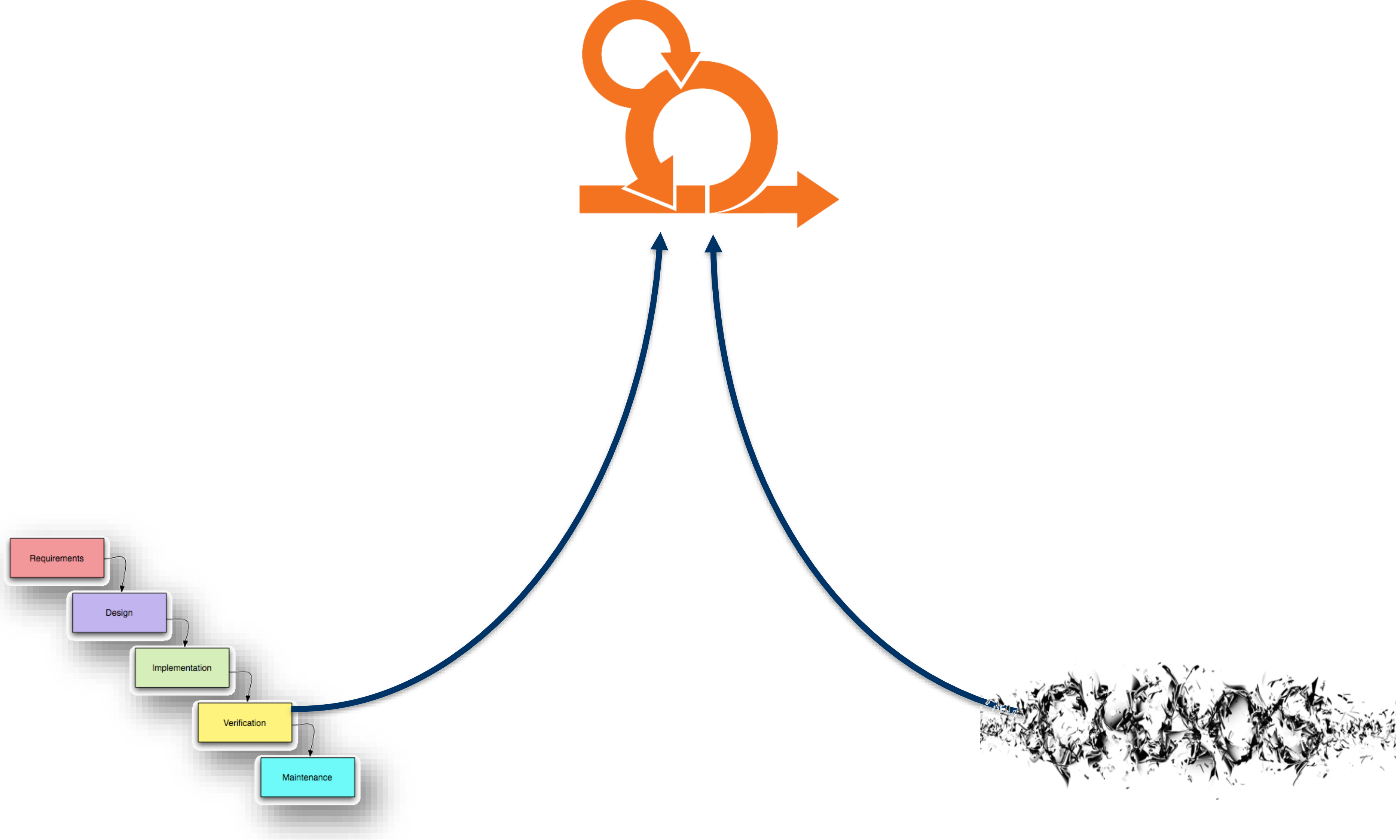
AGILE SOFTWARE DEVELOPMENT

FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ • BABEŞ-BOLYAI • ROMÂNIA

Hybrid Approaches
Scrum Methodology. Roles



Requirements

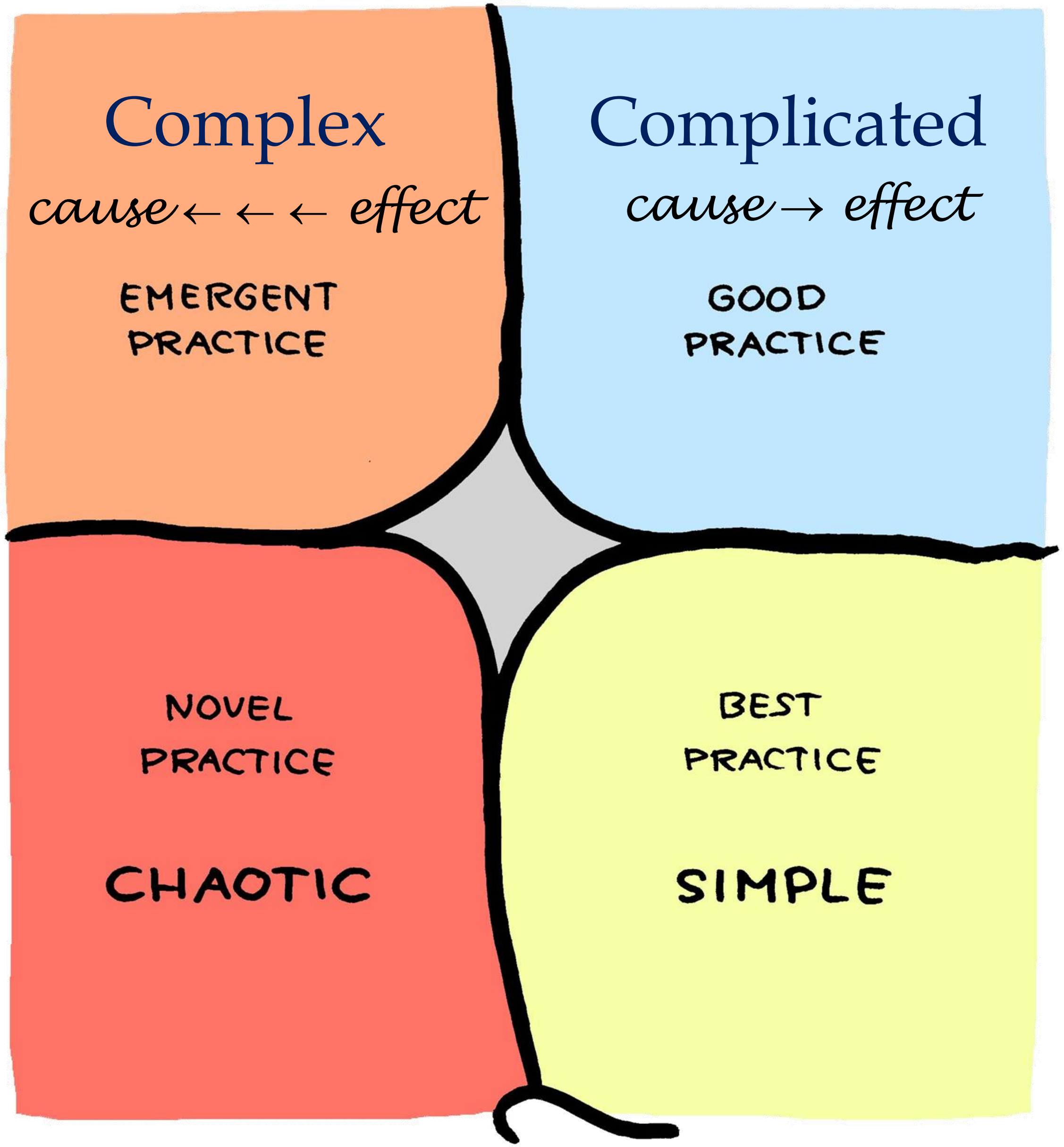
Design

Implementation

Verification

Maintenance

"be more agile!"



