

# **The Agile Way to Build Your Product**

Yu Xudong  
2018.09.09



# Yu Xudong

Agile Coach

HangZhou Agile Community Organizer

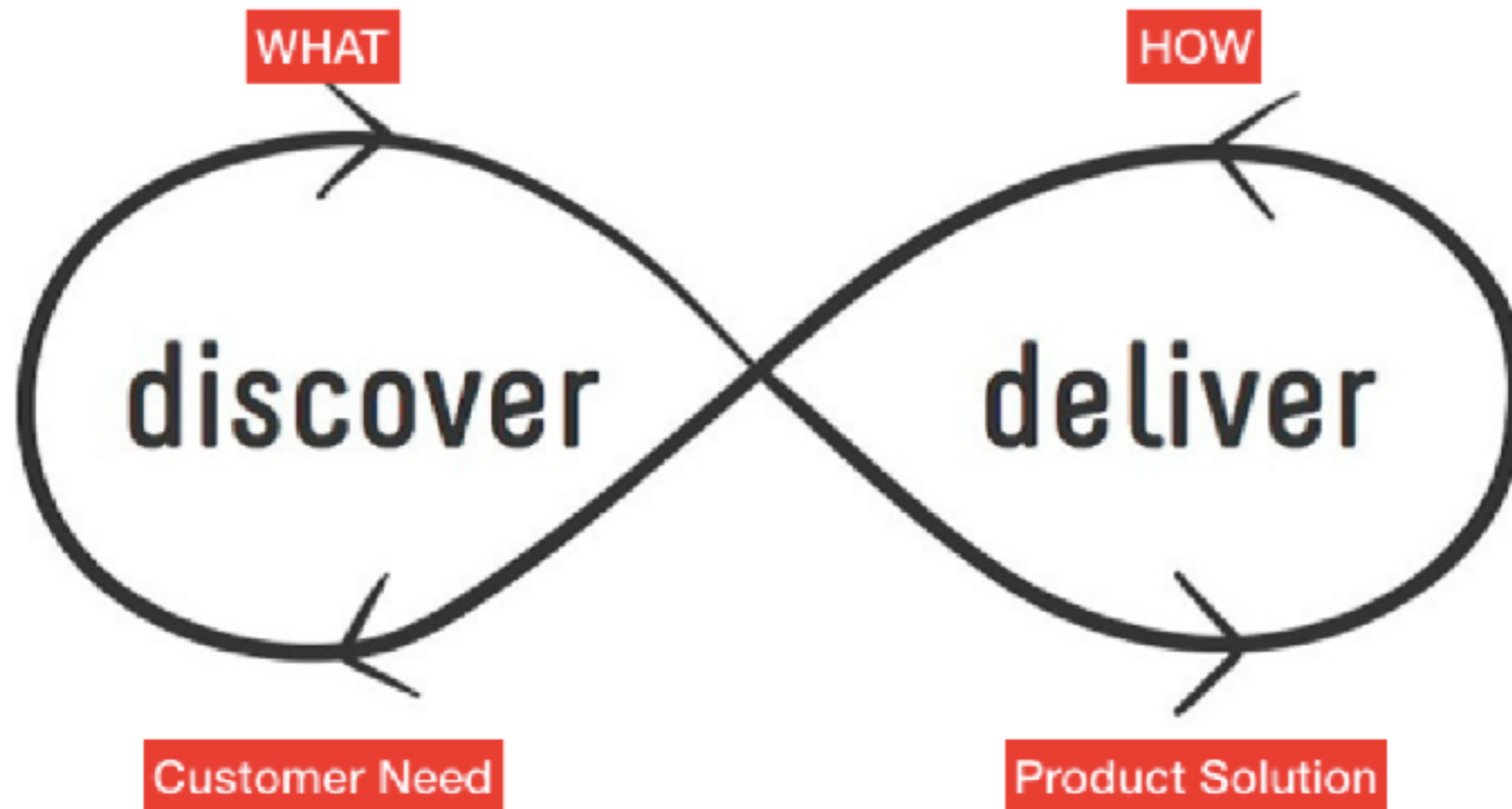
網易 NETEASE  
www.163.com

CAI  
NIAO 菜鸟

NOKIA



# Product Development



# Agenda

- 01 Product Management
- 02 Organization
- 03 Process Management
- 04 Engineering Practice
- 05 Summary

# Background

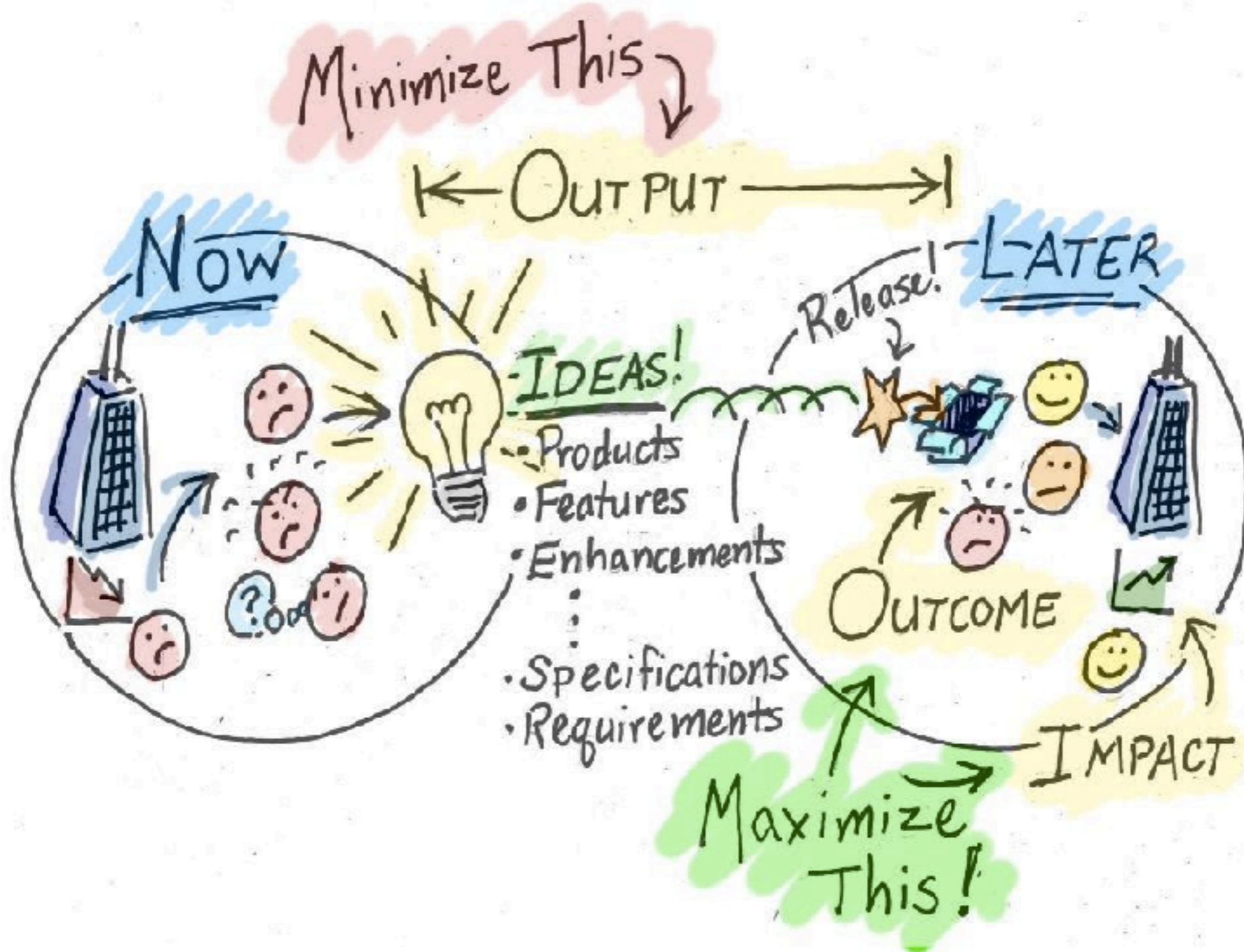
- Internal product
- 10 developers
- Backend and frontend develop separately
- Don't know Agile, Scrum and LeSS
- 1st time cooperation



