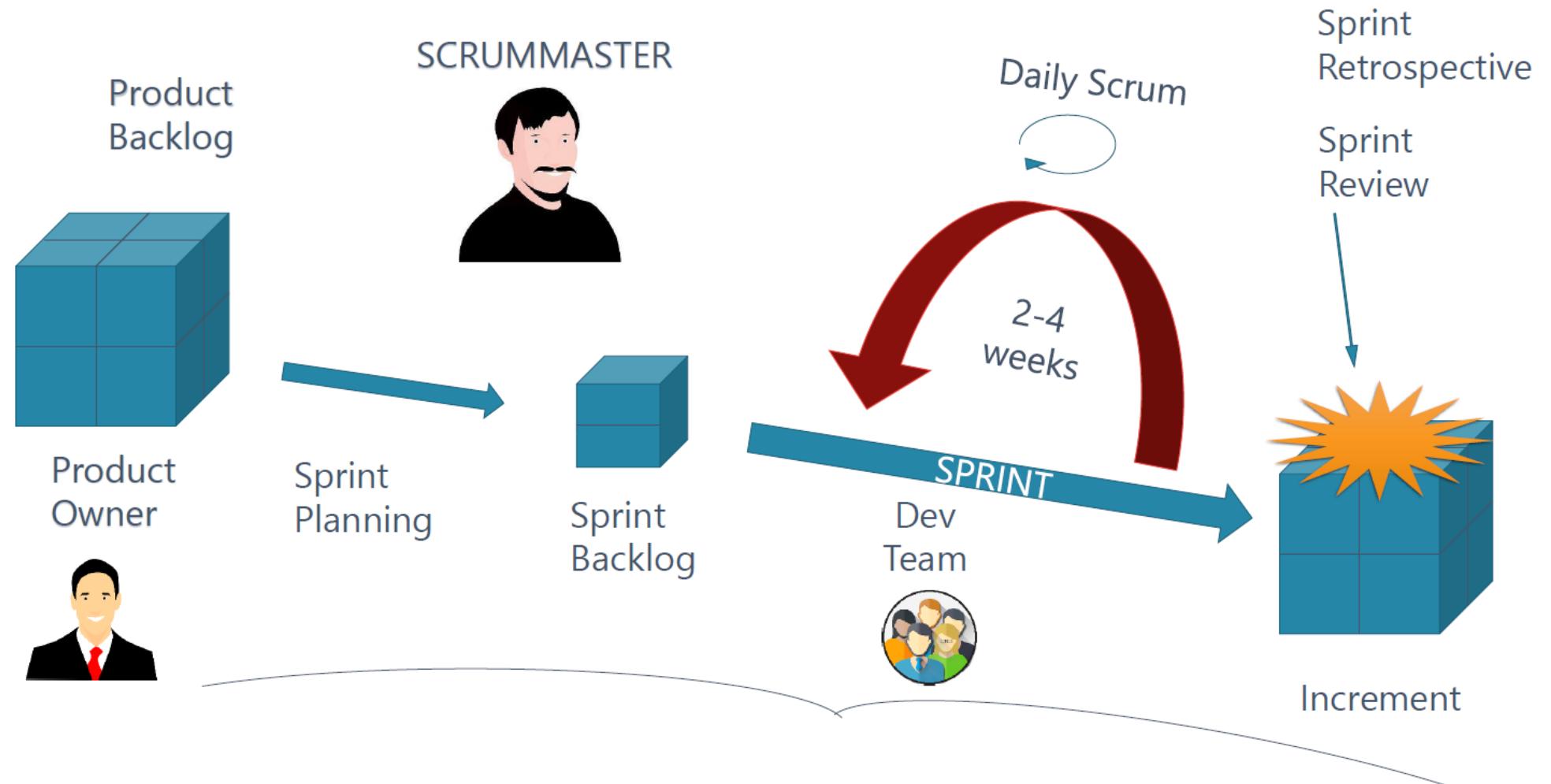
# Cone of Uncertainty

---



# **SCRUM PROCESS FLOW**



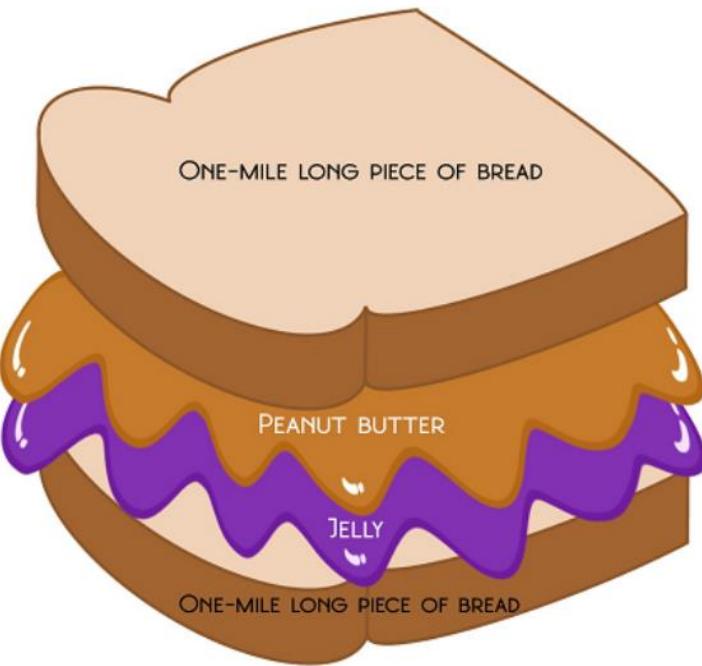


**ITERATIVE**

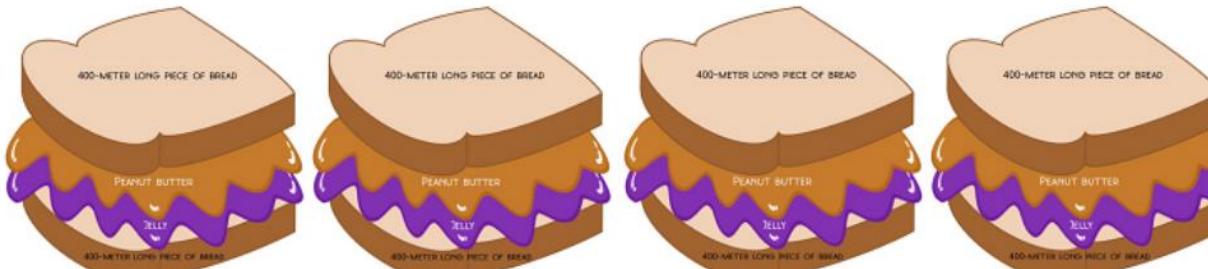


**INCREMENTAL**



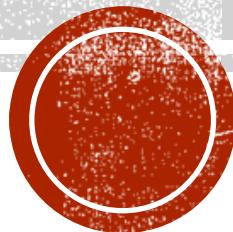


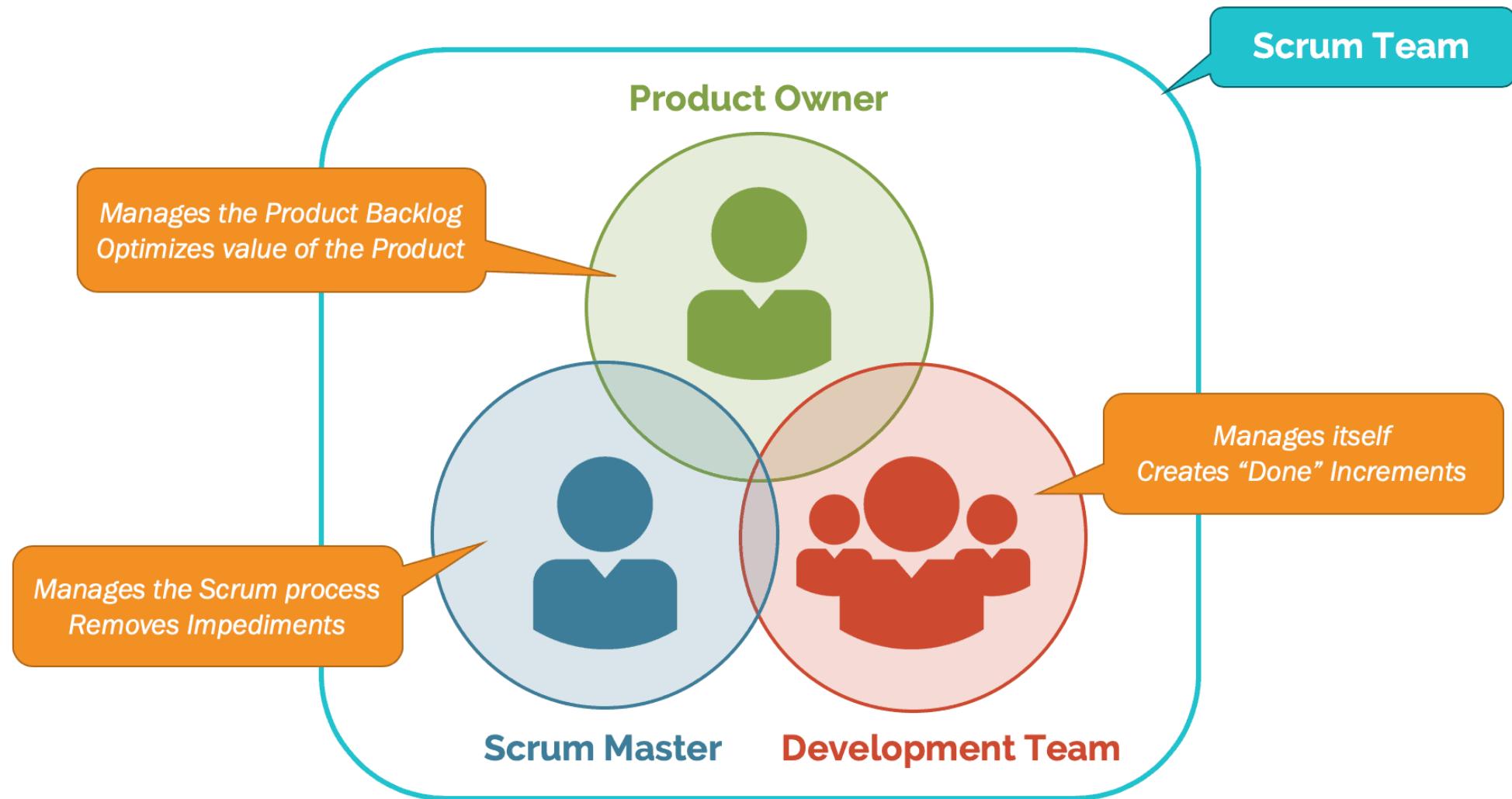
WITH A WATERFALL APPROACH THE TEAM MIGHT FIRST BAKE ALL THE BREAD AND THEN SMOOTH ON ALL THE PEANUT BUTTER AND JELLY. THE CUSTOMER DOESN'T RECEIVE ANYTHING UNTIL THE ENTIRE SANDWICH IS DONE.