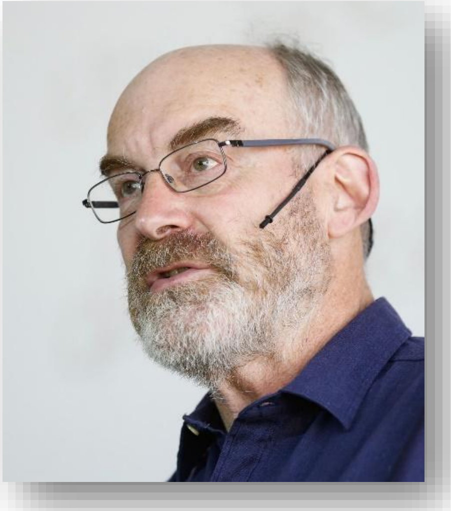
Complex

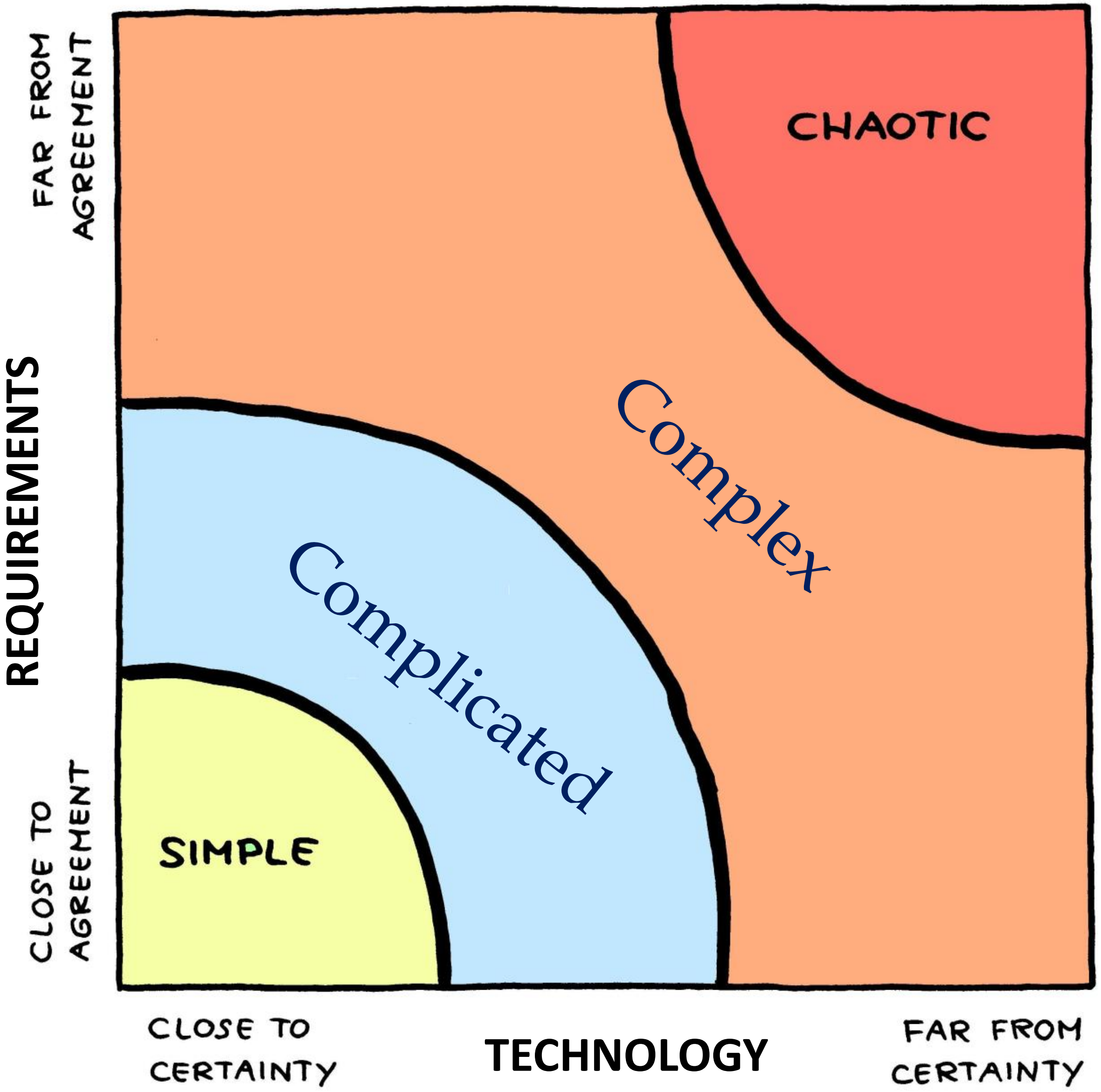
cause ← ← ← effect

Complicated

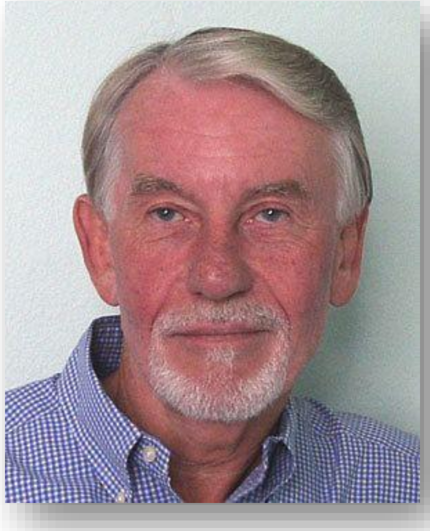
cause → effect

Dave Snowden





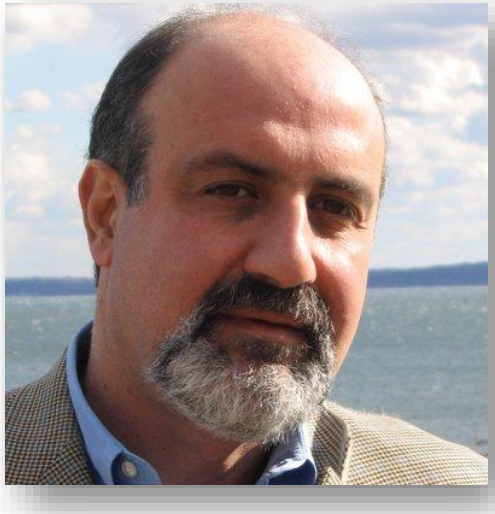
Ralph D. Stacey



Complicated

Complex

Nassim Taleb



Complicated

CAR

HOUSE

BOOK

COMPUTER

Complex

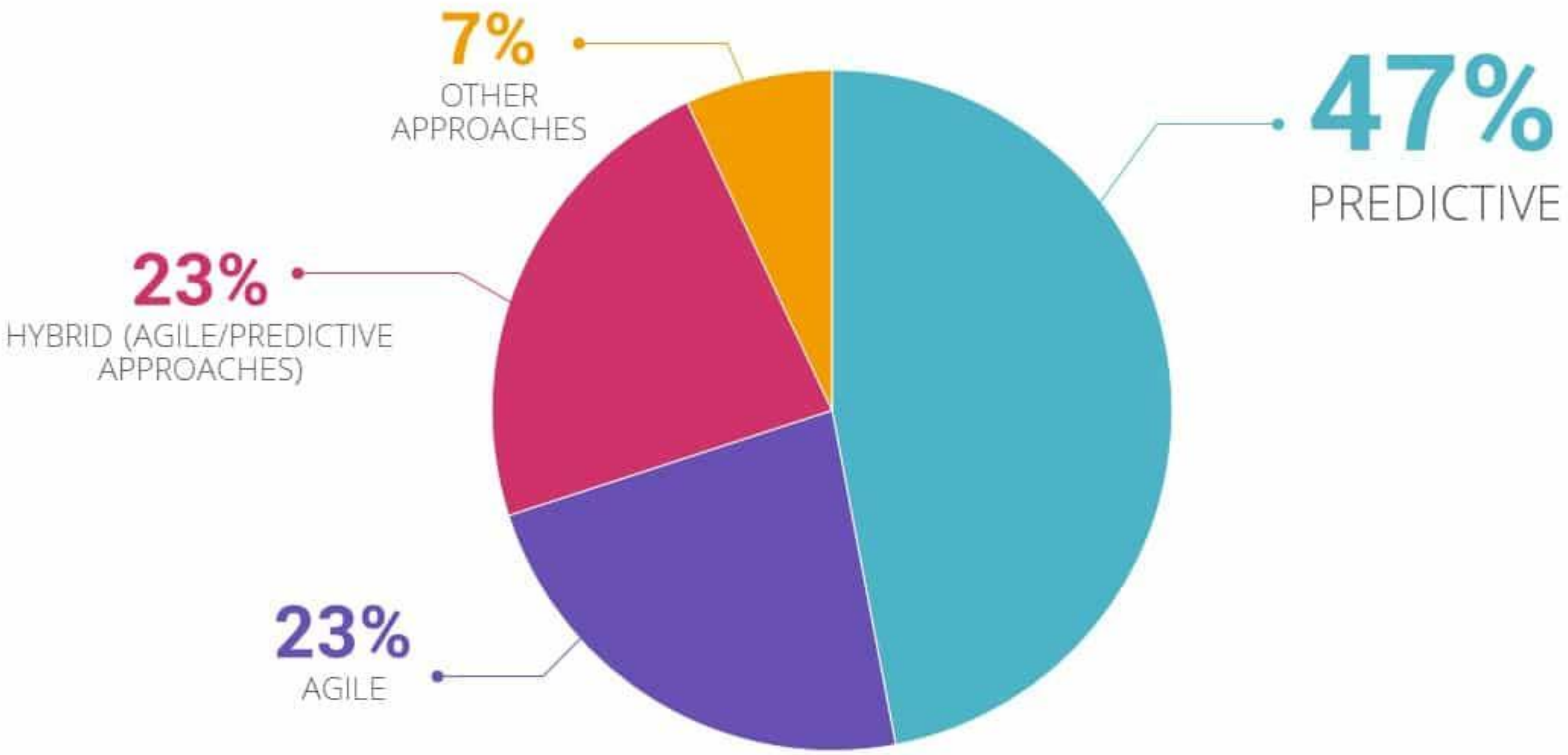
TRAFFIC

HOUSEHOLD

THOUGHTS

SOFTWARE

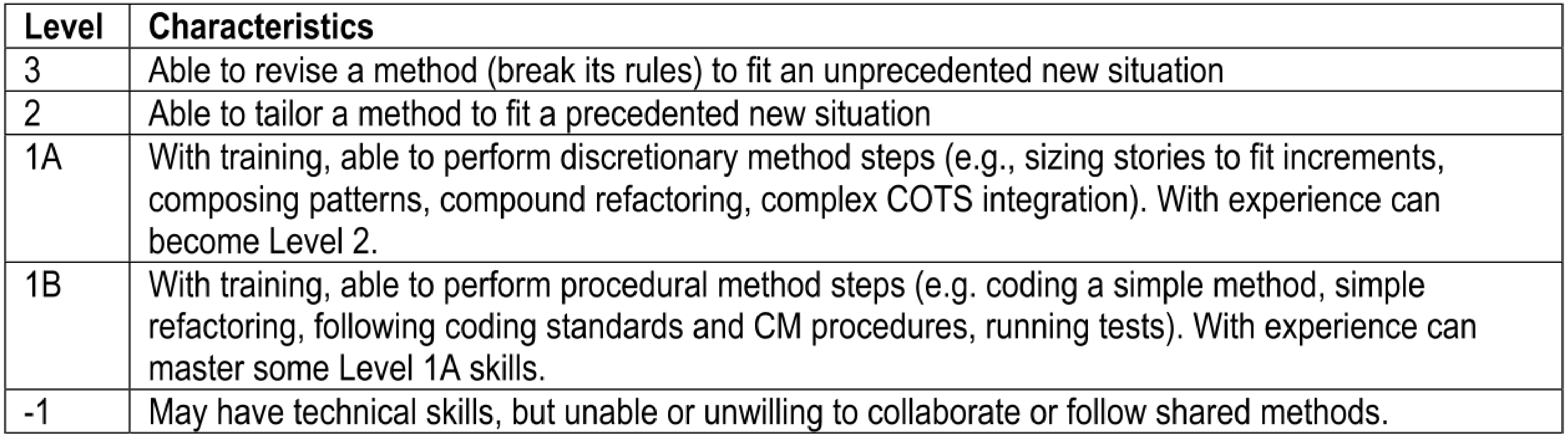
DEVELOPMENT APPROACH USED



Source: PMI 2018 Pulse of The Profession

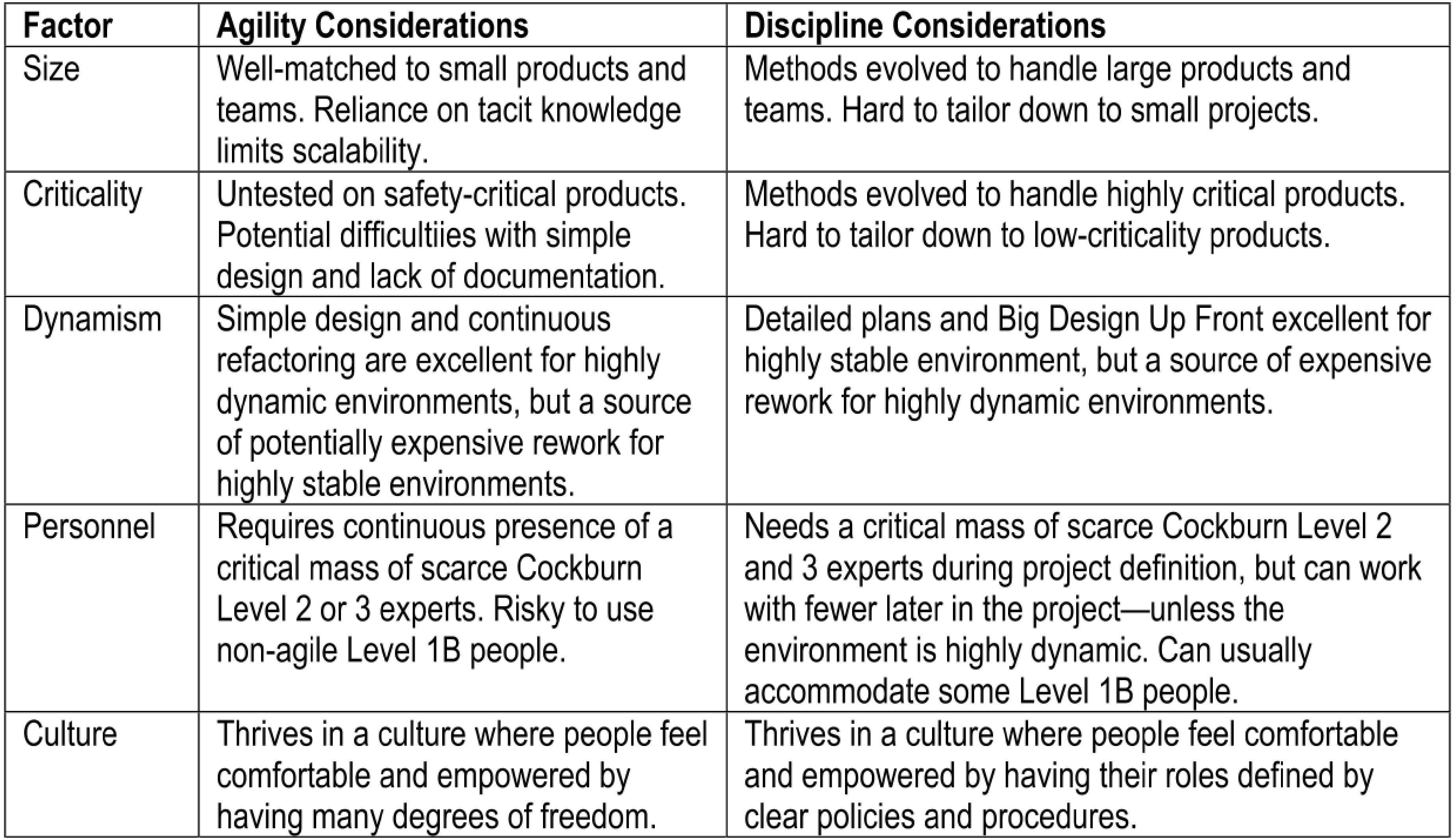
WWW.VITALITYCHICAGO.COM

Levels of Software Method Understanding and Use



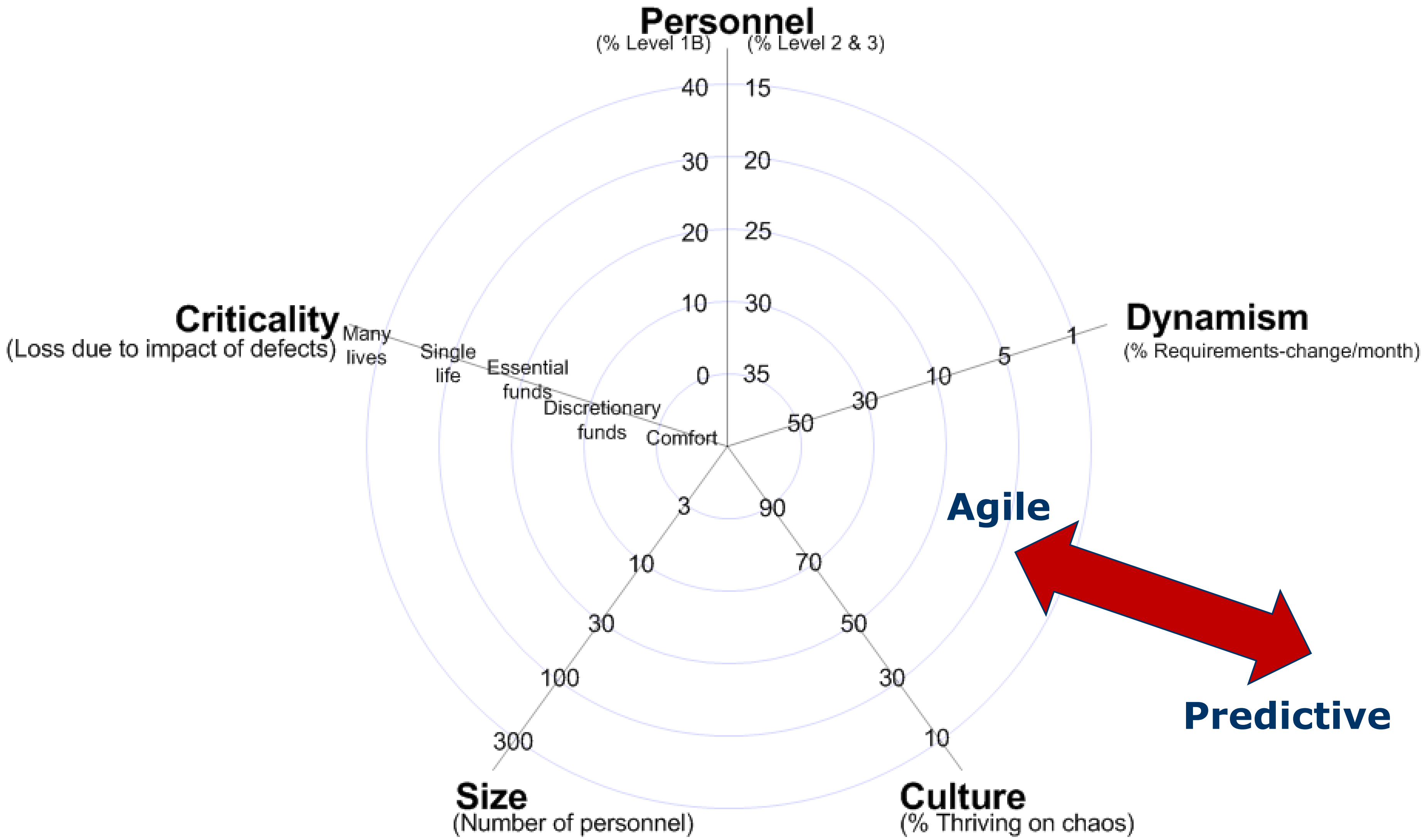
Level	Characteristics
3	Able to revise a method (break its rules) to fit an unprecedented new situation
2	Able to tailor a method to fit a precedented new situation
1A	With training, able to perform discretionary method steps (e.g., sizing stories to fit increments, composing patterns, compound refactoring, complex COTS integration). With experience can become Level 2.
1B	With training, able to perform procedural method steps (e.g. coding a simple method, simple refactoring, following coding standards and CM procedures, running tests). With experience can master some Level 1A skills.
-1	May have technical skills, but unable or unwilling to collaborate or follow shared methods.