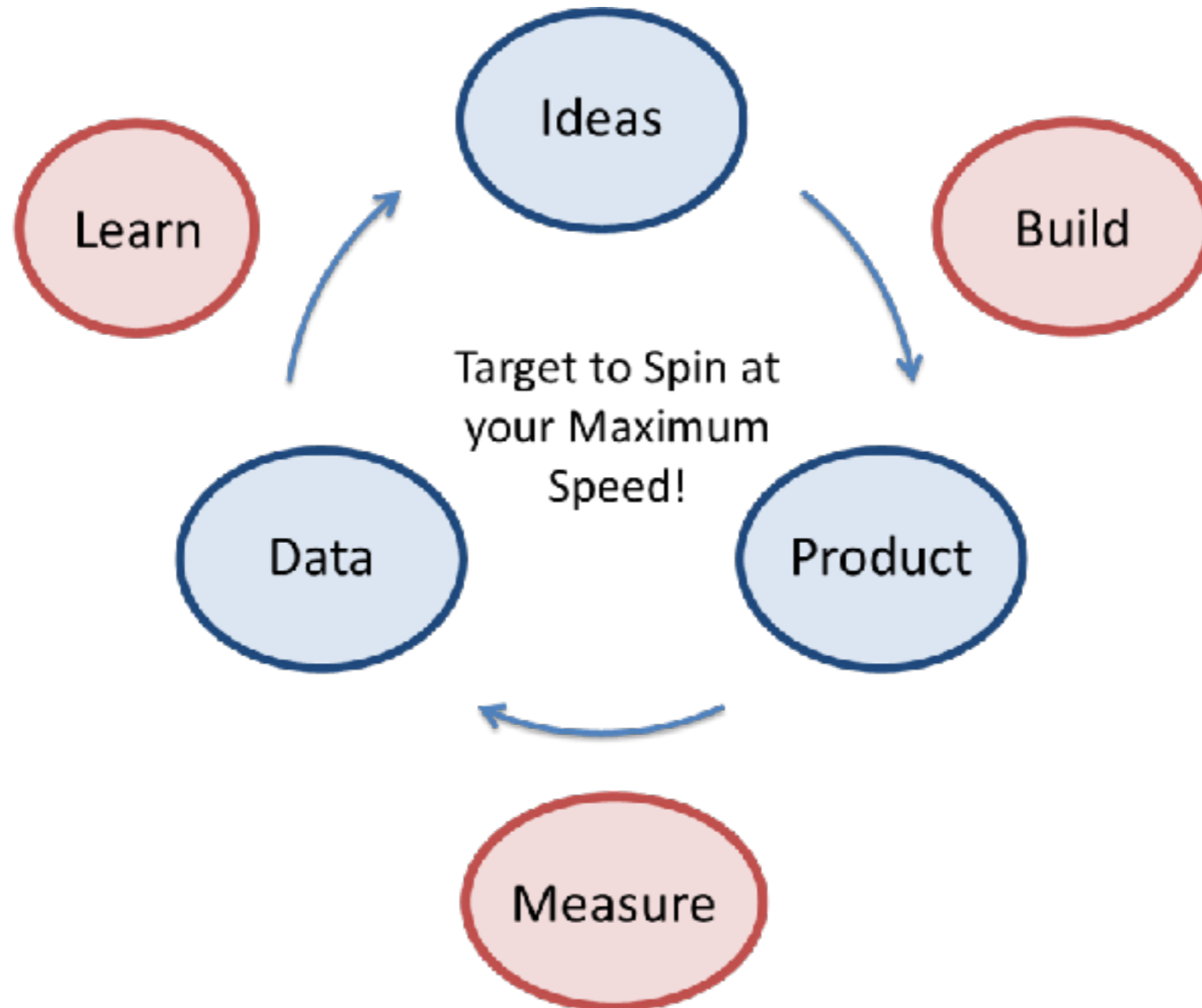
# Product Management

---

# Output vs Outcome



# Lean Startup





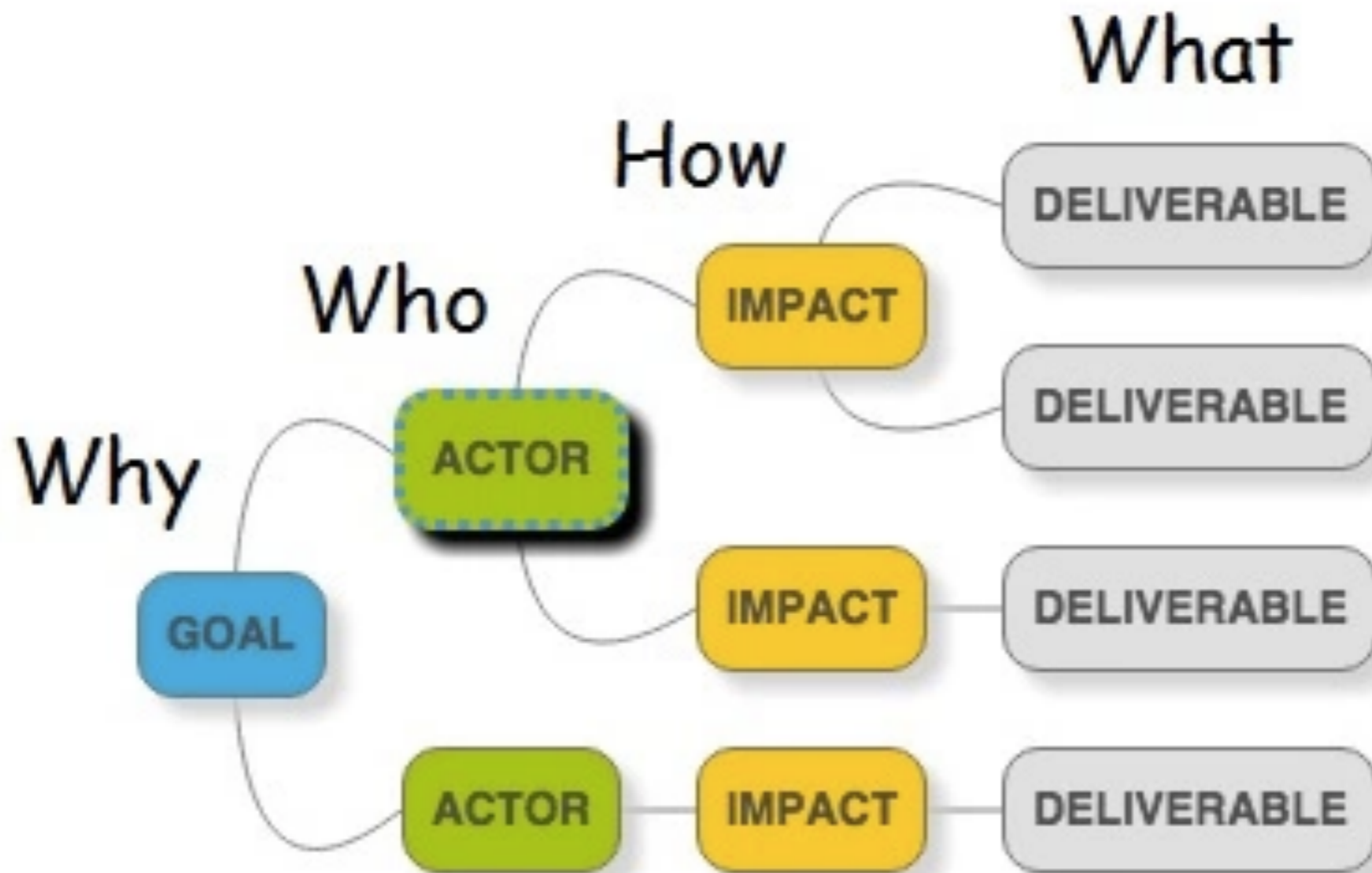


# Product OKR



# Product Data

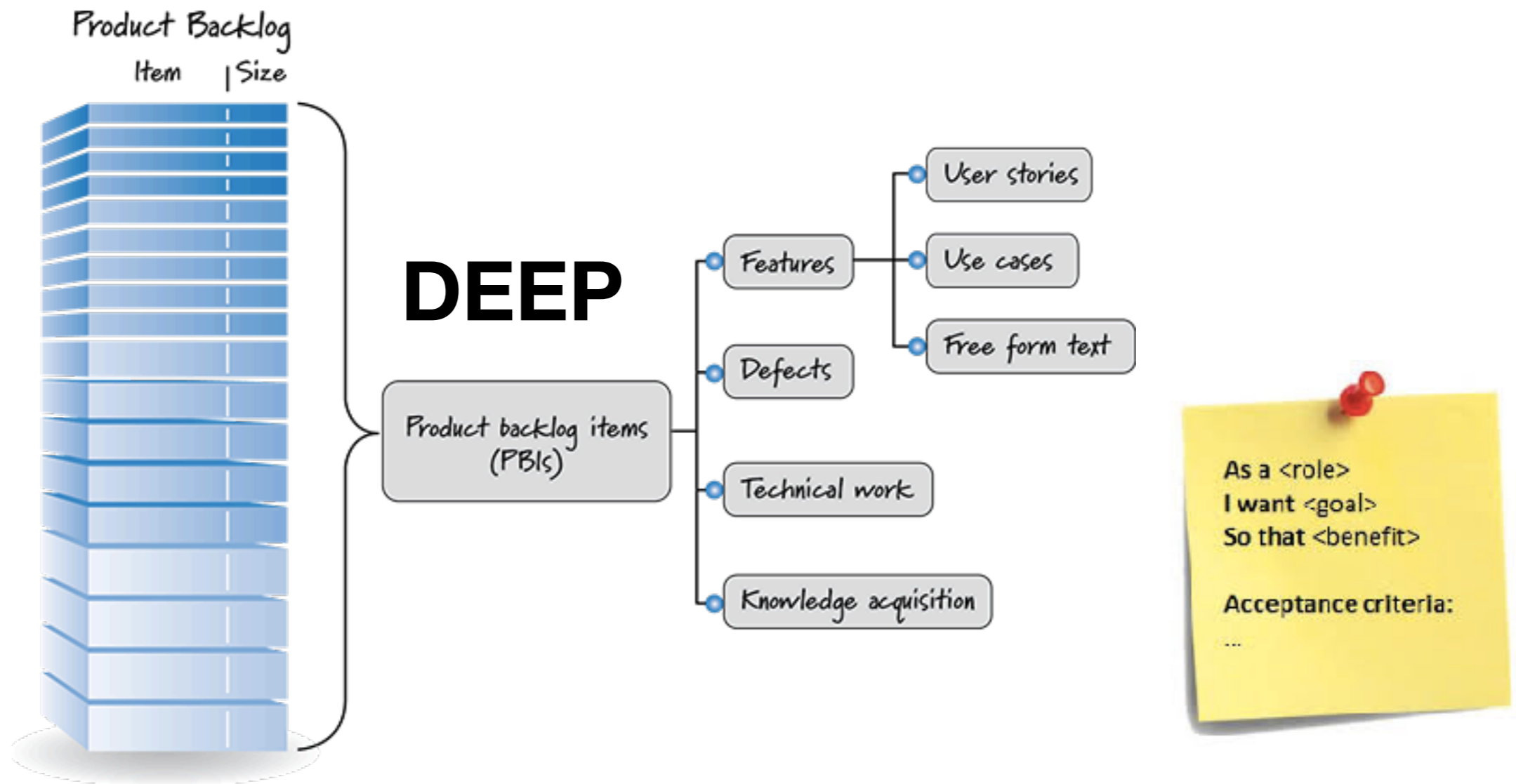
# Impact Mapping



# .....**Impact Mapping**.....

..... **Focus on MVP** .....

