Advanced Certified Scrum Master (A-CSM) Training

via zoom

ROCKET NINE SOLUTIONS

powered by Agile Coffee

Victor Bonacci
Certified Scrum Trainer
Host of the *Agile Coffee Podcast*
victor@rocketninesolutions.com

V4.1 – Oct 2022

Getting Present

1) Minimize *distractions* in your surroundings

Turn off or mute your phone

Close doors & refrain from interruptions

Close your email

Turn off instant message apps

2) Briefly, write down your *intentions* for this session

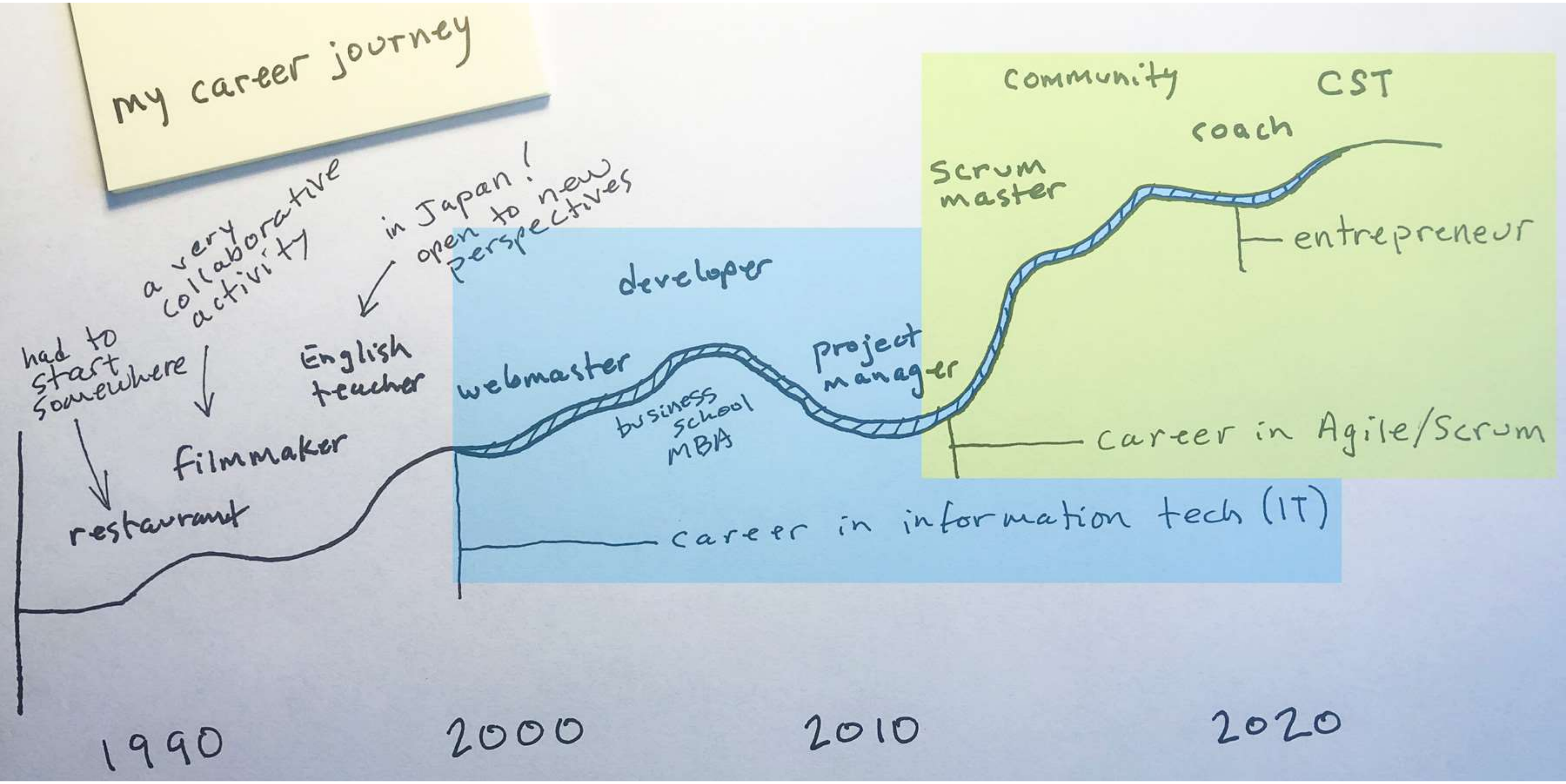
This note will stay private to you only.

Put in a visible place.