From "Agile Software Development", A. Cockburn (Addison-Wesley, 2002)

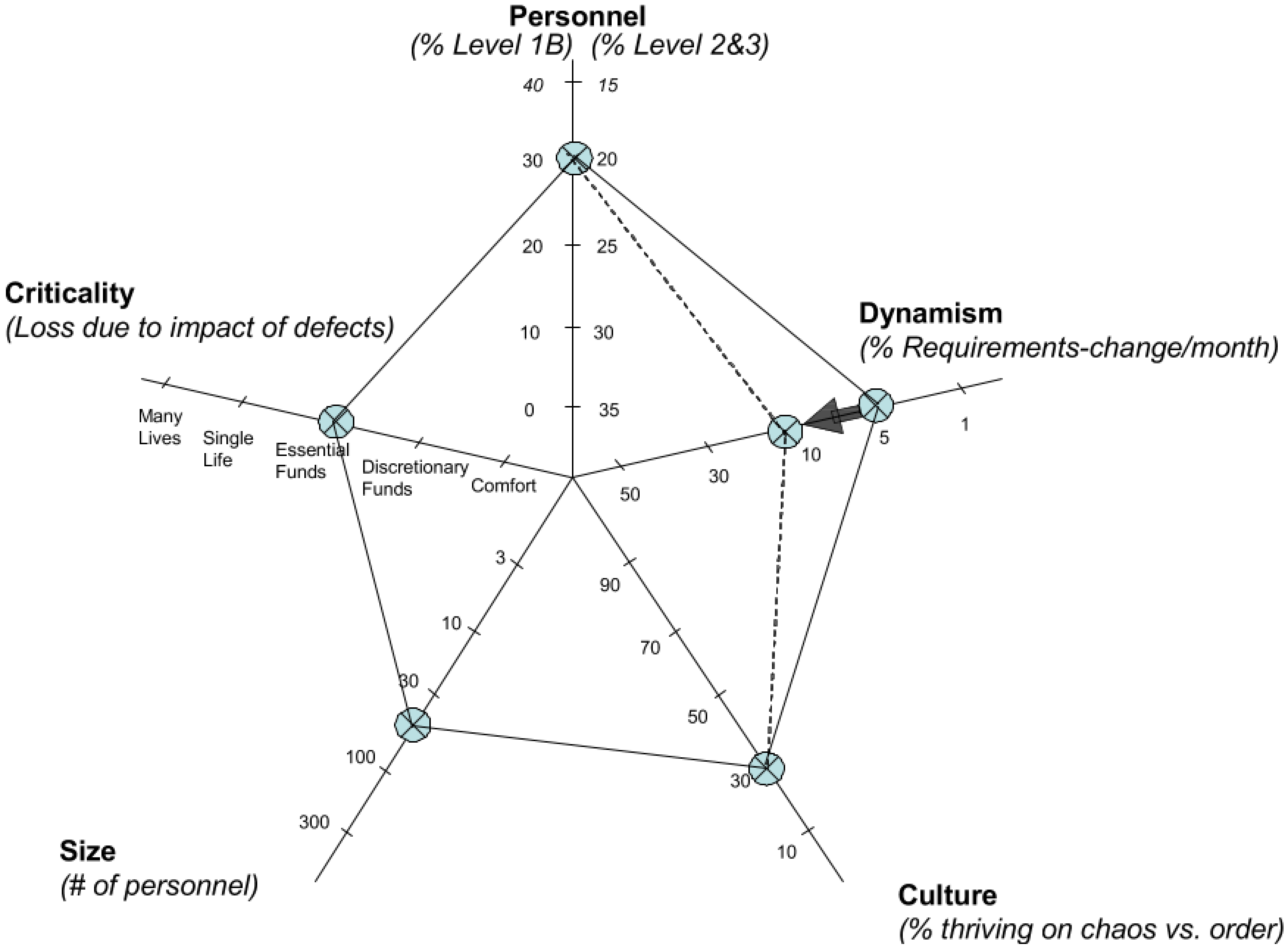


Factor	Agility Considerations	Discipline Considerations
Size	Well-matched to small products and teams. Reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams. Hard to tailor down to small projects.
Criticality	Untested on safety-critical products. Potential difficultiies with simple design and lack of documentation.	Methods evolved to handle highly critical products. Hard to tailor down to low-criticality products.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments, but a source of potentially expensive rework for highly stable environments.	Detailed plans and Big Design Up Front excellent for highly stable environment, but a source of expensive rework for highly dynamic environments.
Personnel	Requires continuous presence of a critical mass of scarce Cockburn Level 2 or 3 experts. Risky to use non-agile Level 1B people.	Needs a critical mass of scarce Cockburn Level 2 and 3 experts during project definition, but can work with fewer later in the project—unless the environment is highly dynamic. Can usually accommodate some Level 1B people.
Culture	Thrives in a culture where people feel comfortable and empowered by having many degrees of freedom.	Thrives in a culture where people feel comfortable and empowered by having their roles defined by clear policies and procedures.