USING THE SCRUM FRAMEWORK, TEAM WOULD APPROACH THE WORK IN SMALLER BITES WITH THE GOAL OF DELIVERING A PIECE OF THE WHOLE EACH SPRINT. THE TEAM MIGHT BAKE BREAD AND SMOOTH ON PEANUT BUTTER FOR A SMALLER PIECE OF THE SANDWICH USING THE INCREMENTAL APPROACH TO DELIVERY. THE PRODUCT OWNER MAY CHOOSE TO RELEASE THE SANDWICH TO THE CUSTOMER A BITE AT A TIME.

# **SCRUM ACCOUNTABILITIES**







# Product Owner: Wrap-up

---

- “Is accountable for maximizing the value of the product resulting from work of the Scrum Team”
- “is the sole person accountable for managing the Product Backlog”
- Others might do work related to Product Backlog management, but the Product Owner remains accountable
- **Product Owner must have authority**

# The Developers



- Transform Product Backlog items into a usable Increment
- Cross-functional
- Self-managed
- Typically, composed of ten or fewer members

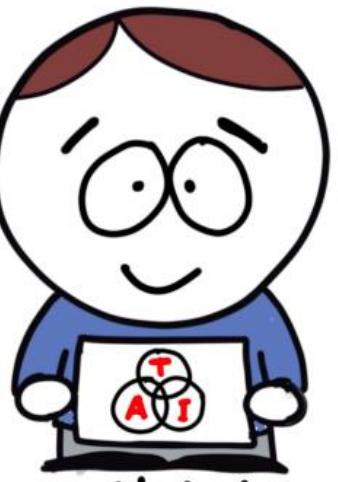
# Scrum Master



- Accountable for establishing Scrum
- Causes the removal of impediments
- Leads and facilitates Scrum adoption



impediment  
remover



facilitator



coach

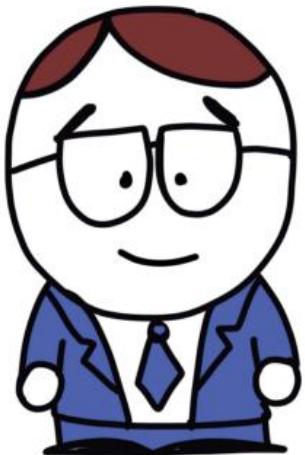


teacher

# SCRUM MASTER!



servant leader



manager



change agent



mentor



# **IMPROVEMENT KATA**

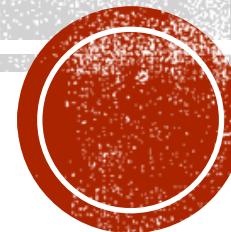
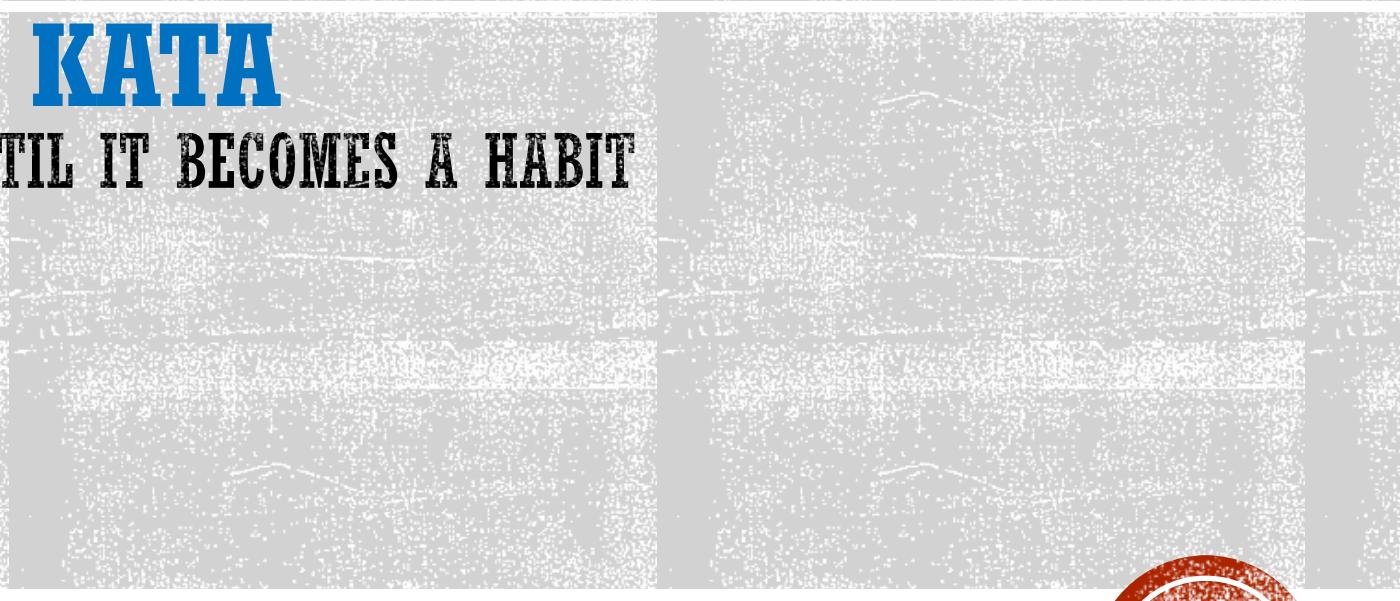
PRACTICE IMPROVEMENT UNTIL IT BECOMES A HABIT

## **SHU-HA-RI**

FOLLOW THE RULE

BREAK THE RULE

BE THE RULE



## **UNCONSCIOUS INCOMPETENCE**

You are unaware of the skill  
and your lack of proficiency

## **UNCONSCIOUS COMPETENCE**

Performing the skill  
becomes automatic

## **CONSCIOUS INCOMPETENCE**

You are aware of the skill but  
not yet proficient

## **CONSCIOUS COMPETENCE**

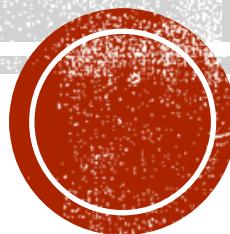
You are able to use the skill,  
but only with effort

Give a man a fish and you feed him for a day.  
Teach a man how to fish and you feed him for a lifetime.

-LAO TZU



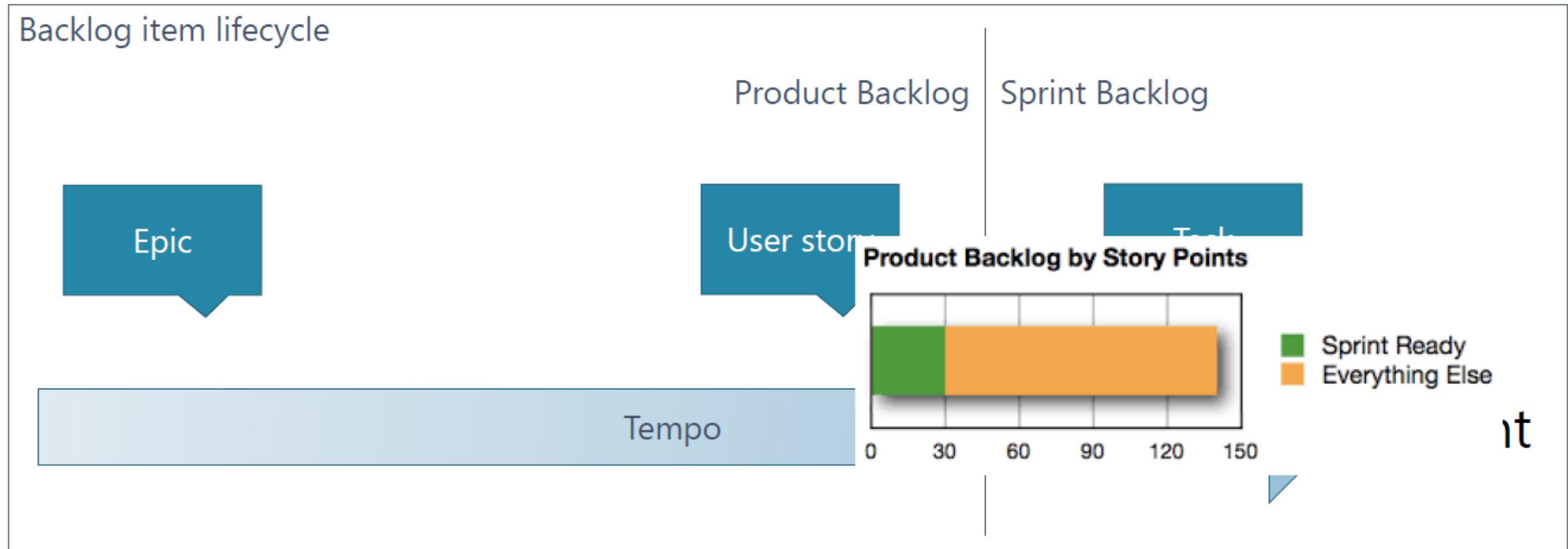
# **SCRUM ARTIFACTS**



# Product Backlog

---

- An emergent, ordered list of what is needed to improve the product



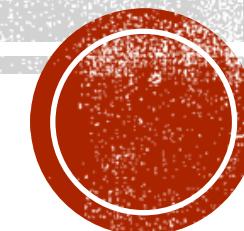
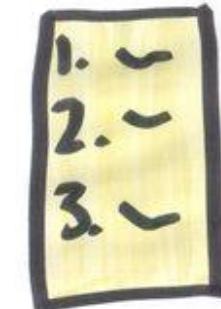
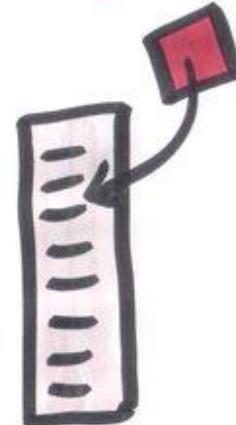
# D. E. E. P