# Product Backlog



# Product Backlog

## Example

标识	主题	问题类型
OVERMIND-1358	作为研发人员,我希望应用发布过程中可以从应用异常趋势查看异常详细信息,这样可以及时发现潜在发布问题	故事
OVERMIND-1378	作为研发人员,我希望应用发布过程中可以查看应用某台部署机器load随时间变化趋势图,这样可以及时发现潜在发布问题	故事
OVERMIND-1379	作为研发人员,我希望应用发布过程中可以查看应用集群维度机器load随时间变化趋势图,这样可以及时发现潜在发布问题	故事
OVERMIND-1380	作为研发人员,我希望应用发布过程中可以查看应用某台部署机器GC随时间变化趋势图,这样可以及时发现潜在发布问题	故事
OVERMIND-1381	作为研发人员,我希望应用发布过程中可以查看应用集群维度部署机器GC随时间变化趋势图,这样可以及时发现潜在发布问题	故事

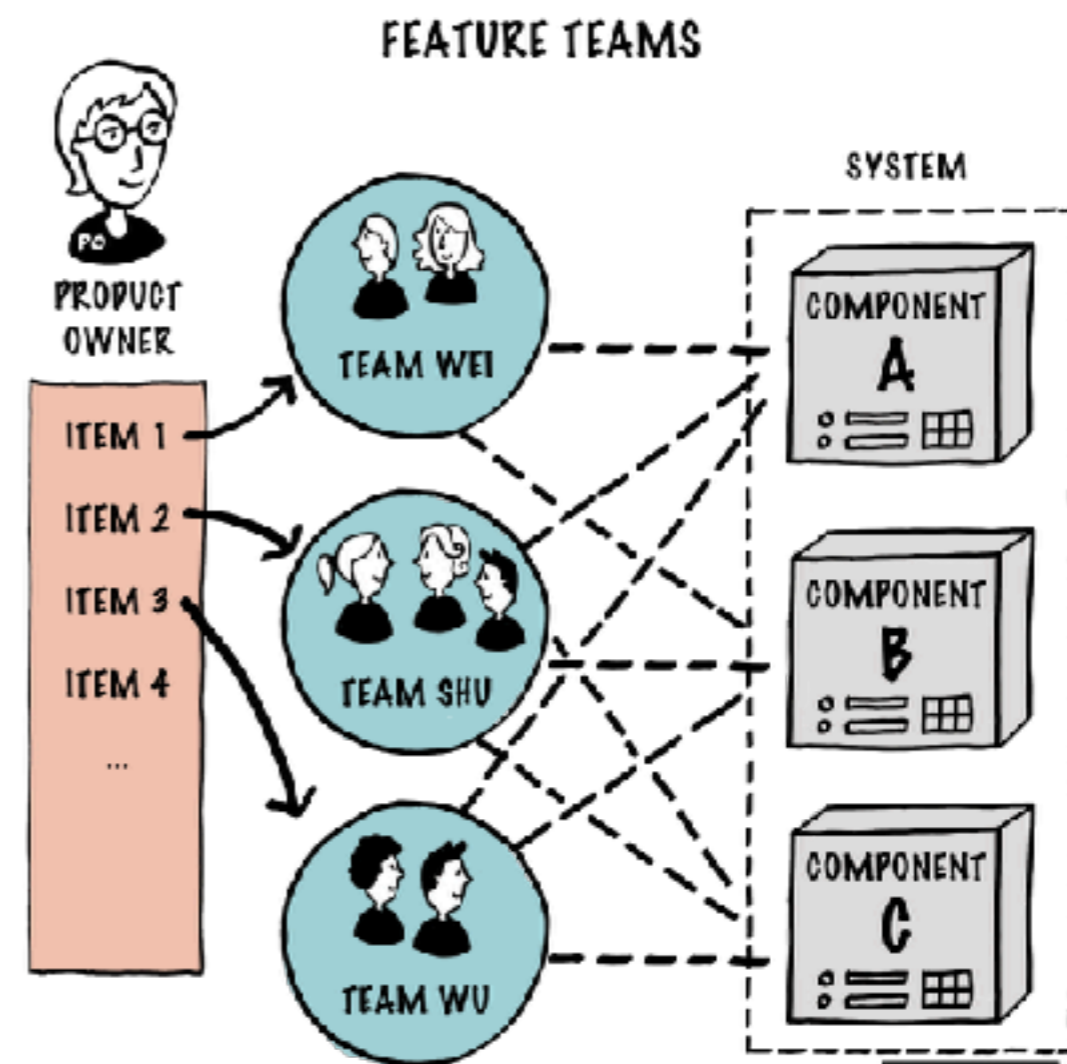
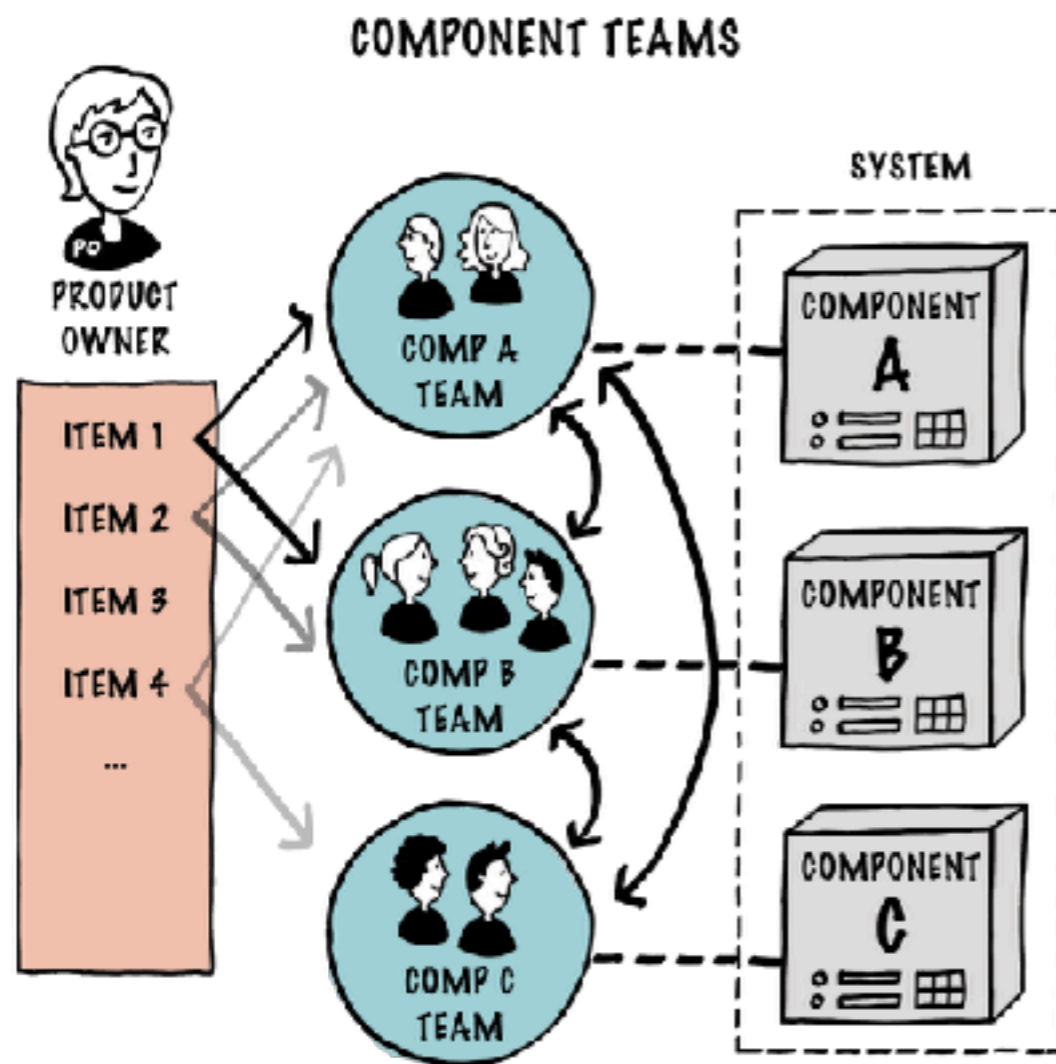