from "Balancing Agility and Discipline", Barry Boehm & Richard Turner



from "Balancing Agility and Discipline", Barry Boehm & Richard Turner

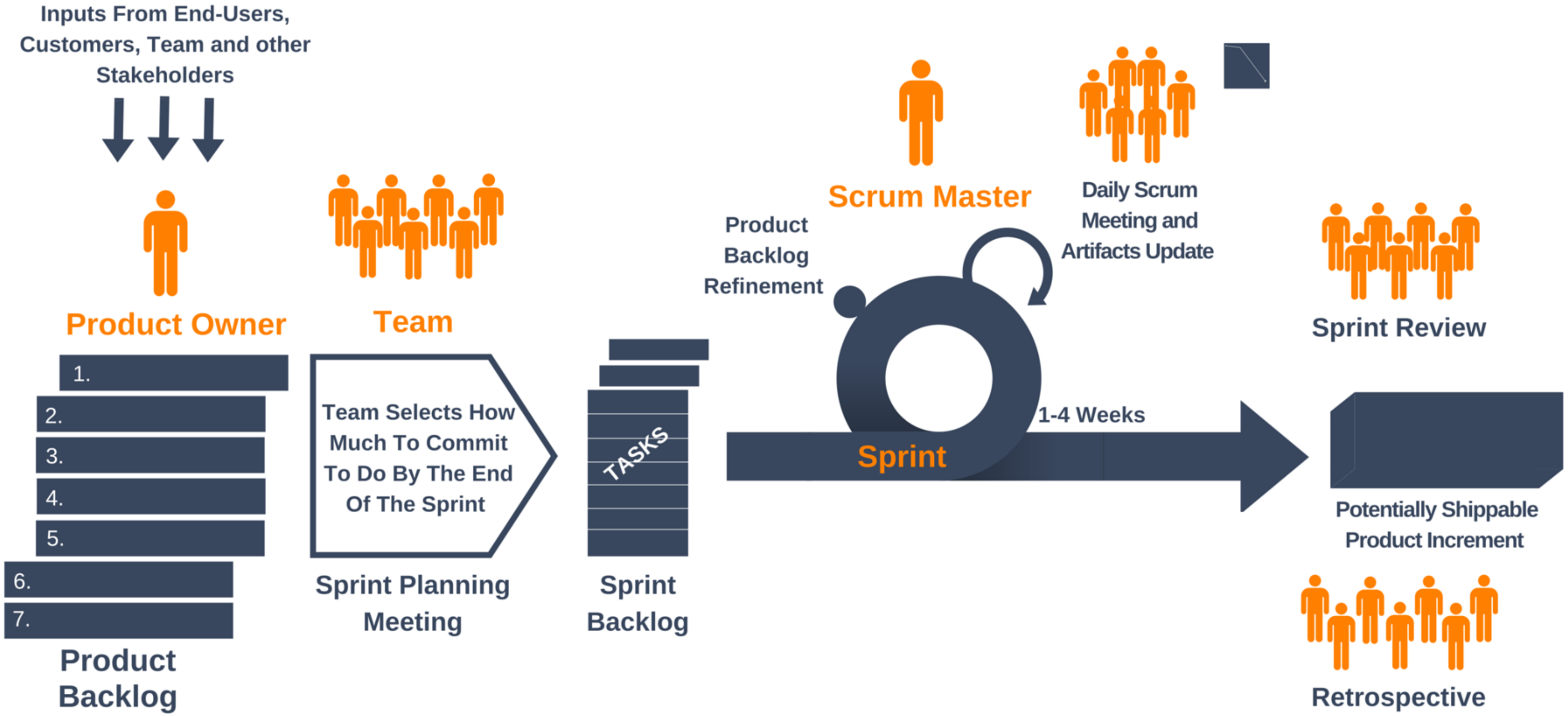

Personnel
(% Level 1B) (% Level 2&3)
40
15
30
20
20
25
10
30
0
35
Criticality
(Loss due to impact of defects)
Many Lives
Single Life
Essential Funds
Discretionary Funds
Comfort
Dynamism
(% Requirements-change/month)
1
5
10
30
50
Size
(# of personnel)
3
10
30
100
300
Culture
(% thriving on chaos vs. order)
90
70
50
30
10

Christopher Avery

"We can never direct a living system, only disturb it and wait to see the response…

We can't know all the forces shaping an organization we wish to change, so all we can do is provoke the system in some way be experimenting with a force we think might have some impact, then watch to see what happens."

Scrum Methodology



Inputs From End-Users, Customers, Team and other Stakeholders

Product Owner

Team

1.

2.

3.

4.

5.

6.

7.

Product Backlog

Team Selects How Much To Commit To Do By The End Of The Sprint

Sprint Planning Meeting

TASKS

Sprint Backlog

Product Backlog Refinement

Scrum Master

Daily Scrum Meeting and Artifacts Update

Sprint

1-4 Weeks

Sprint Review

Potentially Shippable Product Increment

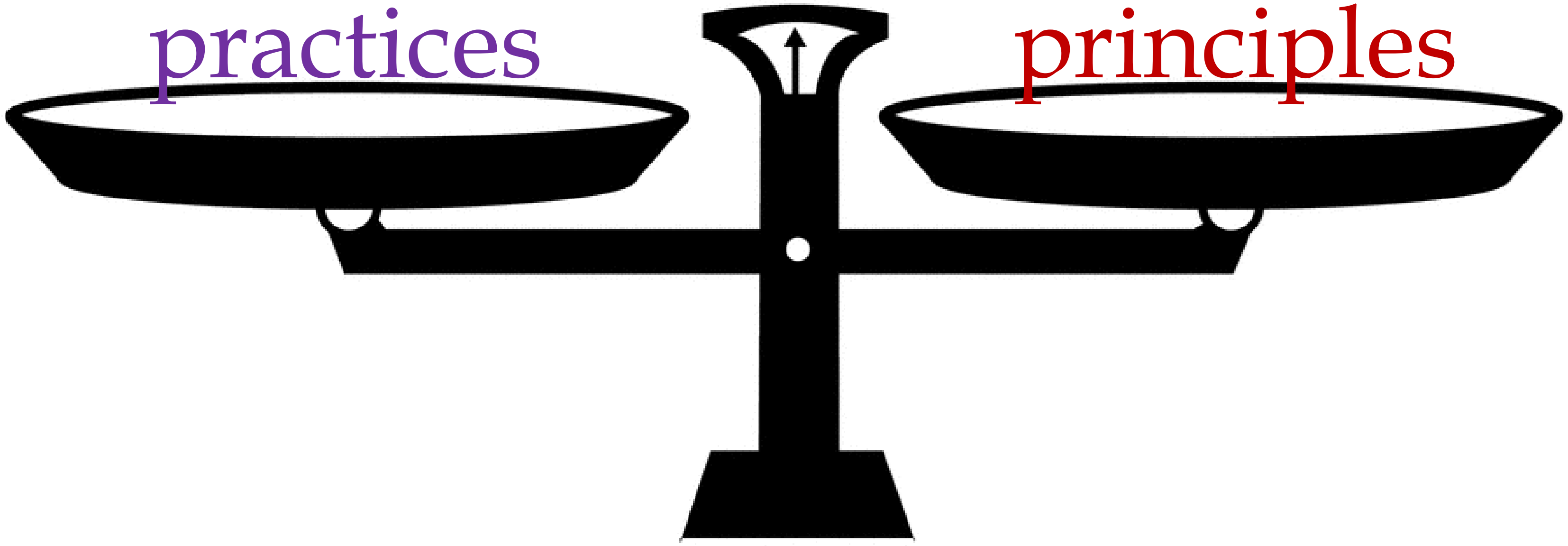
Retrospective

SCRUM

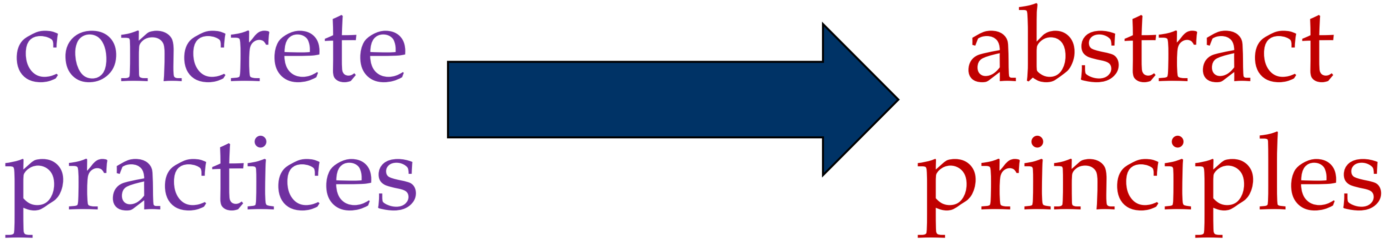
…. hits an ideal balance between

concrete practices

abstract principles



SCRUM



concrete practices

abstract principles

SCRUM

learn fast?

is a **lightweight** framework designed to help

small, close-knit teams of people

7 ± 2

develop **complex** products.