Detailed  
appropriately

Estimated

Emergent

Prioritised



# VARIOUS METHOD TO PRIORITIZE BACKLOG

**KANO Model : Mandatory, Linear, Exciter**

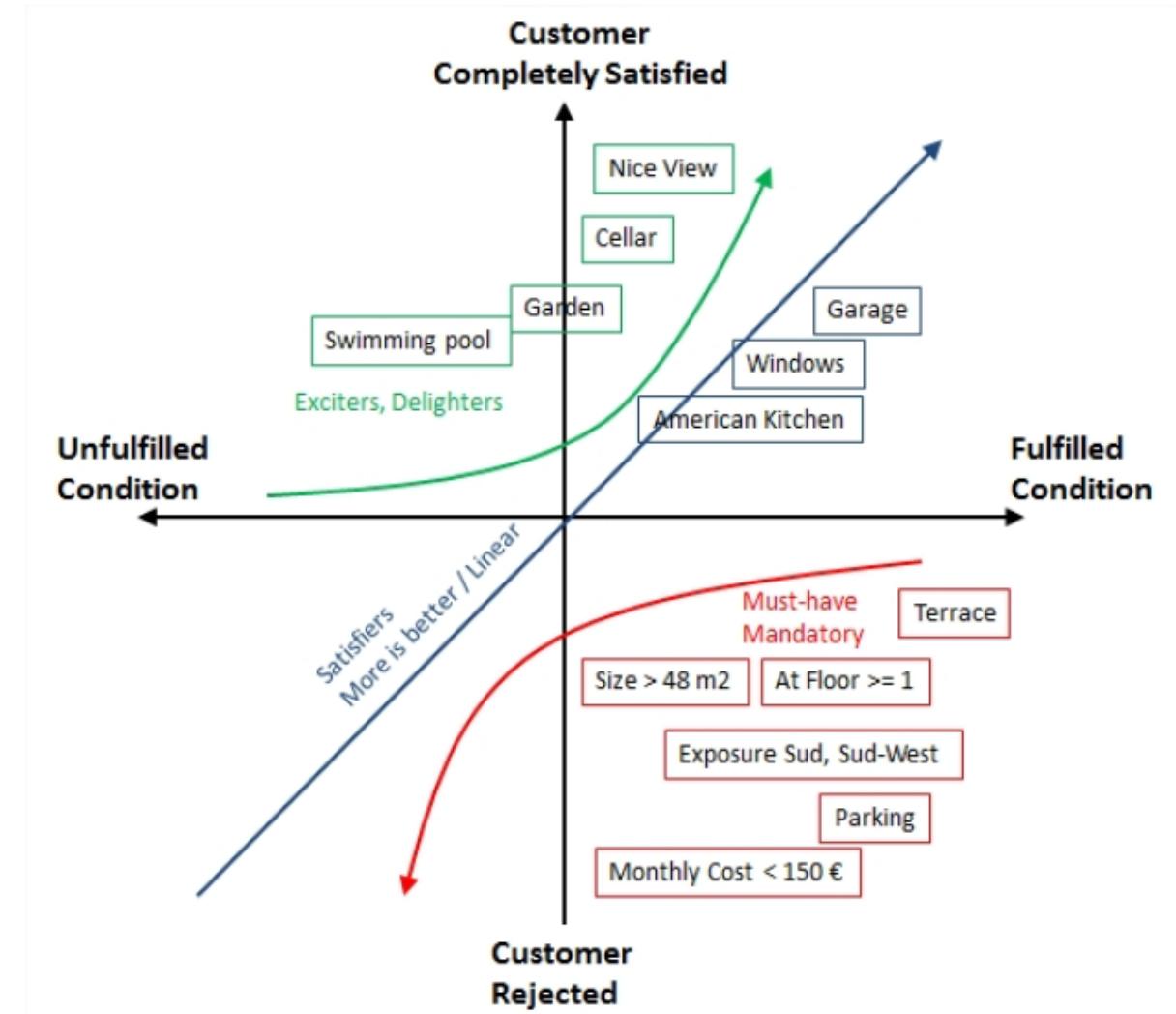
**MoSCoW : Must have, Should have, Could have, Won't have**

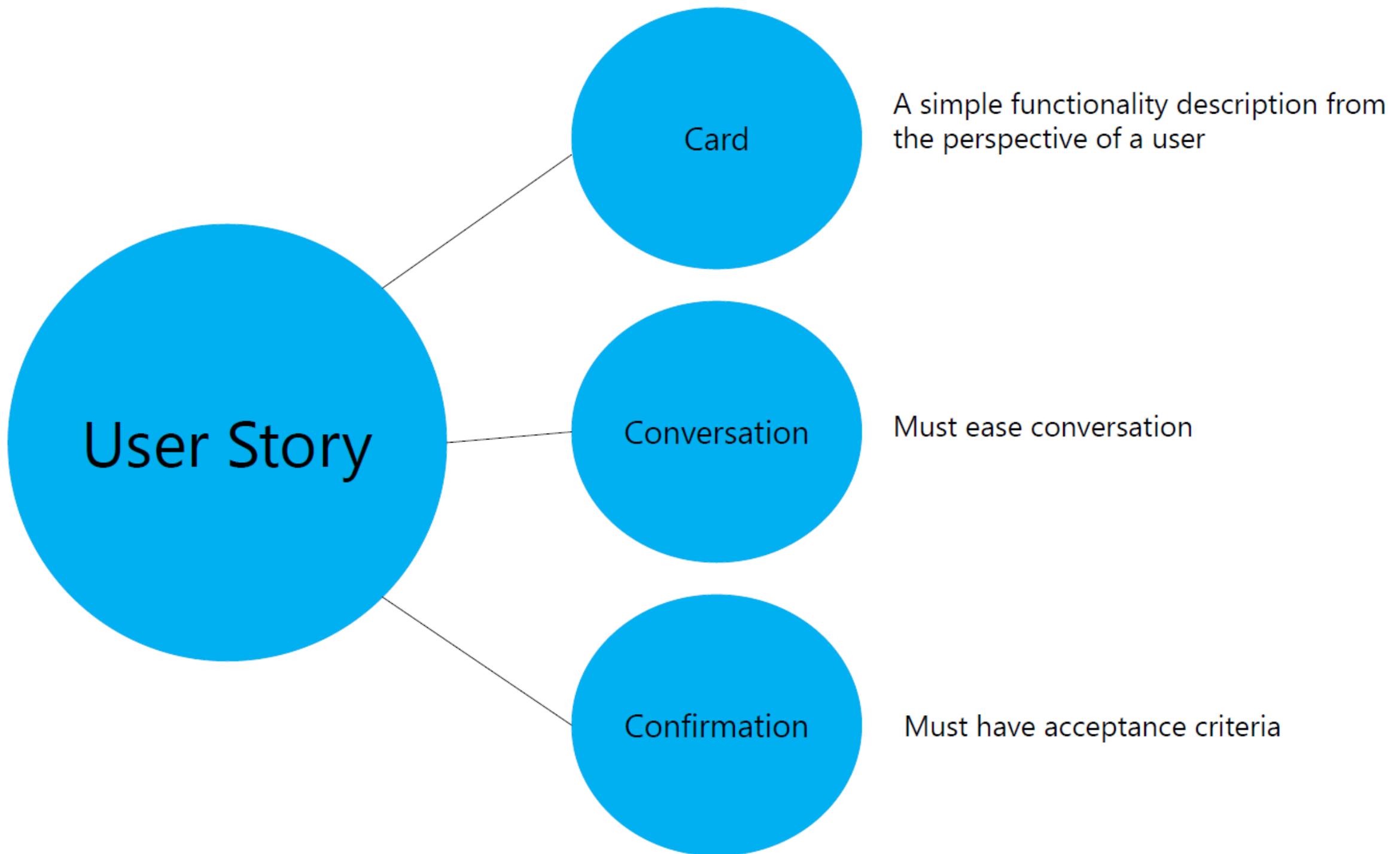
## WSJF

| Feature | User Business Value | Time Criticality | Risk Reduction | COD        | Job Size | WSJF       |
|---------|---------------------|------------------|----------------|------------|----------|------------|
| XYZ     | 1                   | 1                | 1              | 1+1+1 =3   | 1        | 3/1 =3     |
| YZA     | 3                   | 5                | 3              | 3+5+3 = 11 | 5        | 11/5 = 2.2 |

# Desired-based prioritization (Kano Model)

- Features as one of three categories
  - Mandatory
    - Basic stuff. Its presence won't add much value, but its absence has a tremendous negative impact
  - Linear
    - The more, the better. The more linear features, the more customers pay!
  - Exciter
    - Unknown desires