# Organization

---



# ..... Component Team vs Feature Team.....



# ..... Component Team vs Feature Team.....

component team	feature team
focus on maximum own component delivery	focus on maximum customer value delivery
results in 'waterfall' development, later integration	iterative development, continuous integration
tragedy of the commons, local optimize	customer focus, global optimize
temporary project team, low quality and motivation	stable team, high team productivity
multiple project tasks, long delivery cycle	dedicated and focus, delivery quickly
dependencies between teams leads to additional planning	minimizes dependencies between teams to increase flexibility
requirement map skill implementing lower-value features	skill map requirement focus on high-value features
leads to 'invented' work and a forever-growing organization	leads to customer focus, visibility, and smaller organizations
less skills and component	multiple skills and component
individual/team code ownership	shared product code ownership
traditional way - follows Conway's law	modern' way— avoids Conway's law
seemingly easy to implement	seemingly difficult to implement

# Team Setup

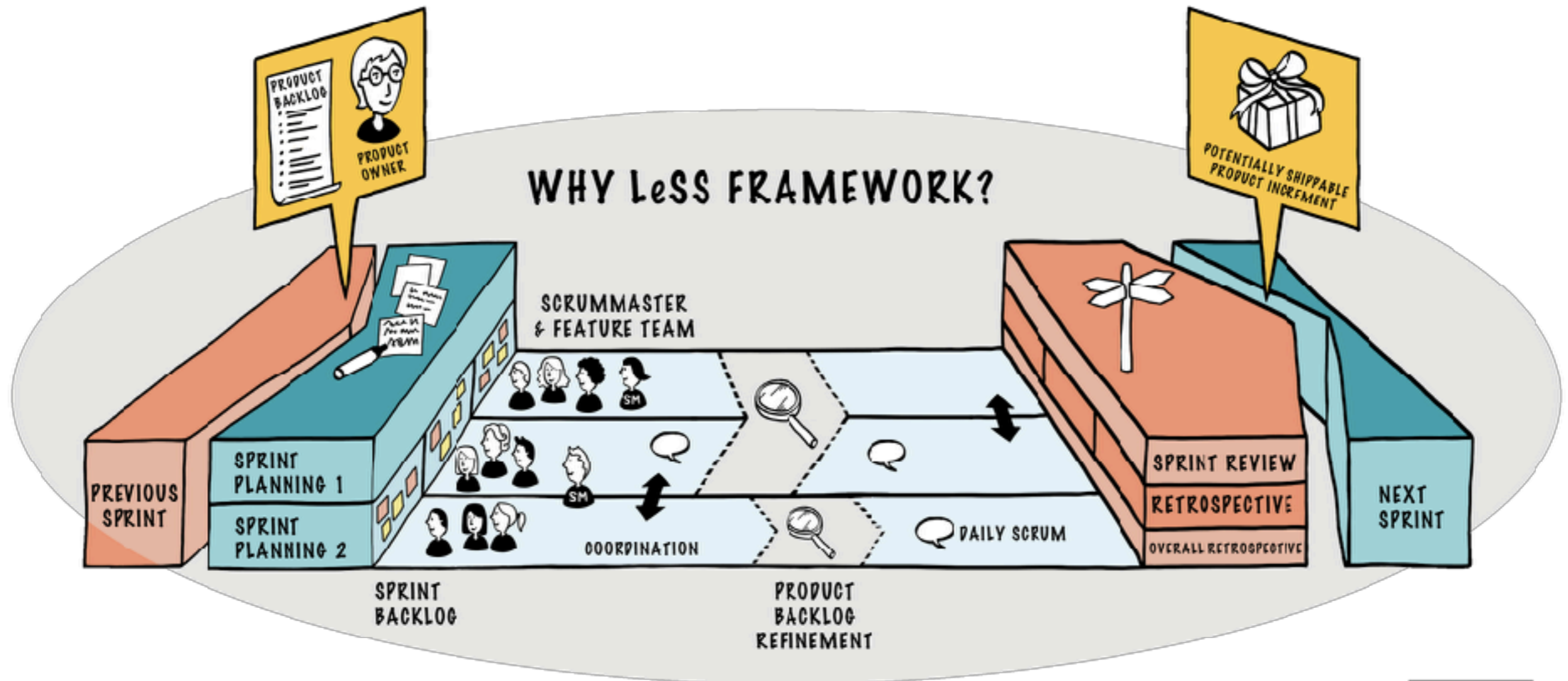




# Process Management

---

# LeSS Framework



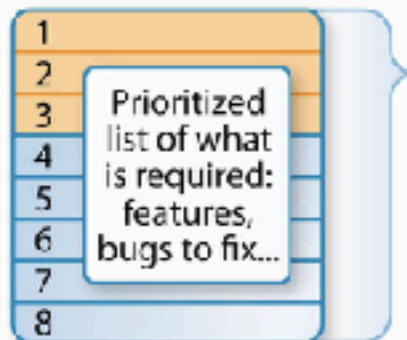
# Scrum Framework

## The Agile Scrum Framework at a glance

Inputs from  
Customers, Team,  
Managers, Execs



Product Owner



Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning

1st Week  
Monday

2nd week  
Monday or Tuesday

Product Backlog Refinement



The Team



Scrum Master



Burn Down/Up Chart

2  
Weeks  
Sprint

24 Hour  
Sprint



Daily Standup Meeting

2nd week  
Friday



Sprint Review



Finished Work



Sprint Retrospective

Task Breakout

Sprint Backlog

Sprint end date and team deliverable do not change



# Agile Practice

## TOP 5 AGILE TECHNIQUES



**90%**  
ITERATION  
PLANNING



**88%**  
DAILY  
STANDUP



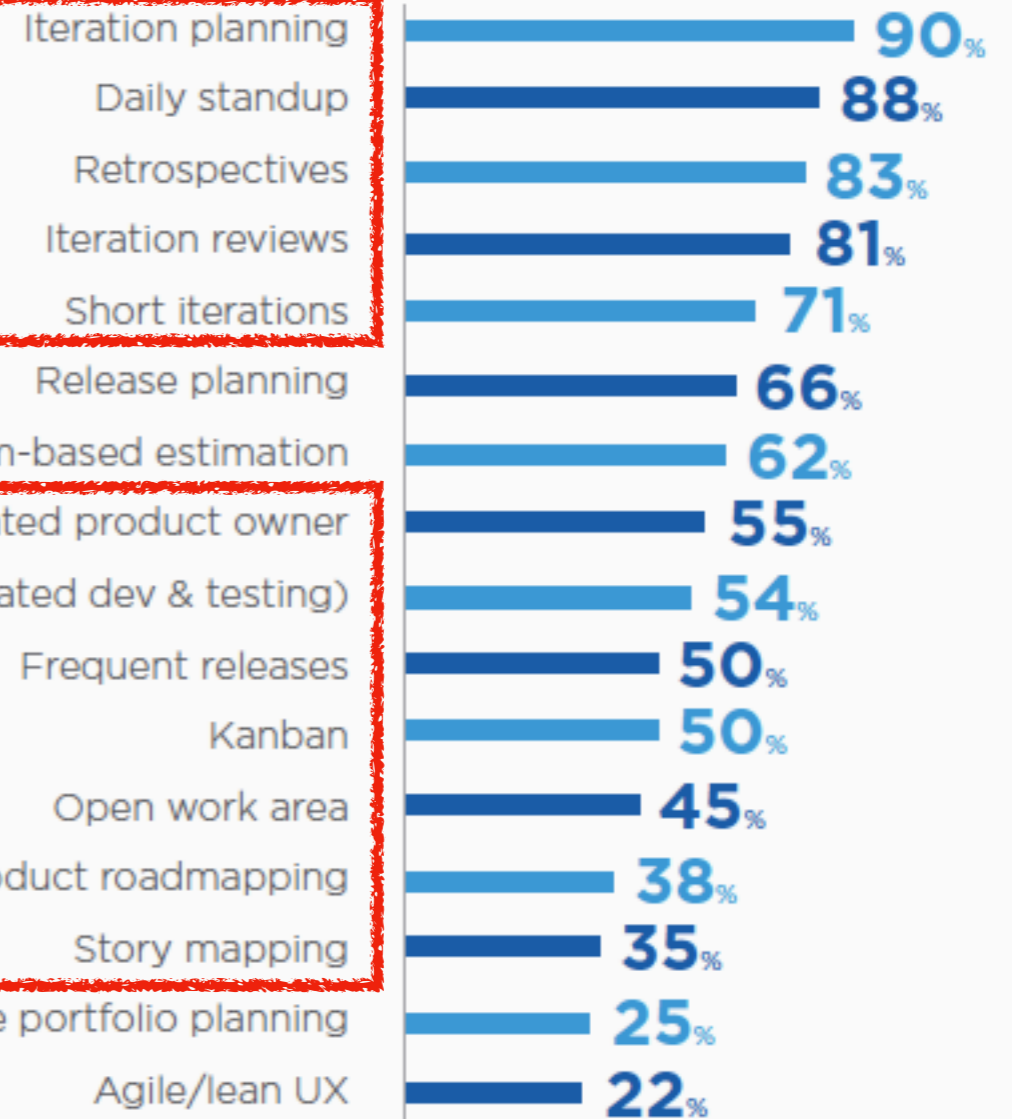
**83%**  
RETROSPECTIVES



**81%**  
ITERATION  
REVIEWS



**71%**  
SHORT  
ITERATIONS



\*Respondents were able to make multiple selections.