remember
Cynefin framework

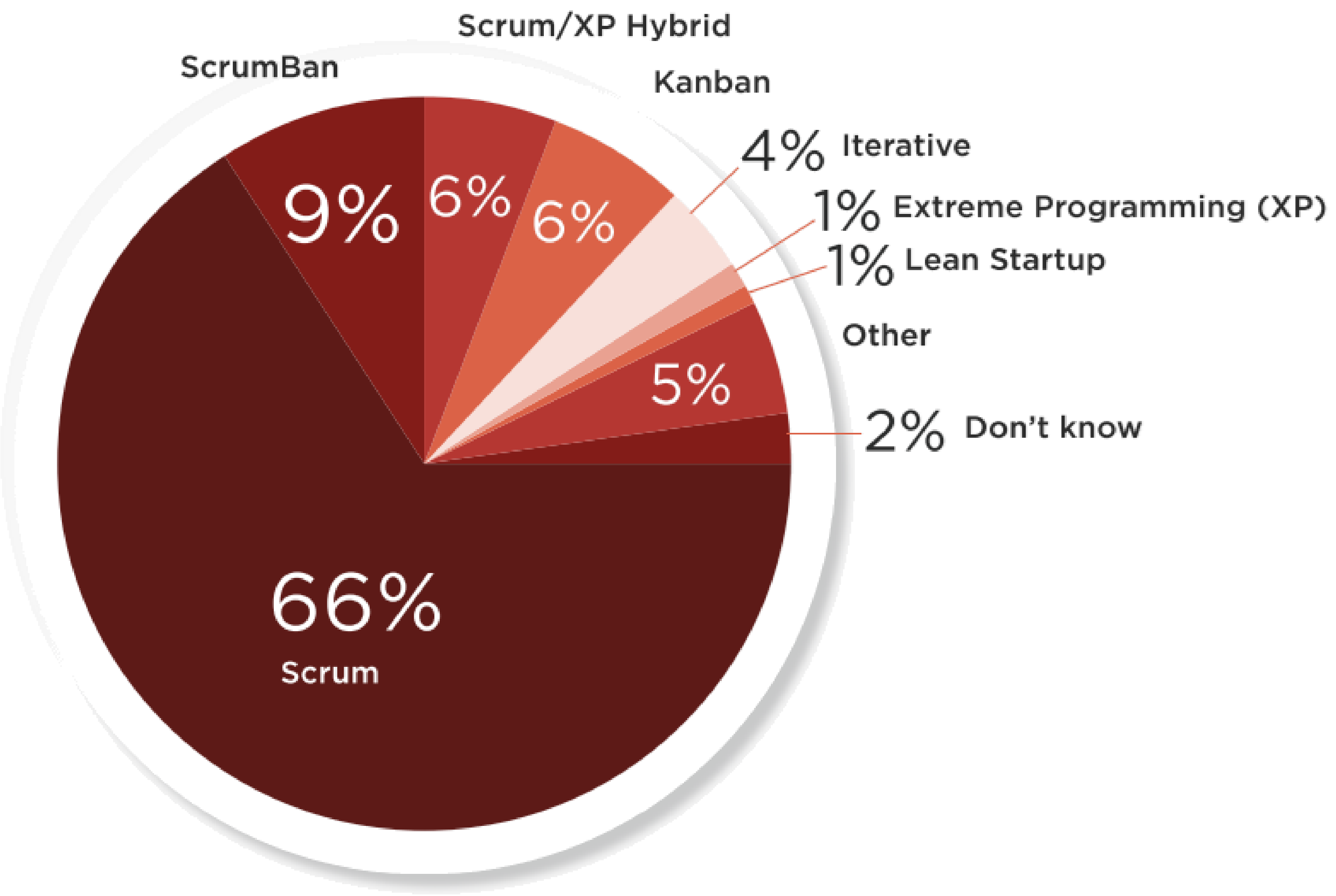
Scrum Pillars

The pillars of Scrum

- Transparency
- Inspection
- Adaptation

They refer to

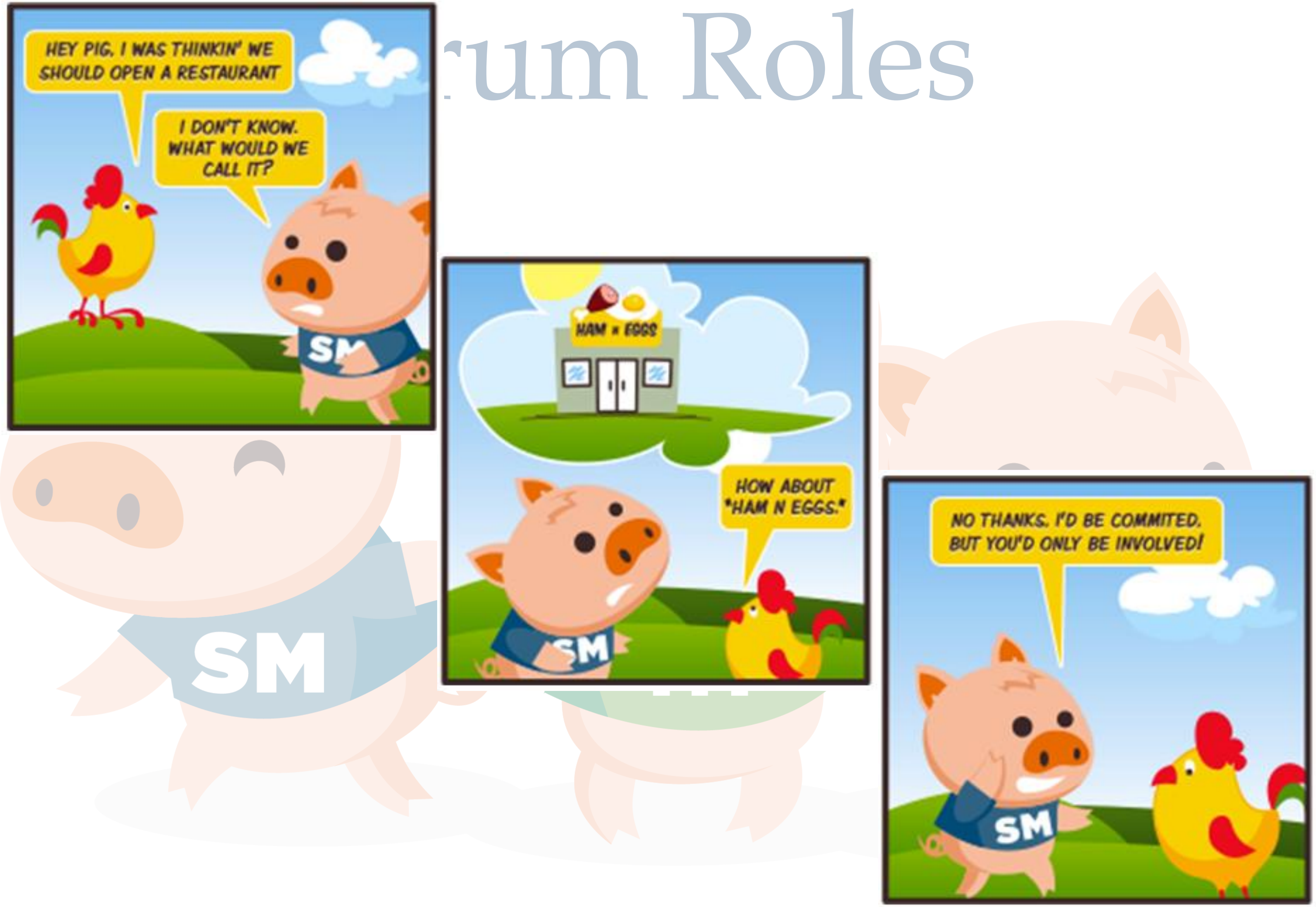
- Process
- Results



15th Annual State of Agile Report, 2021
https://digital.ai/resource-center/analyst-reports/state-of-agile-report

Scrum Roles

rum Roles


HEY PIG, I WAS THINKIN' WE SHOULD OPEN A RESTAURANT
I DON'T KNOW. WHAT WOULD WE CALL IT?
SM
HAM n EGGS
HOW ABOUT "HAM N EGGS."
SM
NO THANKS. I'D BE COMMITED, BUT YOU'D ONLY BE INVOLVED!
SM

Scrum Roles

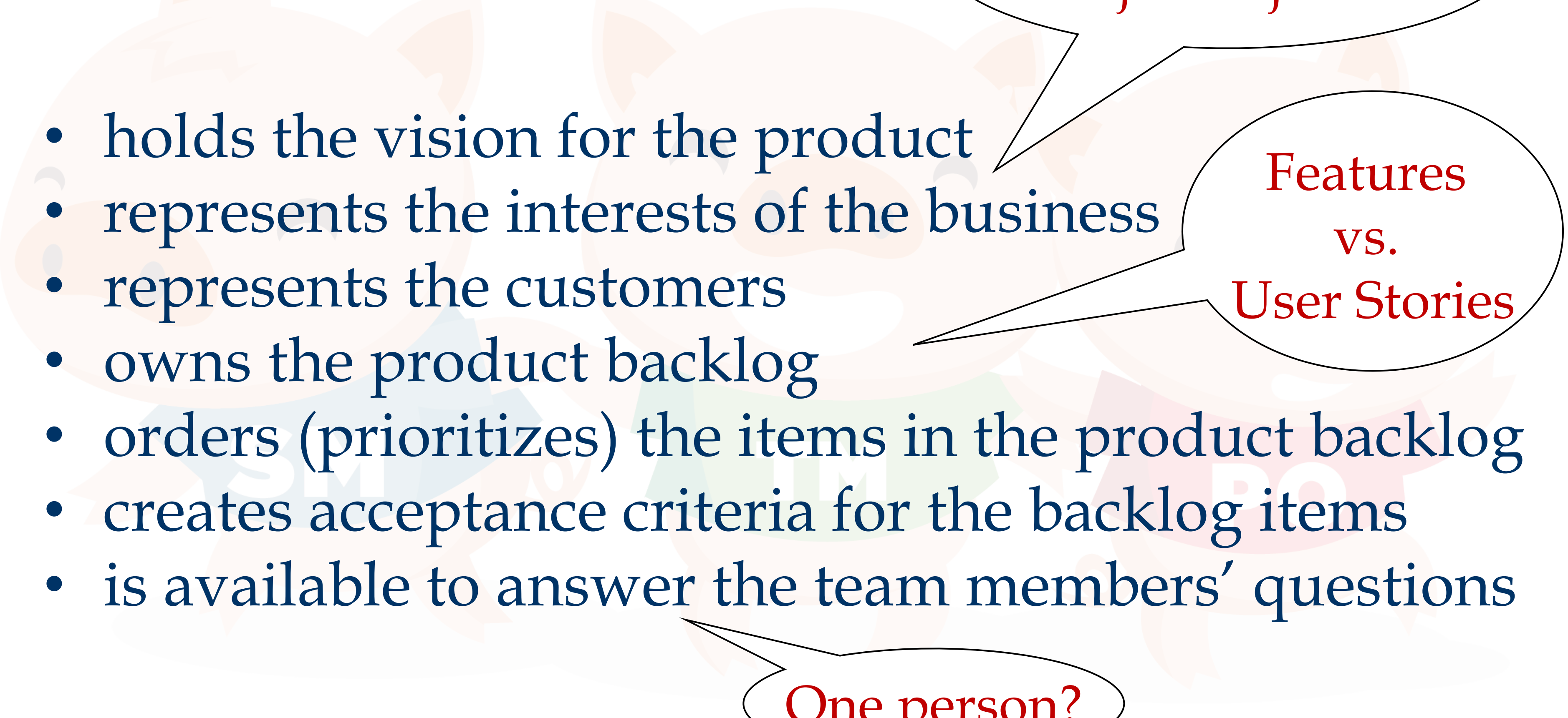
Product Owner

Business objectives
Project objectives

- holds the vision for the product
- represents the interests of the business
- represents the customers
- owns the product backlog
- orders (prioritizes) the items in the product backlog
- creates acceptance criteria for the backlog items
- is available to answer the team members’ questions