**I**ndependent      Standalone PBI with no dependencies.

**N**egotiable      It can be changed in anytime.

**V**aluable      Having a good value for the end user.

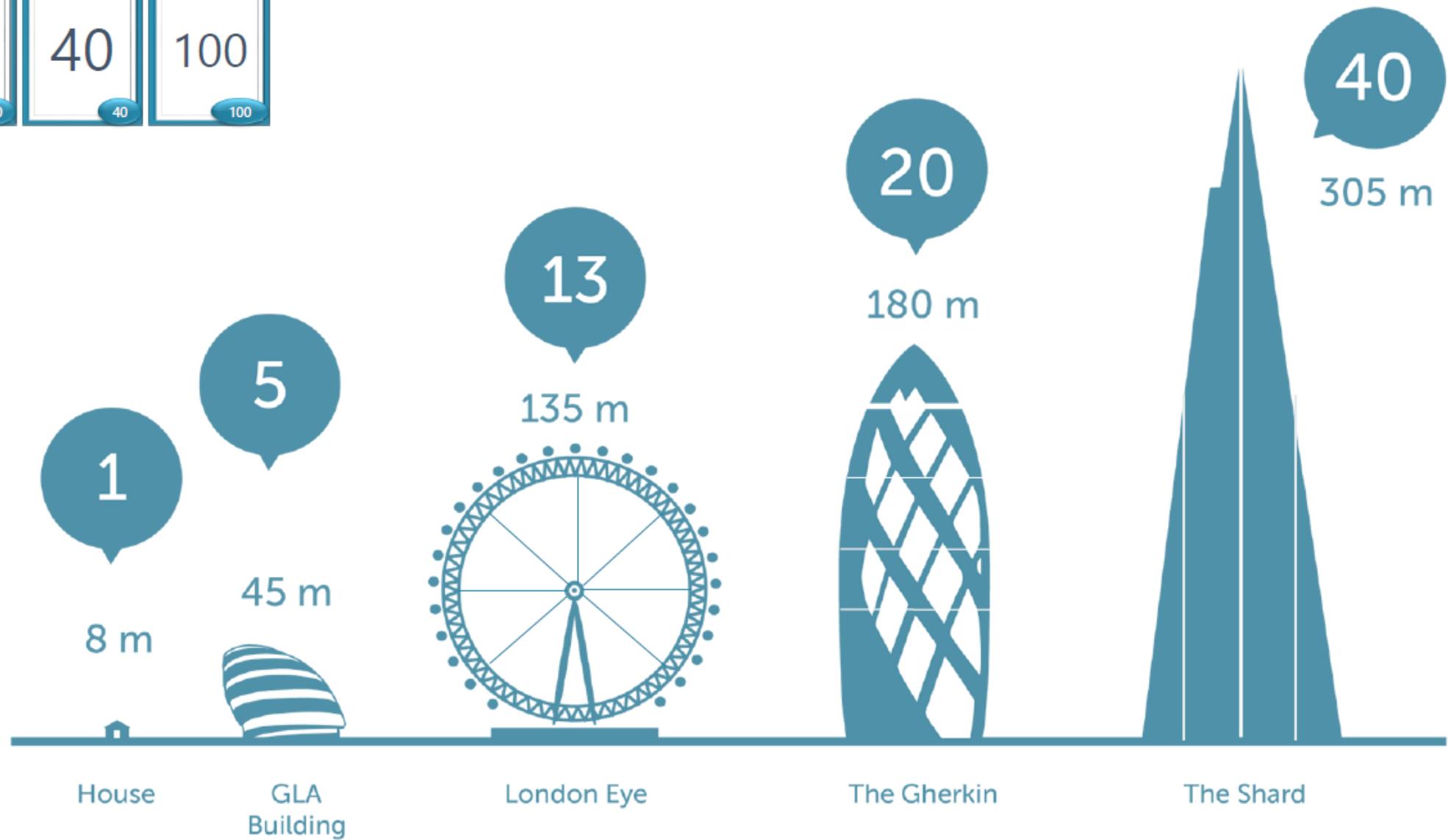
**E**stimable      The team is able to estimate its size.

**S**mall      Small enough to be developed and tested.

**T**estable      Testing is possible from AC and DOD.



|   |    |    |    |     |
|---|----|----|----|-----|
| 0 | 1  | 2  | 3  | 5   |
| 0 | 1  | 2  | 3  | 5   |
| 8 | 13 | 20 | 40 | 100 |



# WHY NOT MAN HOURS FOR ESTIMATION

- Story points doesn't decay with time.
- Team level comparison.
- Man hours encourages I can/Me can tendency in team.
- In Man hours people think from their perspective/not the teams perspective.

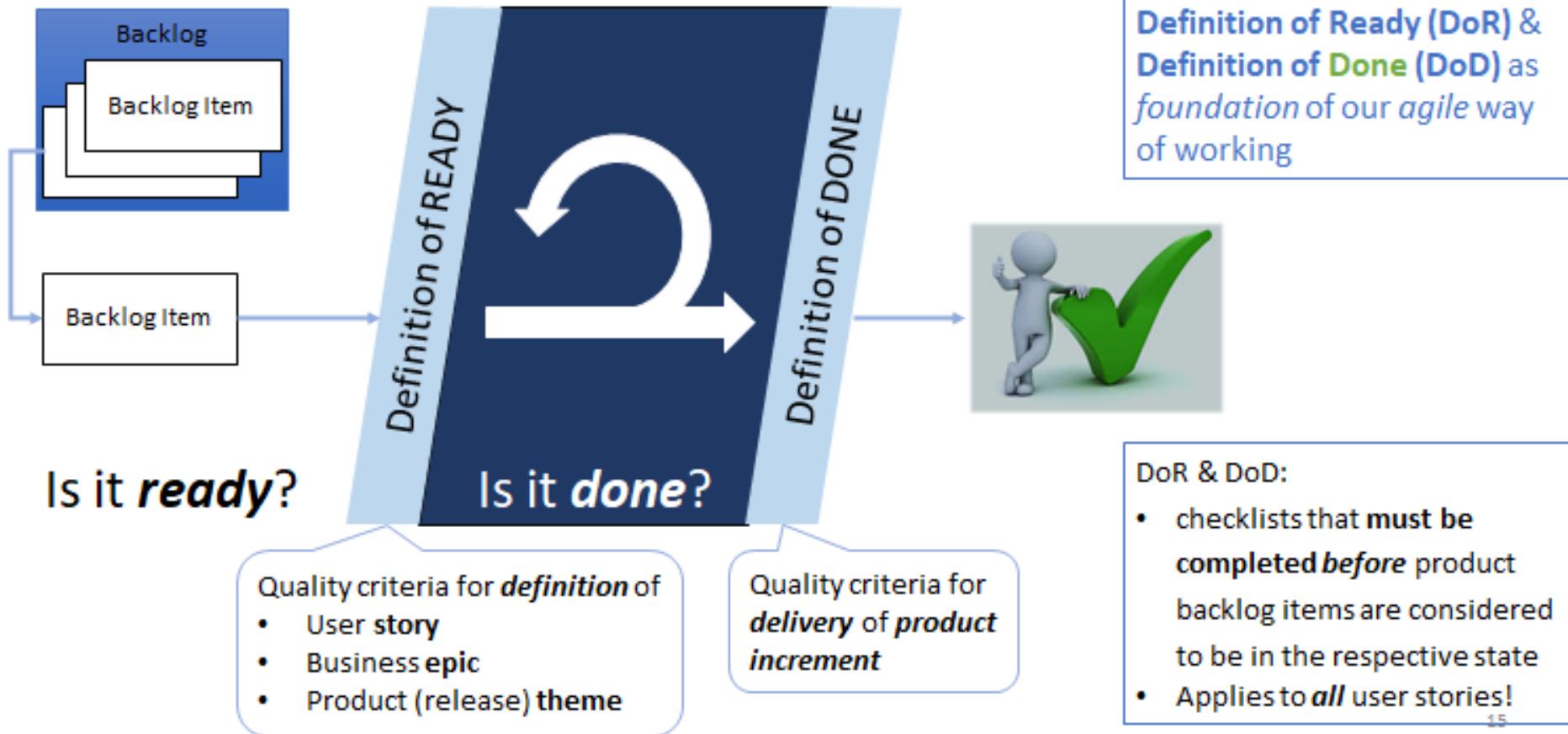


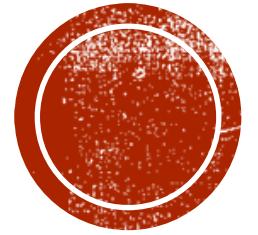
# Product Backlog Refinement



- Is the act of breaking down and further defining Product Backlog items into smaller more precise items
- Usually, it is what enables getting the Product Backlog Items Ready
- The Scrum Team decides **how** and **when** refinement is done