# Planning

- **No user story estimation, no task split and estimation, but have a target scope, trust team who can do their best**
  - split user story (independent, negotiable, valuable, small, testable), can change or stop anytime
  - develop user story based on priority
  - start finish, finish start
  - frequent communication between team and PO
  - it's better to understand the technology



# ..... Daily Standup Meeting .....

- **No daily standup meeting**

- less member per team
- working in same meeting room and face to face communication on demand
- teams manage the dependencies among teams
- leading team responsible for integration
- “Can I help you?” PO ask team at least one time every day

# Review & Retro

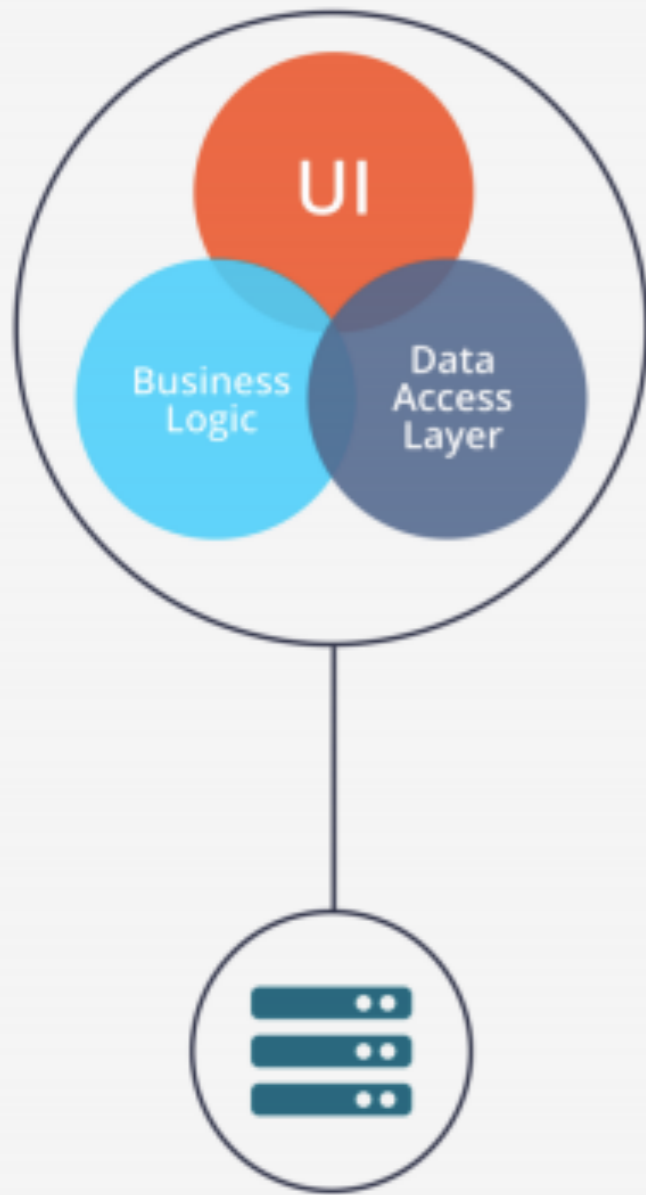
- **Continually review as early as possible**
  - walk through product demo with stakeholder to test the solution before sprint start
  - ask PO&Stakeholder review when one user story finished to avoid surprise during review meeting
  - delivery to production environment to receive user feedback as quickly as possible