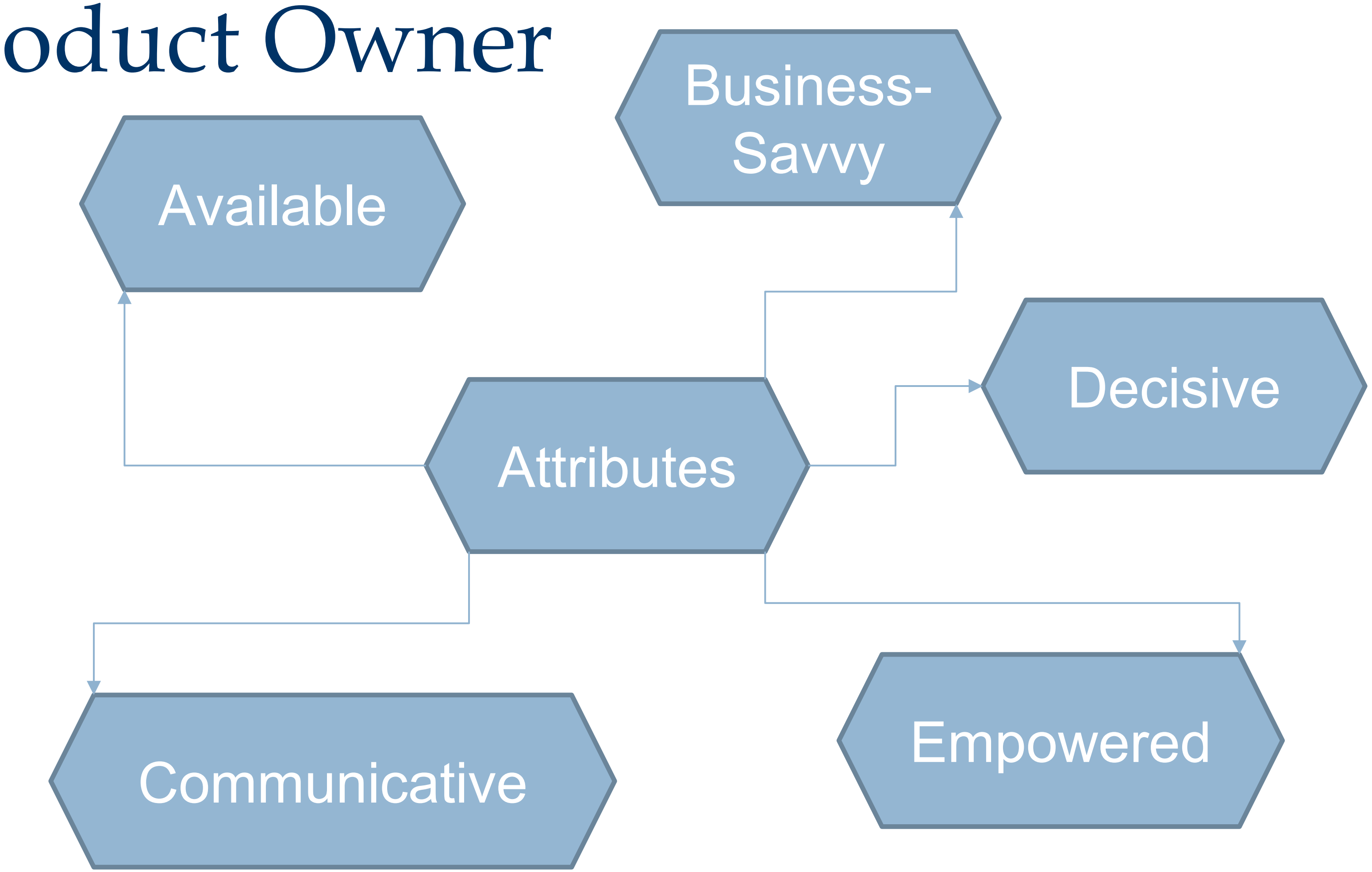
Features
vs.
User Stories

One person?



Scrum Roles

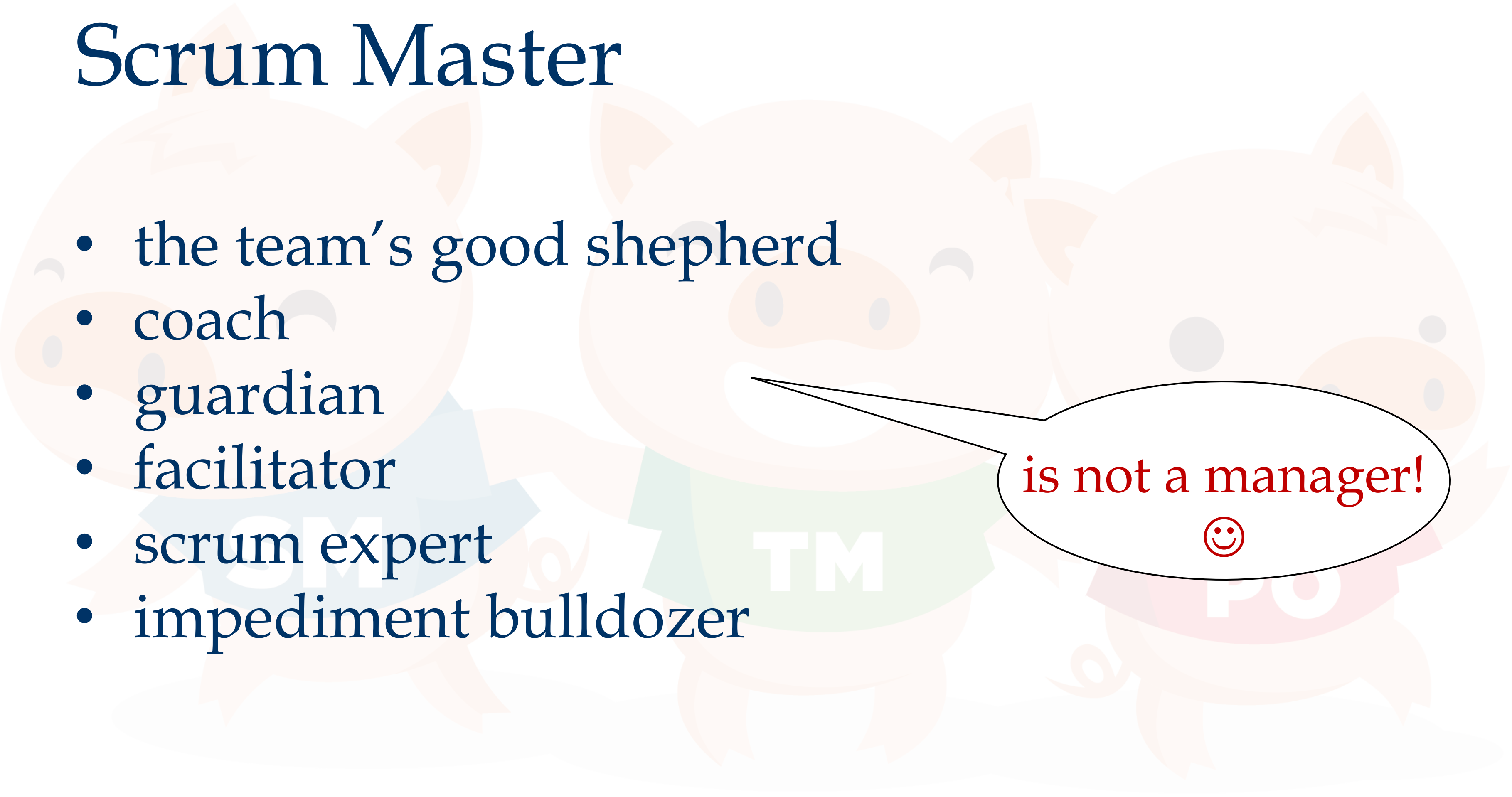
Product Owner



Scrum Roles

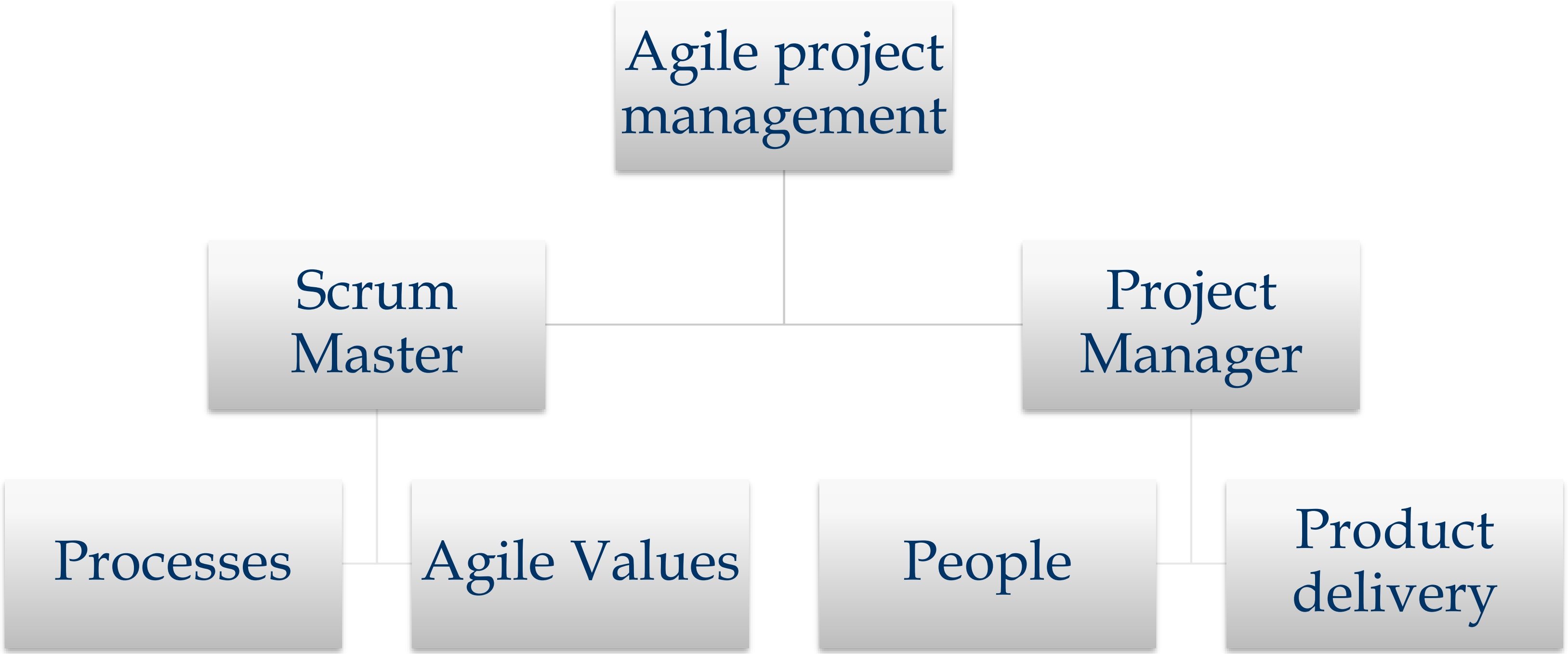
Scrum Master

- the team's good shepherd
- coach
- guardian
- facilitator
- scrum expert
- impediment bulldozer