## *Quality Built-In* by implementing DoR & DoD

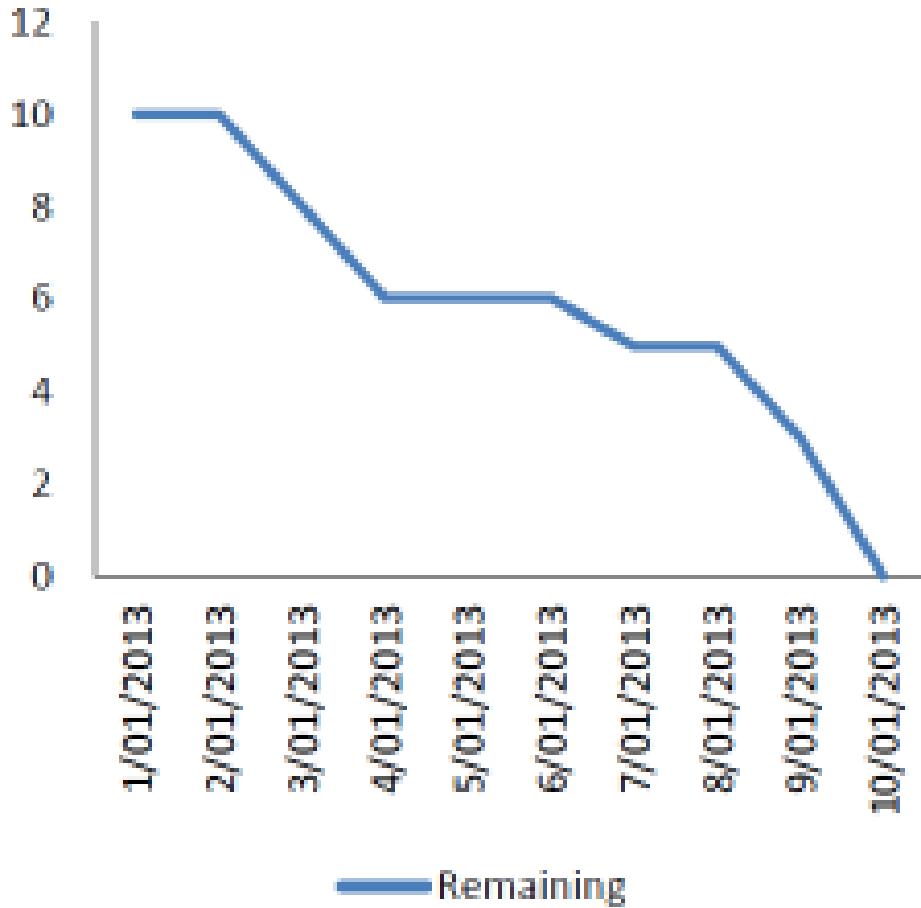




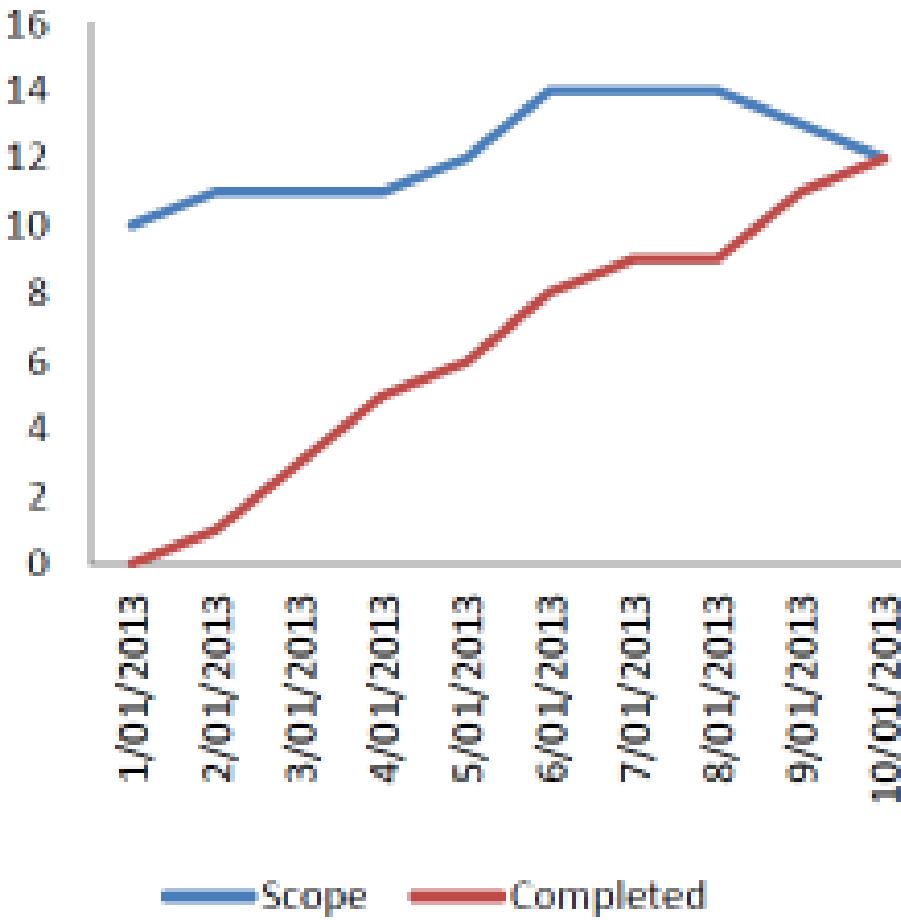
# REPORTS



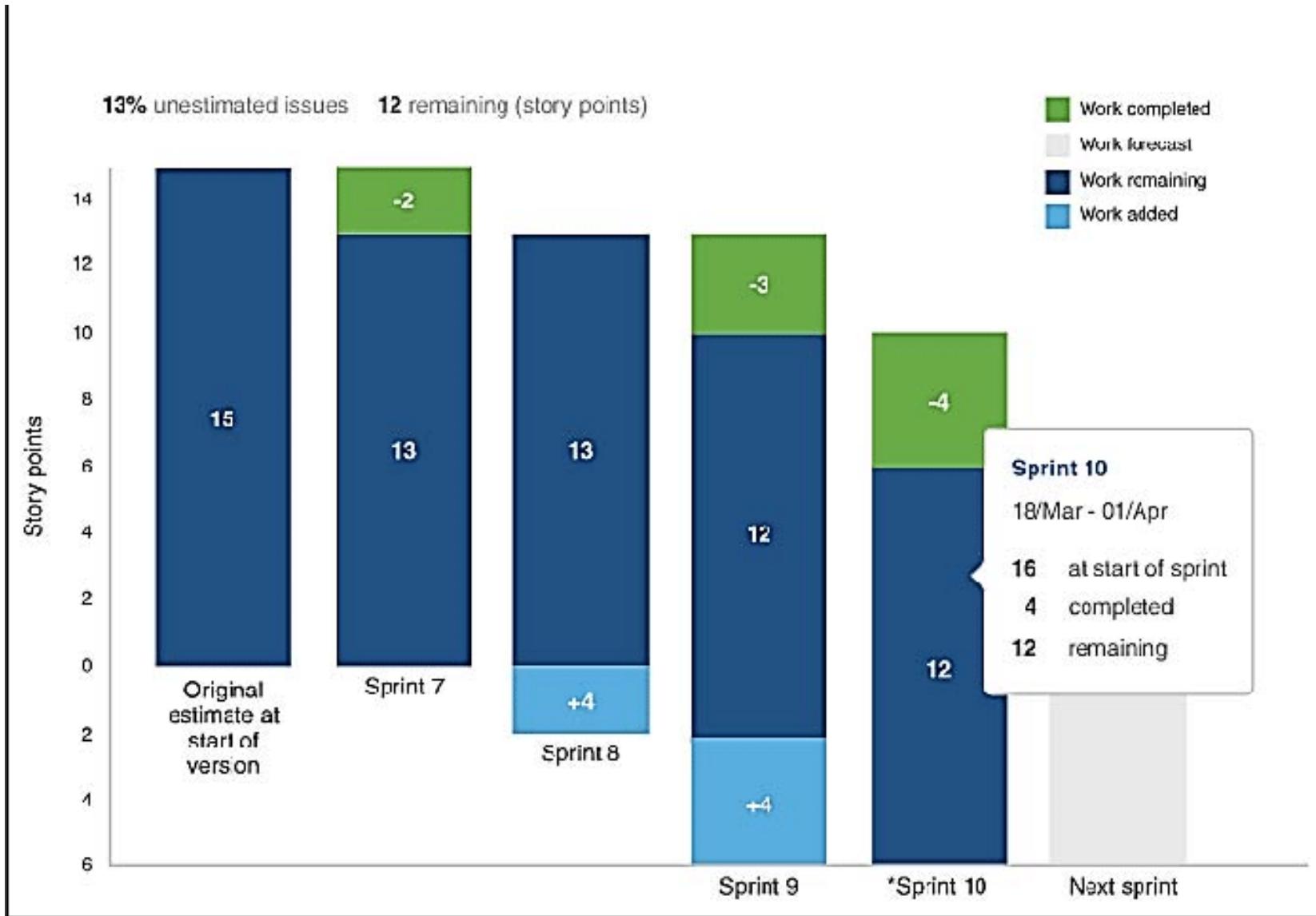
## Burn down



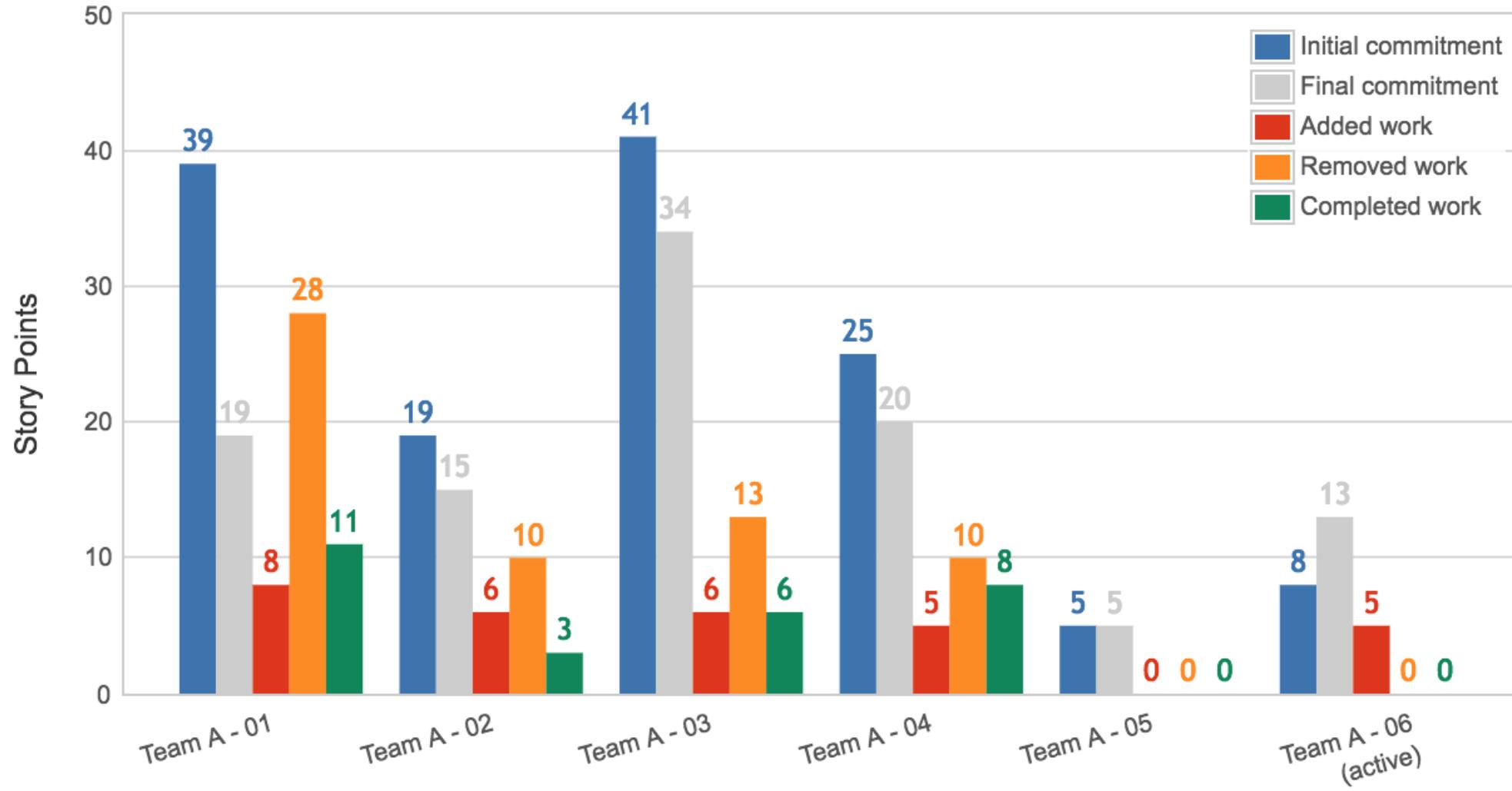
## Burn up



# EPIC Burndown



## Team A - Velocity Chart

 Export Support

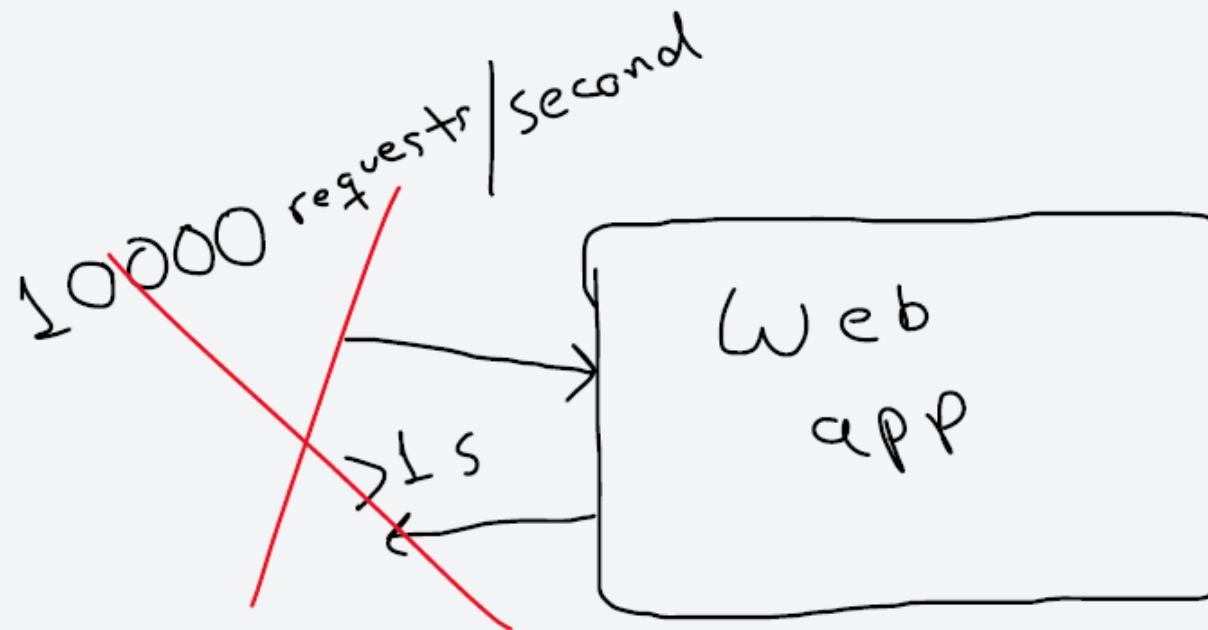
- **Cumulative Flow Diagram**
- **Build Failures**
- **Cyclomatic Complexity**
- **Code duplication**
- **% Code test Automation**



# Technical Debt

---

- Consequences of poor software development practices
- Might lead to code decay and architecture deterioration
- Scrum can manage it
- It is not necessarily bad



# **Scrum Ceremonies**



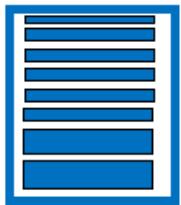
## Inputs

## Sprint Planning

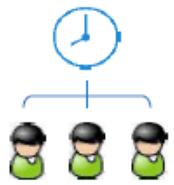
## Outputs



Value increase opportunities



Product Backlog



Capacity



Past performance



Definition of Done



Invited Stakeholders



Scrum Team

# How?



Sprint Goal



Selected PBI  
+  
Plan

Sprint Backlog



8h1month

# The Sprint



- “The heart of Scrum”
- Time-boxed to, at most, one month
- Must have fixed durations
- During the Sprint
  - No changes are made that would endanger the Sprint Goal
  - Quality goals do not decrease
  - The Product Backlog is refined as needed
  - Scope may be clarified and re-negotiated with the Product Owner as more is learned.

# \* Sprint \*



- Hardening (or Integration or Stabilization) Sprint
- Sprint 0
- Release Sprint
- Design Sprint

# Daily Scrum

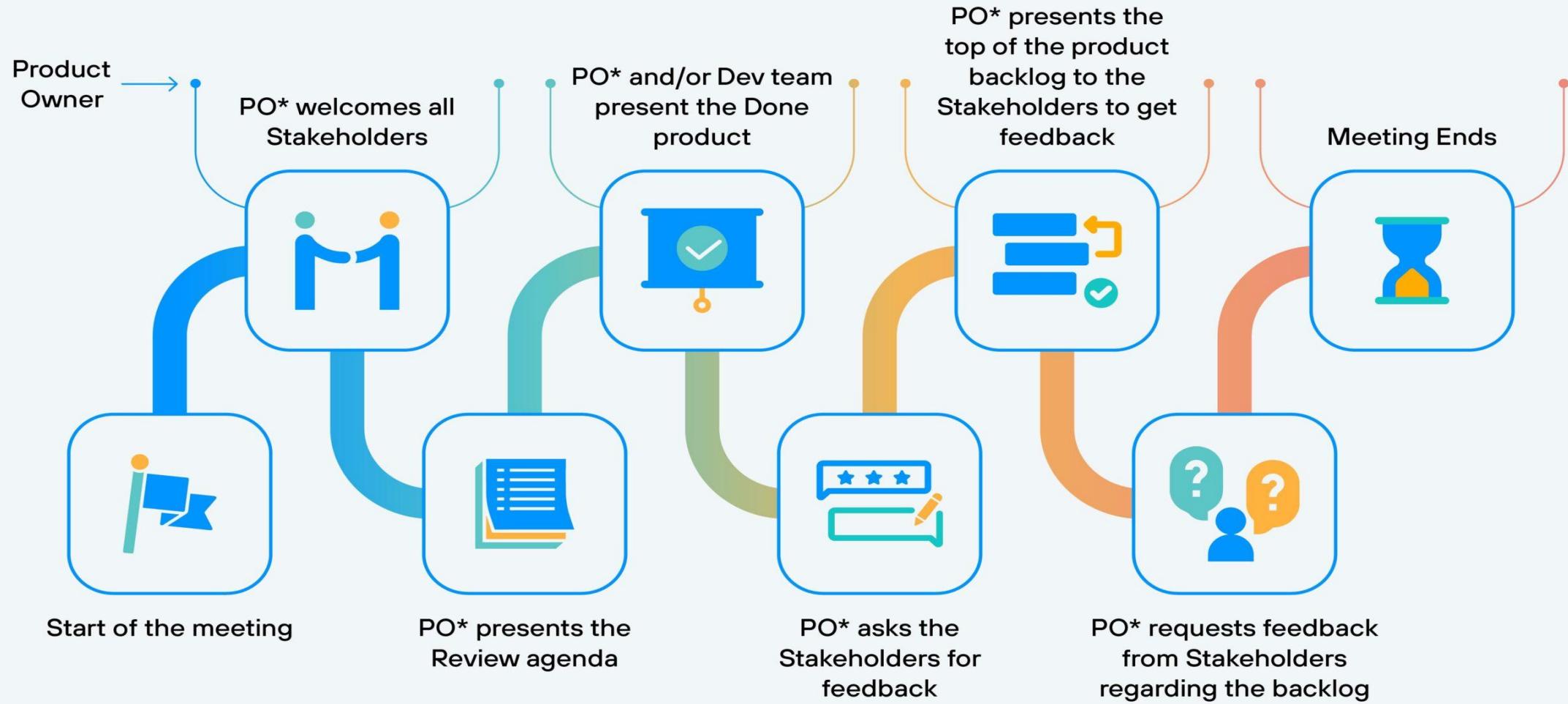
---

Since last meeting?  
Until next meeting?  