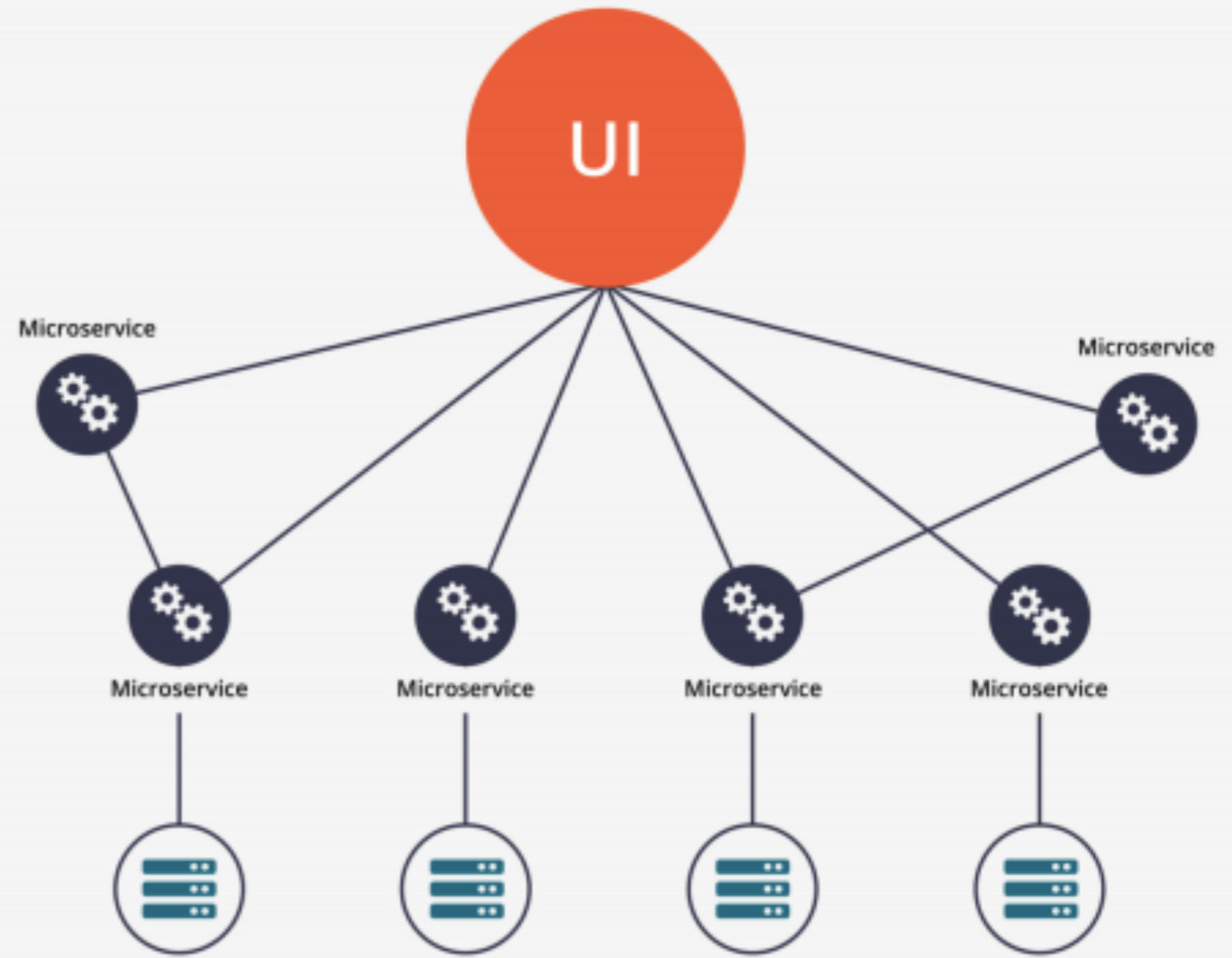
# Engineering Practice

---

# ..... Monolithic to Microservice .....

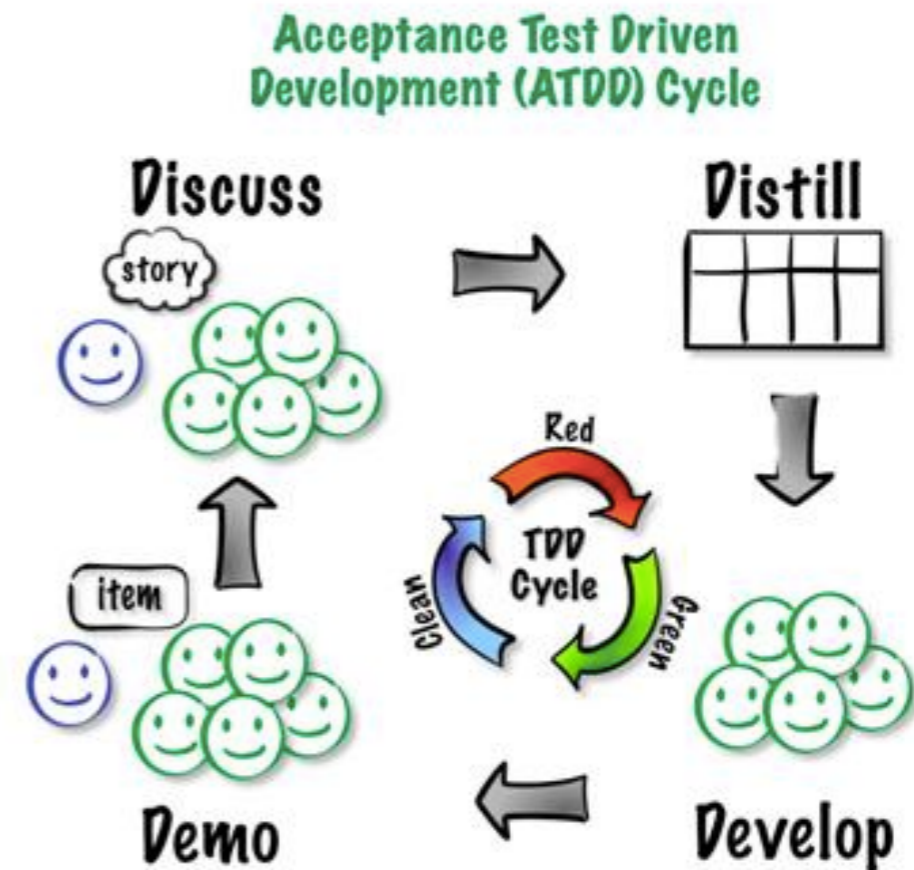
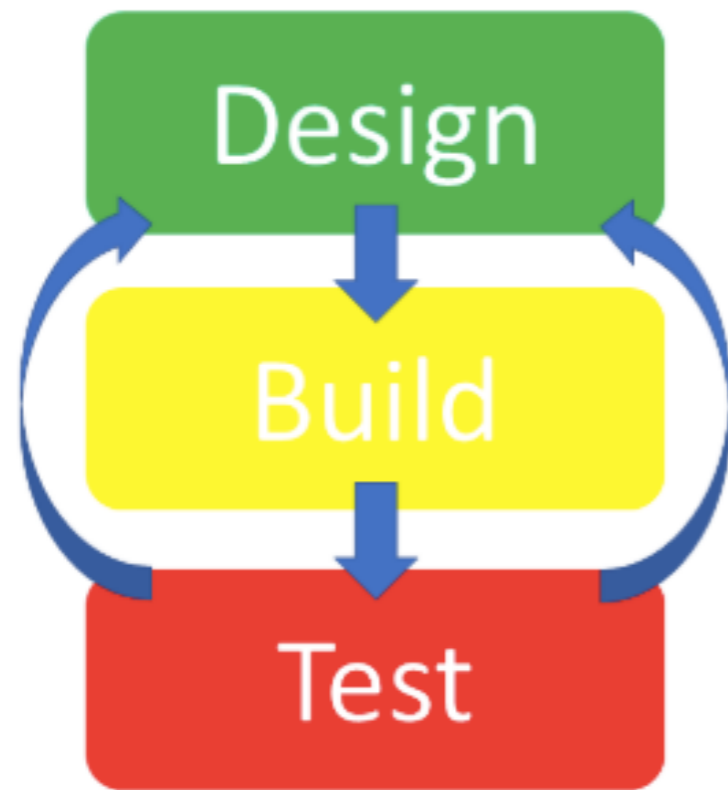


Monolithic Architecture



Microservice Architecture

# ..... Test-later to Test-before .....



# 05 PART

## Summary

---

# ..... Product Result .....

# Agile Benefit

## | Before

1. Deliver product later later and later
2. Customer not satisfied with product including functionality and quality
3. Less process visibility
4. Feedback later and hard to change

## | After

1. Deliver product continually and early
2. Customer satisfied including functionality and quality
3. Good process visibility
4. Receive feedback early and easily respond to change



# Key Point

- one product owner and one product backlog
- feature team
- lean startup and data driven
- focus on MVP
- user story split (INVEST) and clarification
- fixed time box sprint ( e.g. 2 weeks)
- deliver working product every sprint continually and early
- feedback quickly and continually
- good engineering practice

# Disclaimer

There are **no** such things as **best practices** in product development.

---

There are only **practices** that are **adequate** within a certain **context**.

---

# Q&A

## Thank You