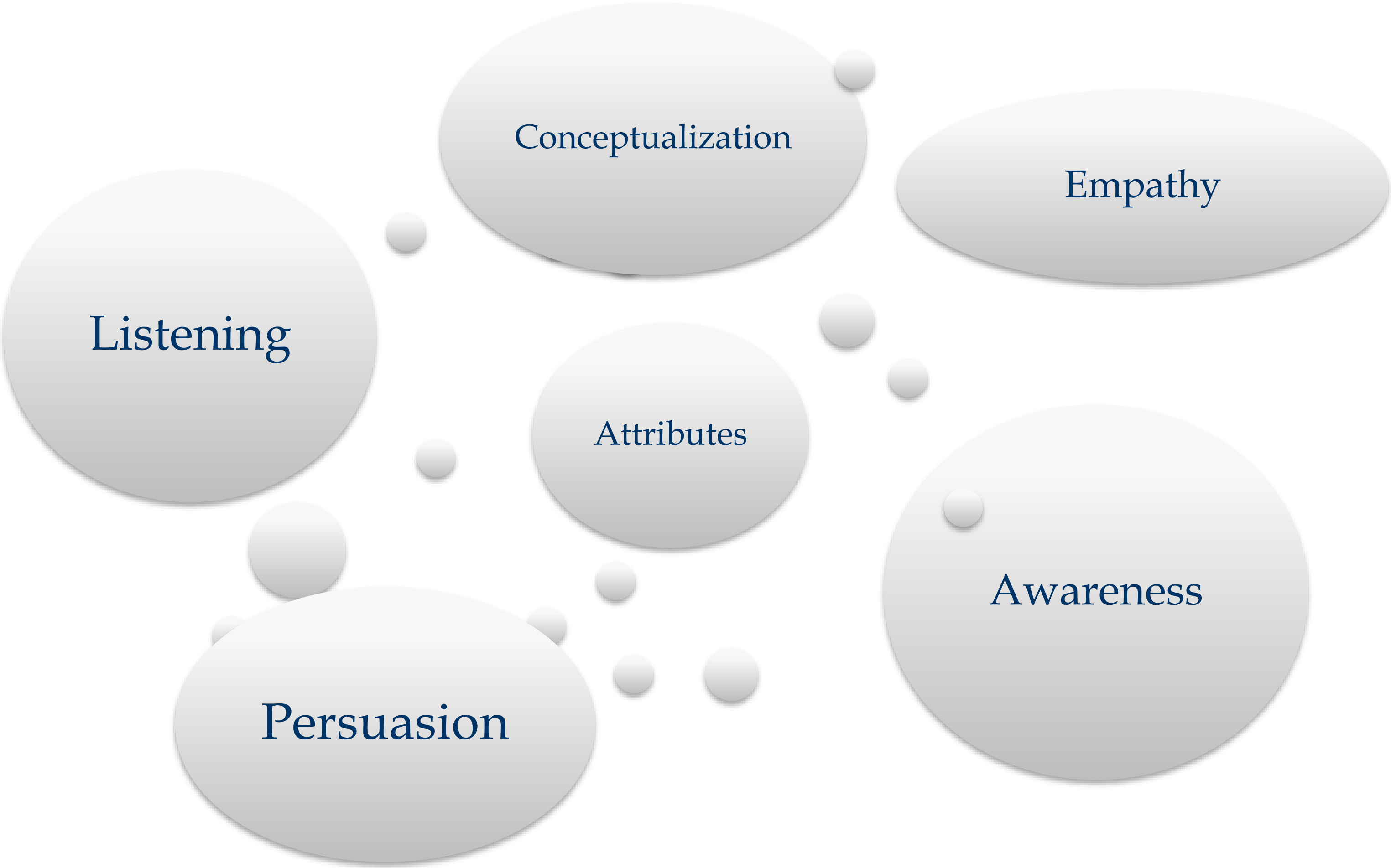


is not a manager!
☺

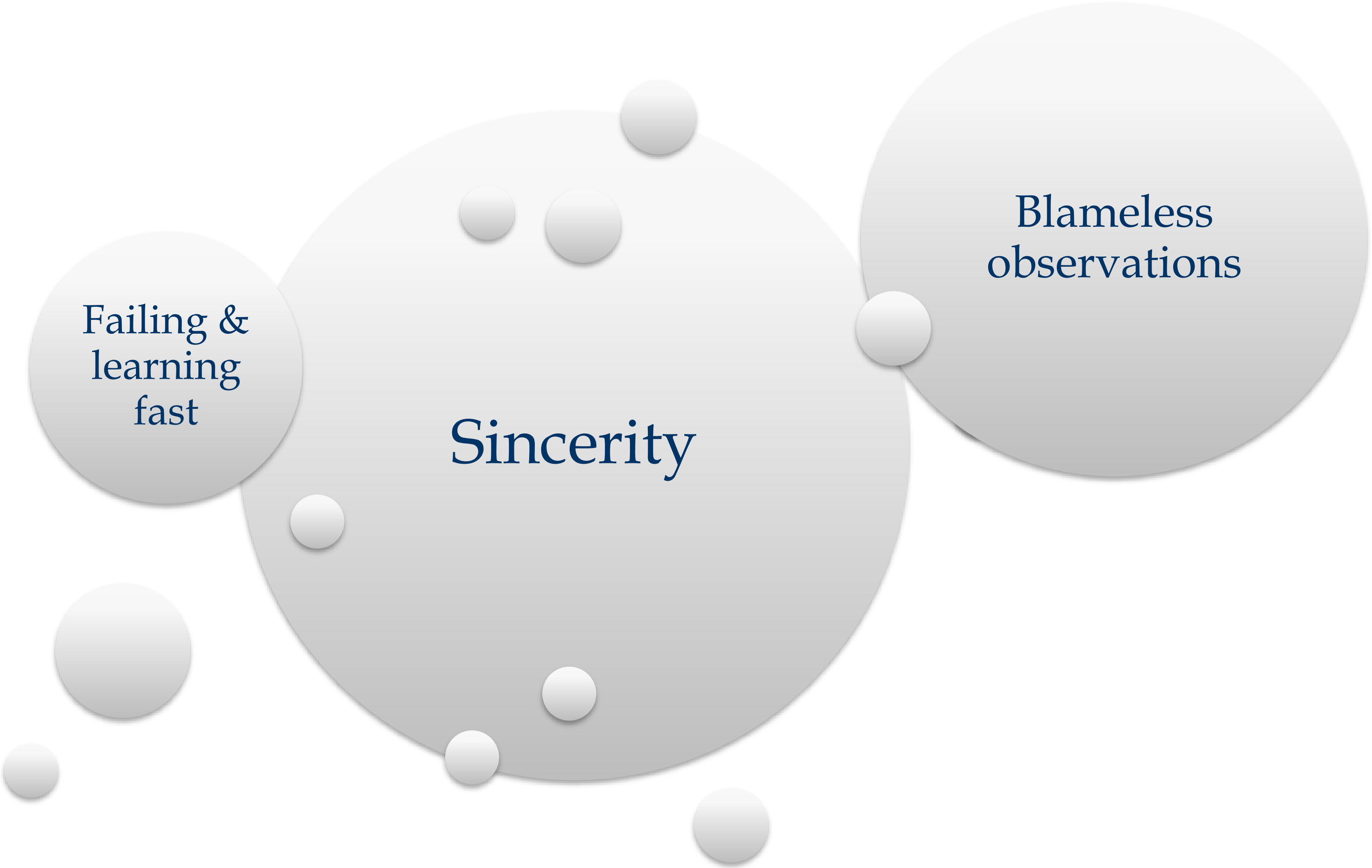
Scrum Master + PM



Scrum Master = Servant Leader



Scrum Master Encourages

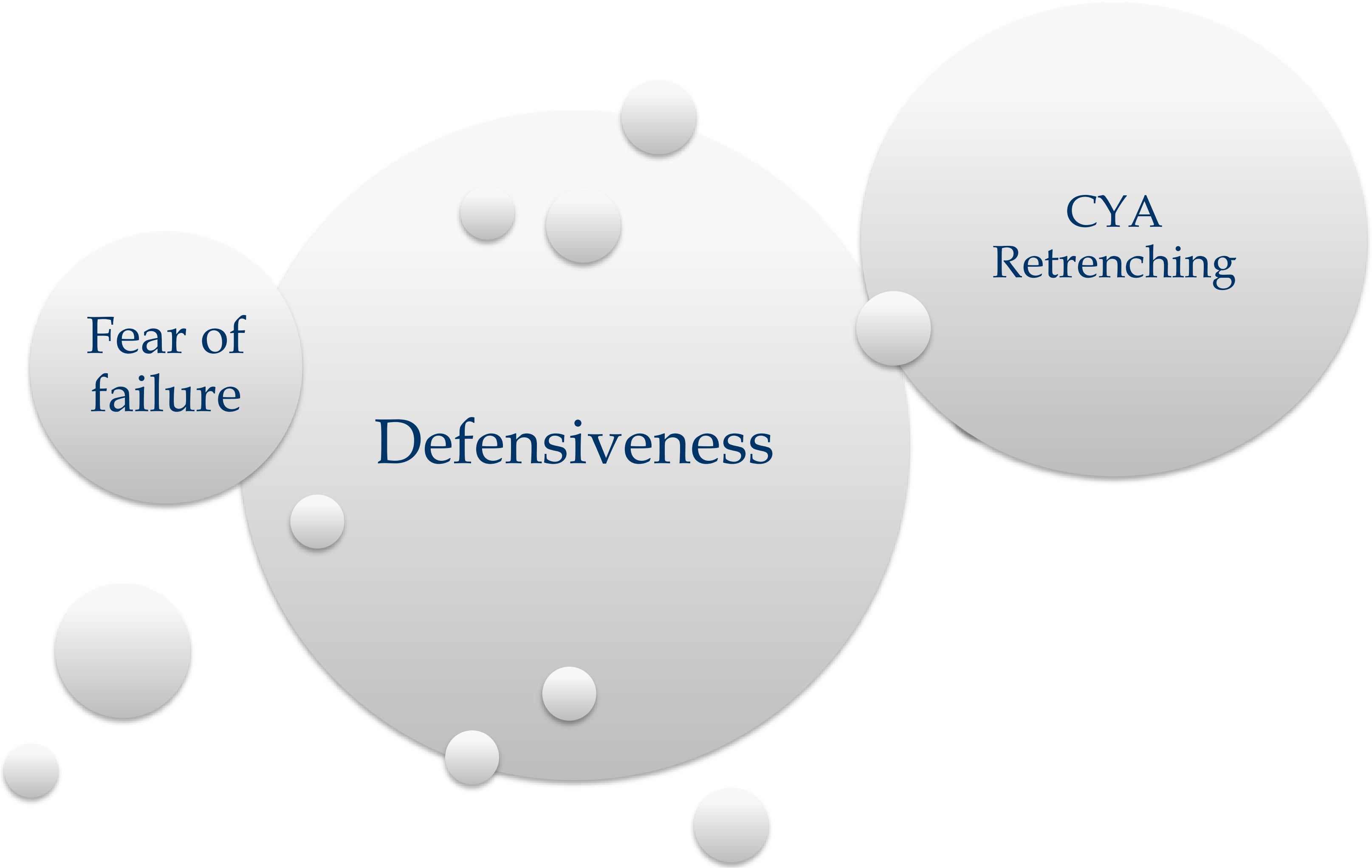


Failing & learning fast

Sincerity

Blameless observations

Scrum Master Discourages



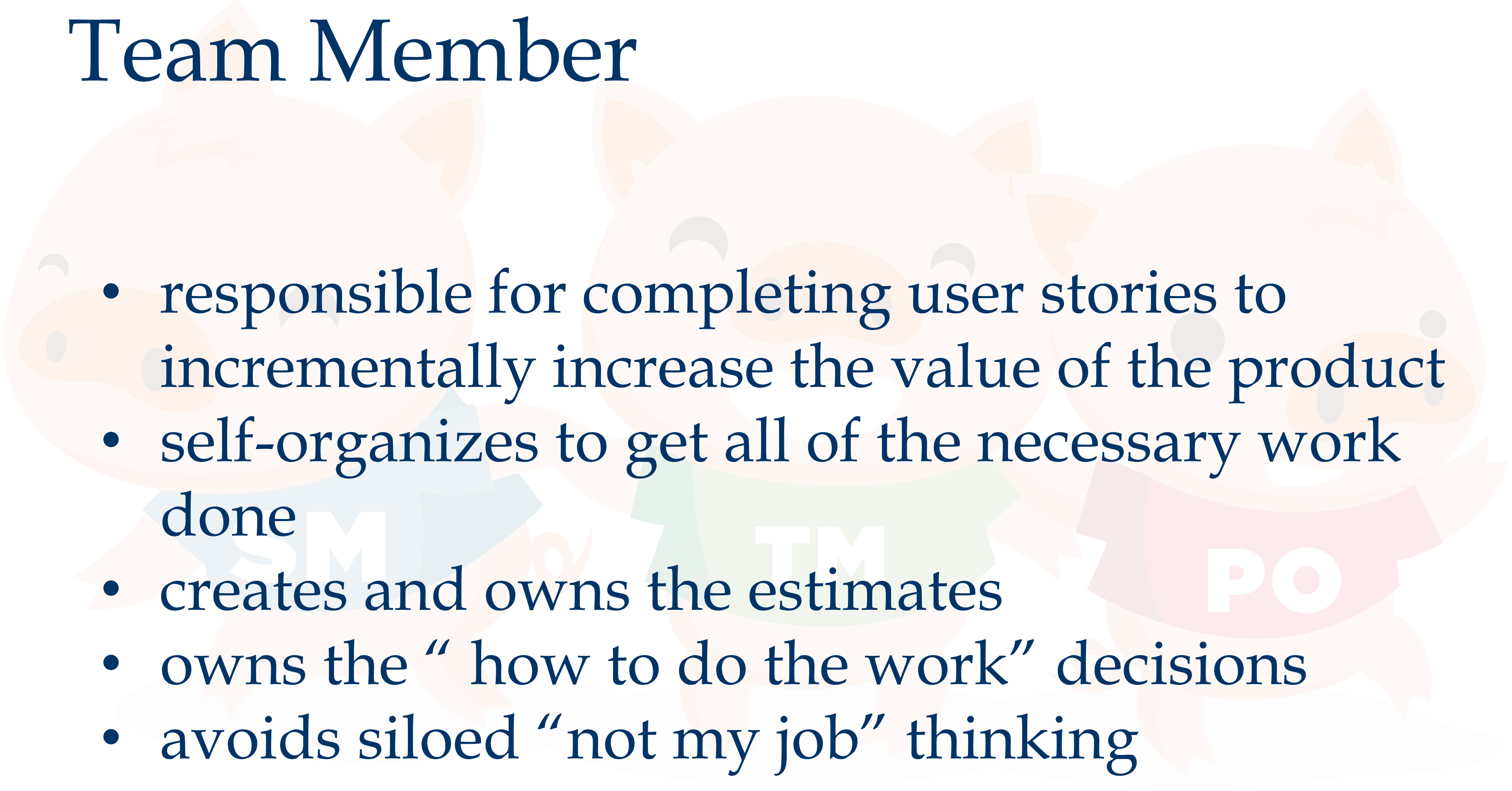
Fear of failure

Defensiveness

CYA Retrenching

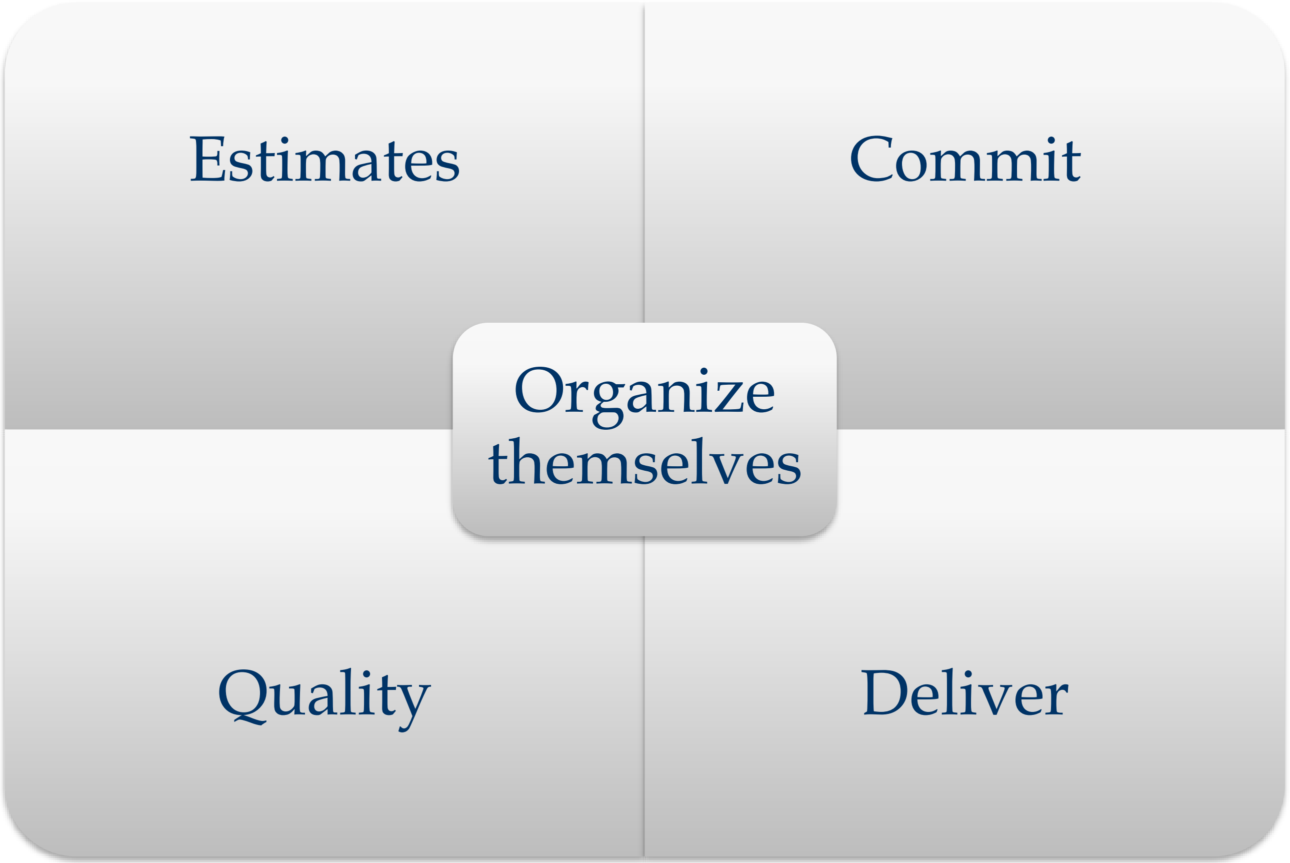
Scrum Roles

Team Member