Impediments



15min

# Sprint Review Meeting



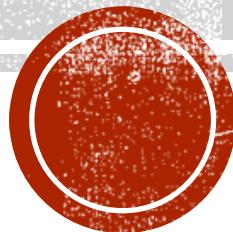
# Sprint Retrospective



- **Mad, Sad & Glad**
- **4Ls → Liked, Learned, Lacked & Longed**
- **Starfish**
- **Sailboat → Wind, Anchor & Islands**
- **KALM → Keep, Add, Less & More**
- **SWOT**
- **Classic Way**
- **Scrum Values**



# **EXTRA TOPICS**



# **DEVELOPMENT APPROACH:**

**TDD – Test driven development**

**BDD – Behavior driven development**

**ATDD – Acceptance Test driven development (3 Amigo meetings)**



# **TEAM HEALTH CHECK:**

**Atlassian Team Health Monitor**

**Spotify Squad Health Check**



# **AGILE REQUIREMENTS: VUCA WORLD**

**V: Volatility**

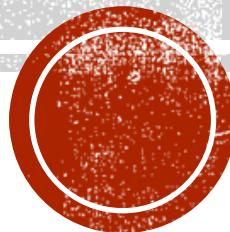
**U: Uncertainty**

**C: Complexity**

**A: Ambiguity**



# **BEST PRACTICES TO REDUCE BAD ESTIMATIONS**



# **REDUCE BAD ESTIMATIONS**

## **1. Reduce Estimated Work:**

**Predict Less**

**Commit Less**

## **2. Keep Estimates with the Estimator:**

**New person new estimates**

## **3. Estimate work as components**

## **4. Choose familiar technologies**

## **5. Find native integrations**

## **6. Never estimate in Man hours**



# Cost of Quality

## Cost of Good Quality

### Prevention Costs

- Planning
- Process Control
- Quality Audits
- Supplier Evaluation
- Training
- Design Review
- Risk Assessment
- FMEA

### Appraisal Costs

- Inspection
- FAT
- Document Review
- Quality Audits
- Calibration
- Test Materials
- Test Product

## Cost of Poor Quality

### Internal Failure Costs

- Scrap
- Rework
- Missing Docs
- Problem Solving
- 100% Sorting
- Retest
- Redesign
- Downgrading
- Variation
- Downtime