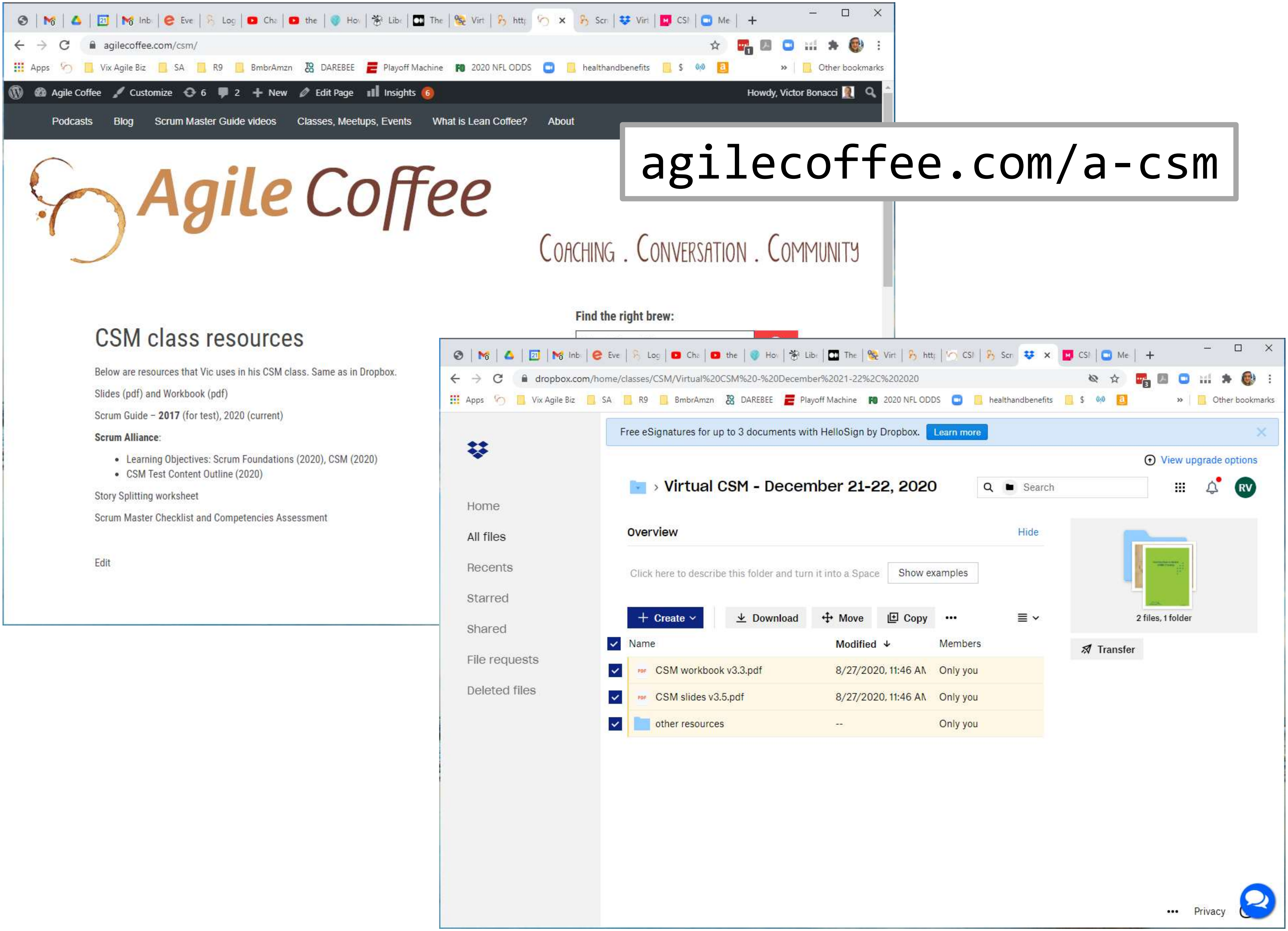
Twitter Check-In

Vic's Journey

Restauranteur, VISTA, Teacher, Filmmaker, Webmaster, Analyst, Programmer, Tech Lead, Product Manager, Project Manager, IT Manager, ScrumMaster, Product Owner, Agile Coach, Scrum Trainer…



agilecoffee.com/a-csm



CSM class resources

Below are resources that Vic uses in his CSM class. Same as in Dropbox.