- responsible for completing user stories to incrementally increase the value of the product
- self-organizes to get all of the necessary work done
- creates and owns the estimates
- owns the “ how to do the work” decisions
- avoids siloed “not my job” thinking

The team responsibilities



Estimates

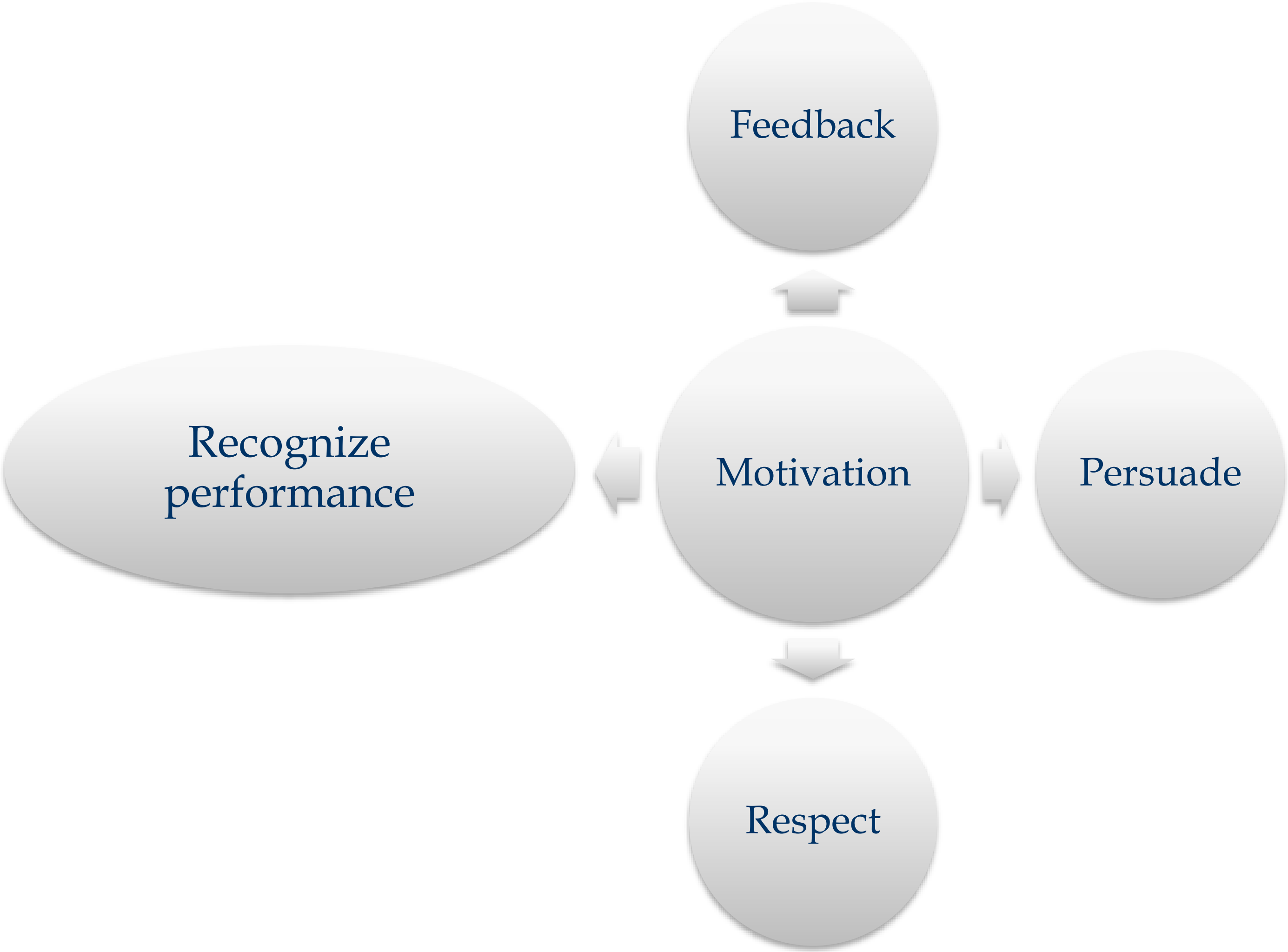
Commit

Organize themselves

Quality

Deliver

The team motivation



Feedback

Recognize performance

Motivation

Persuade

Respect