### External Failure Costs

- Warranty Charges
- Complaints
- Returned Materials
- Late Delivery Penalties
- Rework After Installation
- Lost Opportunities

# Professional Scrum Master I Cheat Sheet

|   |  |  |
|---|--|--|
| <p>The diagram illustrates the Scrum House, a metaphor for Scrum. The roof is labeled "Scrum". Below it, four vertical columns represent the "Pillars" of Scrum: Transparency (left), Inspection (middle-left), Focus (middle-right), and Adaptation (right). Each pillar has associated "Values": Transparency is linked to Commitment and Courage; Inspection is linked to Openness and Respect; Focus is linked to Focus; and Adaptation is linked to Adaptation.</p>  | <p><b>“founded on Lean thinking and Empiricism”</b></p> <p><b>Artifacts and their Commitments</b></p> <p>Product Backlog → Product Goal</p> <p>Sprint Backlog → Sprint Goal</p> <p>Increment → Definition of Done</p> <p style="color: red;"><b>Mandatory!</b></p>   | <p><b>Events</b></p> <p>Formal opportunities for Inspect &amp; Adapt</p> <ul style="list-style-type: none"> <li>• <b>Sprint</b> — max 1 calendar month — Manage risks</li> <li>• <b>Sprint Planning</b> — max 8 hours (usually shorter for shorter Sprints) — The Scrum Team — <u>initiates</u> Sprint.</li> <li>• <b>Daily Scrum</b> — max 15 minutes — The Development Team</li> <li>• <b>Sprint Review</b> — max 4 hours (usually shorter for shorter Sprints) — The Scrum Team and Key Stakeholders — <u>informal</u> session</li> <li>• <b>Sprint Retrospective</b> — max 3 hours (usually shorter for shorter Sprints) — The Scrum Team — <u>closes</u> Sprint.</li> </ul> |
| <p><b>Scrum in a nutshell</b></p> <ol style="list-style-type: none"> <li>1. PO orders Product Backlog (ideas)</li> <li>2. Scrum Team defines turns ideas into value durint a Sprint</li> <li>3. Scrum Team and stakeholder inspects the results and adjusts for the next Sprint.</li> <li>4. <i>Repeat</i></li> </ol> <p><b>Scaled Scrum</b></p> <ol style="list-style-type: none"> <li>1. A <b>single</b> PO. A <b>single</b> Product Backlog. A <b>single</b> Product Goal at a time</li> <li>2. Scrum Teams self-manage             <ol style="list-style-type: none"> <li>1. <b>Don't need to synchronize Sprints</b></li> <li>2. They decide how to handle dependencies</li> </ol> </li> </ol> | <p><b>Scrum Team</b></p> <ul style="list-style-type: none"> <li>• <b>Self-managed</b> and cross-functional</li> <li>• No sub-teams or hierarchies</li> <li>• Collective accountability                     <ul style="list-style-type: none"> <li>• All product related activities</li> <li>• Valuable, usable Increment, each Sprint</li> </ul> </li> </ul> <p><u>Accountabilities</u></p> <ul style="list-style-type: none"> <li>• Scrum Master                     <ul style="list-style-type: none"> <li>• Facilitator</li> <li>• Causes Impediment removal</li> </ul> </li> <li>• Product Owner                     <ul style="list-style-type: none"> <li>• Value maximiser</li> </ul> </li> <li>• Developers                     <ul style="list-style-type: none"> <li>• create any aspect of a usable Increment each Sprint.</li> </ul> </li> </ul> <p><u>Self-managed</u></p> <ul style="list-style-type: none"> <li>• Internally decide who does what, when, and how.</li> </ul> <p><u>Cross-functional</u></p> <ul style="list-style-type: none"> <li>• Have all the skills to create value each Sprint</li> </ul> | <p><b>During the Sprint</b></p> <ul style="list-style-type: none"> <li>• No changes are made that would endanger the Sprint Goal;</li> <li>• Quality goals do not decrease; and,</li> <li>• The Product Backlog is refined as needed</li> <li>• Scope may be clarified and re-negotiated between the Product Owner and Developers as more is learned.</li> <li>• Only <b>Product Owner</b> can <b>cancel</b> it.</li> </ul>  |
|   | <p><b>Product Backlog</b></p> <ul style="list-style-type: none"> <li>• Only one</li> <li>• Only one Product Goal</li> <li>• Estimated by Developers</li> <li>• Refined by Scrum Team. Continuously.</li> <li>• Pursue one Product Goal at a time.</li> </ul>   | <p><b>Increment</b></p> <ul style="list-style-type: none"> <li>• Can only be composed of “Done” work</li> </ul>  |



KANBAN

## What is Kanban

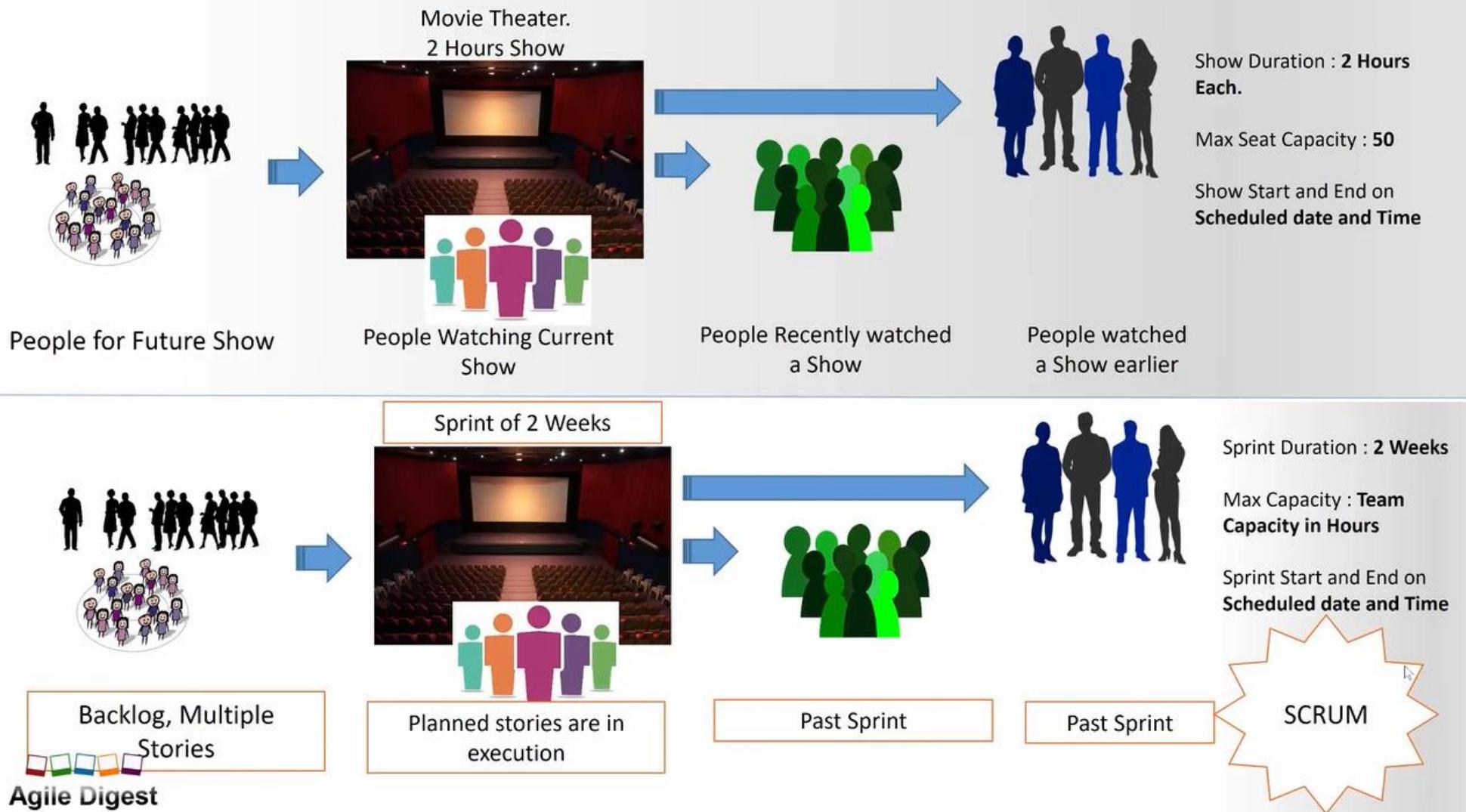
Kanban is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively.

This frame work is highly productive and effective to run

- Ad-hoc Requests,
- Unplanned works,
- Production Support etc.

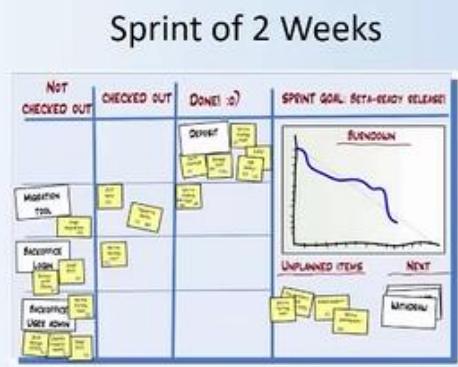


Its a Method of Visualize the flow of work.  
in order to balance demand with available capacity and spot bottlenecks





Backlog, Multiple Stories



Planned stories are in execution



Past Sprint



Past Sprint

Sprint Duration : 2 Weeks

Max Capacity : Team Capacity in Hours

Sprint Start and End on Scheduled date and Time



Active Backlog



Board Duration : No Defined Show Time

Max Capacity : Unlimited

Start and End on : No Defined start and End Time

Board Rule : Max 6 stories at a time.



VISUALIZE THE FLOW  
OF WORK

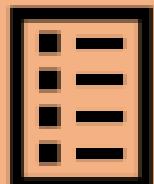


FOCUS ON FLOW



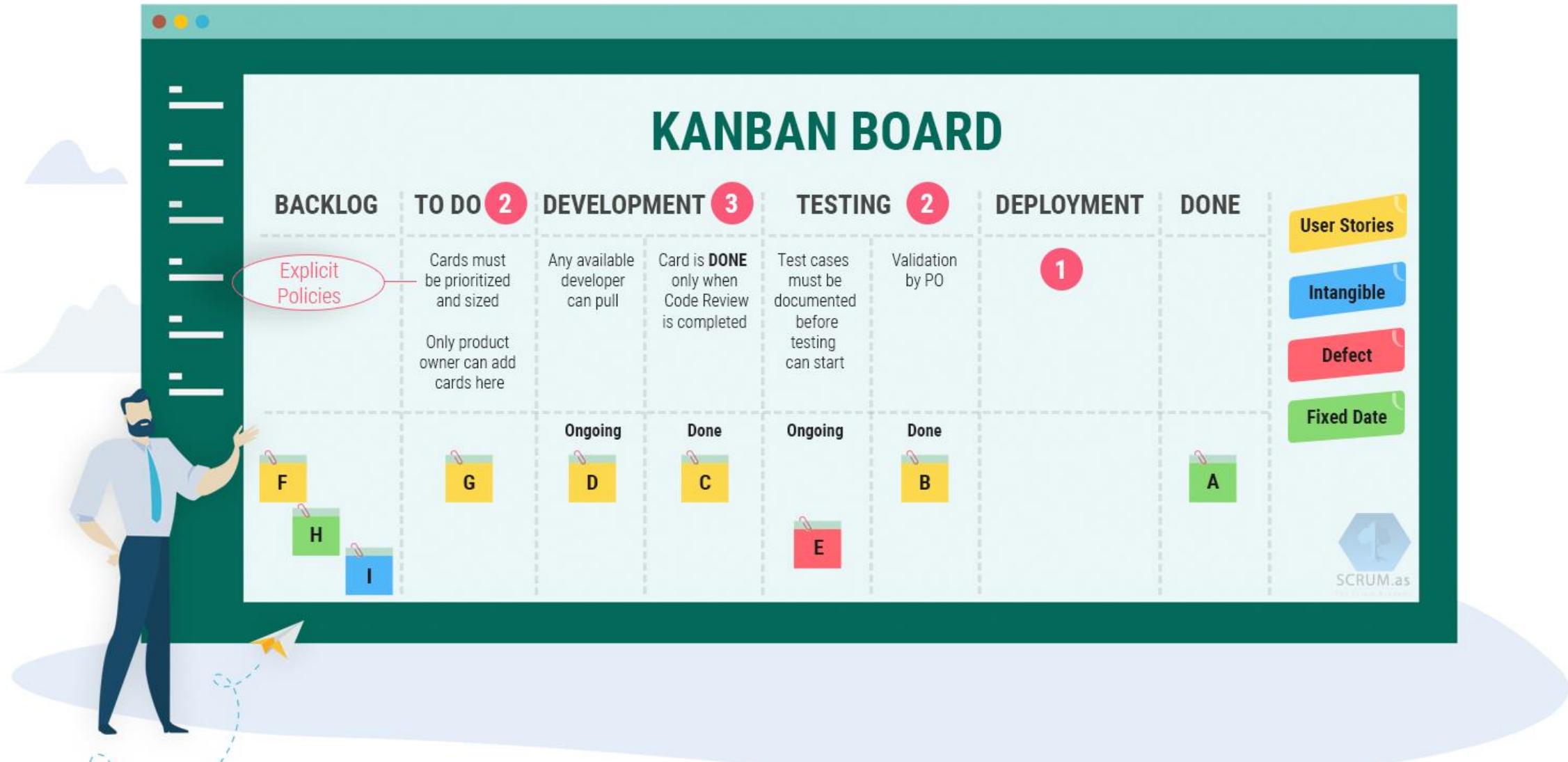
KANBAN  
PRINCIPLES

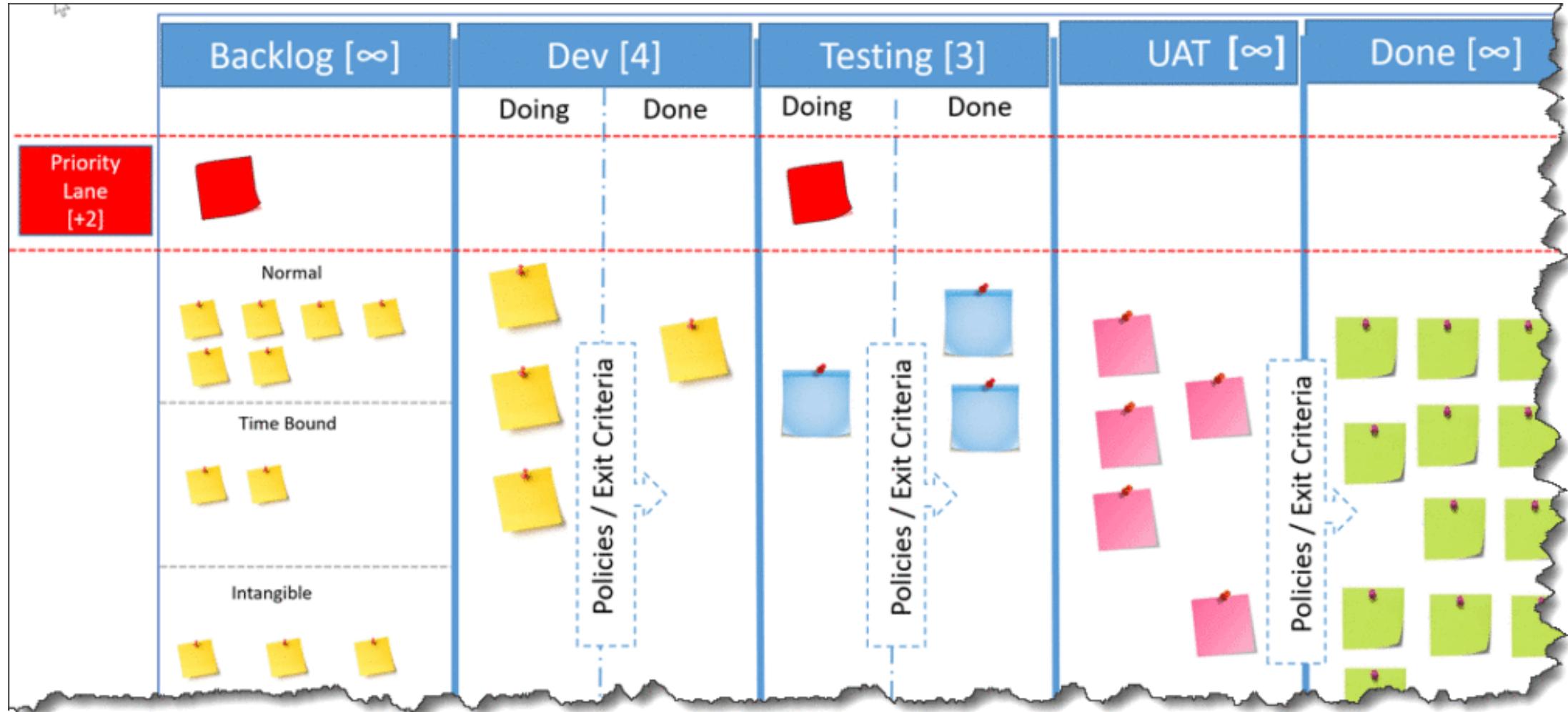
LIMIT WORK IN  
PROCESS (WIP)



CONTINUOUSLY  
IMPROVE







### Scrum

Cadence / Delivery

Regular Time box in Sprints

Release Frequency

At the end of each time box or later

Roles

Scrum Master, Product Owner,  
Development Team

key Metric

Velocity

Scope

Scope planned at Sprint Planning,  
in a batch with bundle of works

Change Mechanism

Scope planned at Sprint Planning,  
No Changes allowed mid sprint

Applicability

More appropriate in situations  
where work can be prioritized in  
batches that can be left alone

### Kanban

Continuous Flow

Continuous delivery

No defined Roles except the  
development team, Some team  
consult with Agile Coach

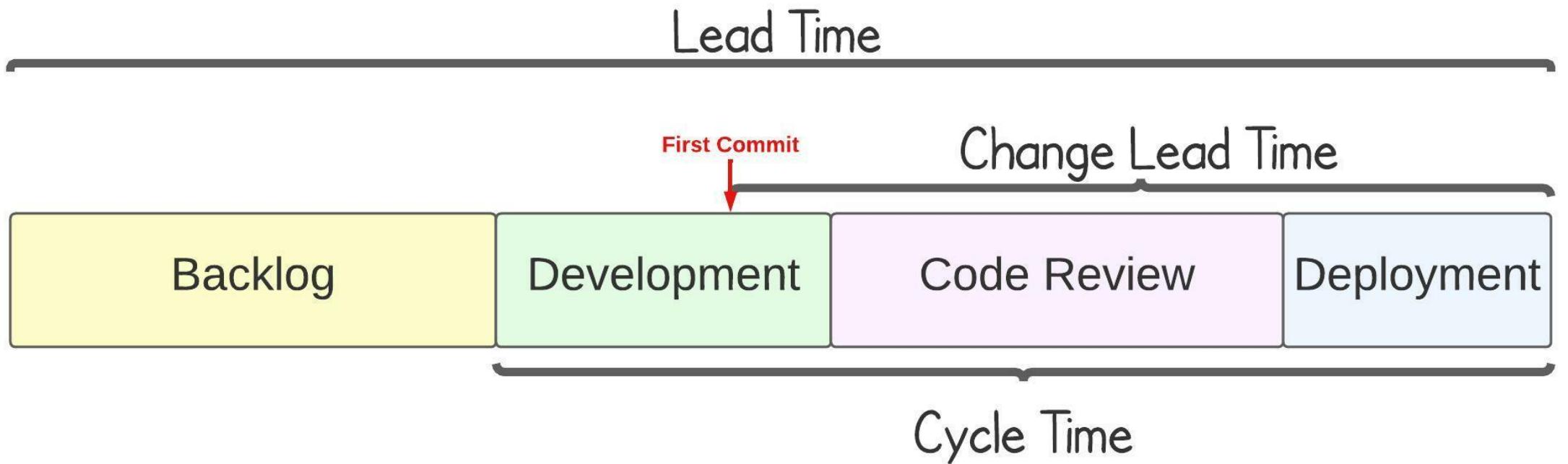
Cycle Tyme

Work pull into the system , one by  
one

Changes can be made any time.

More appropriate in operational  
environments with a high degree  
of variability in priority





## Cumulative Flow Diagram