Slides (pdf) and Workbook (pdf)

Scrum Guide – **2017** (for test), 2020 (current)

Scrum Alliance:

- Learning Objectives: Scrum Foundations (2020), CSM (2020)
- CSM Test Content Outline (2020)

Story Splitting worksheet

Scrum Master Checklist and Competencies Assessment

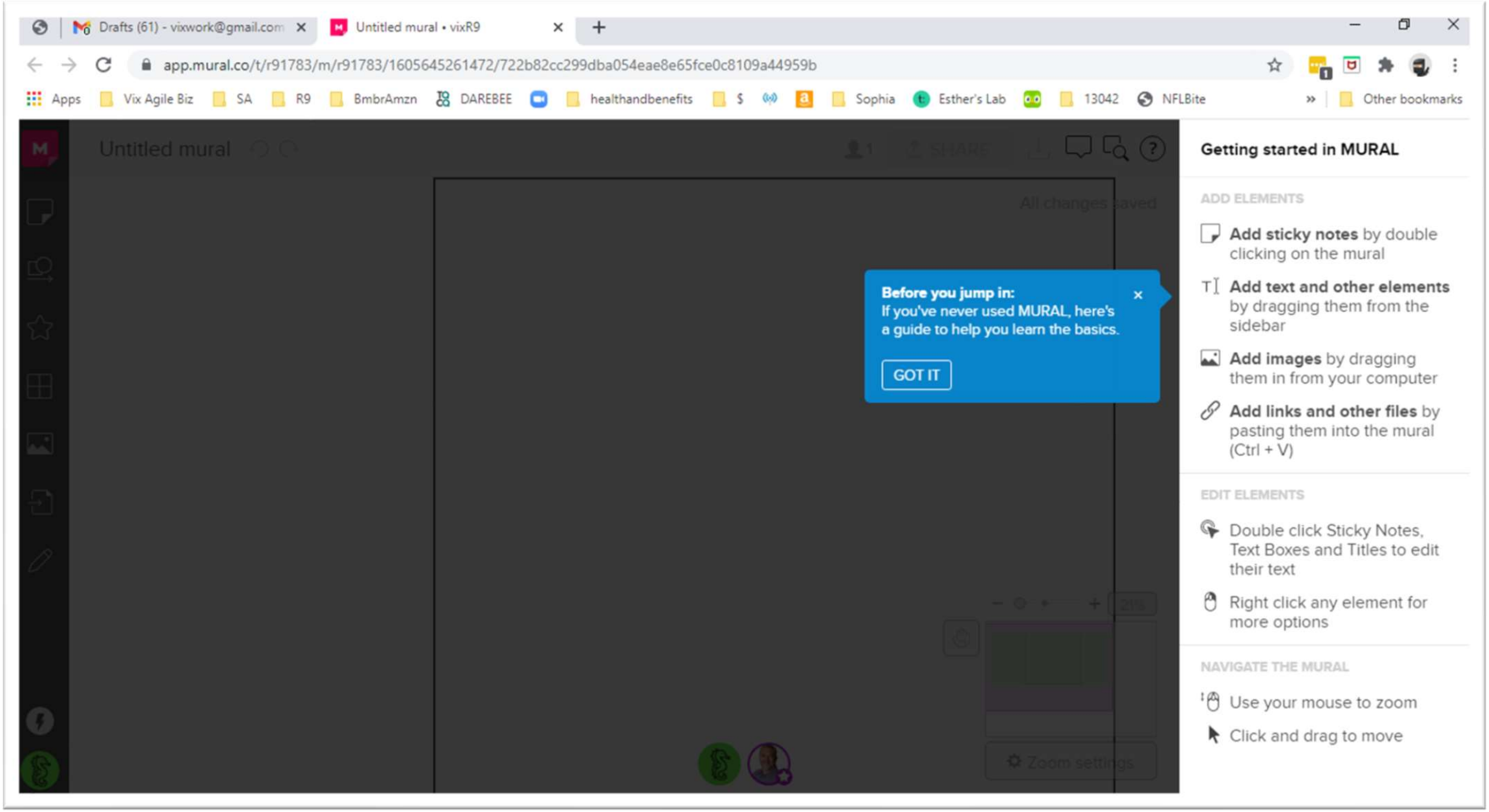
Edit

dropbox.com/home/classes/CSM/Virtual%20CSM%20-%20December%2021-22%2C%202020
Free eSignatures for up to 3 documents with HelloSign by Dropbox.
Learn more
View upgrade options
Home
All files
Recents
Starred
Shared
File requests
Deleted files
Search
RV
Overview
Hide
Click here to describe this folder and turn it into a Space
Show examples
Create
Download
Move
Copy
2 files, 1 folder
Transfer
Privacy

Virtual CSM - December 21-22, 2020

Name	Modified	Members
CSM workbook v3.3.pdf	8/27/2020, 11:46 AM	Only you
CSM slides v3.5.pdf	8/27/2020, 11:46 AM	Only you
other resources	--	Only you

MURAL



Before you jump in:
If you've never used MURAL, here's a guide to help you learn the basics.

GOT IT

Getting started in MURAL

ADD ELEMENTS

Add sticky notes by double clicking on the mural

Add text and other elements by dragging them from the sidebar

Add images by dragging them in from your computer

Add links and other files by pasting them into the mural (Ctrl + V)

EDIT ELEMENTS

Double click Sticky Notes, Text Boxes and Titles to edit their text

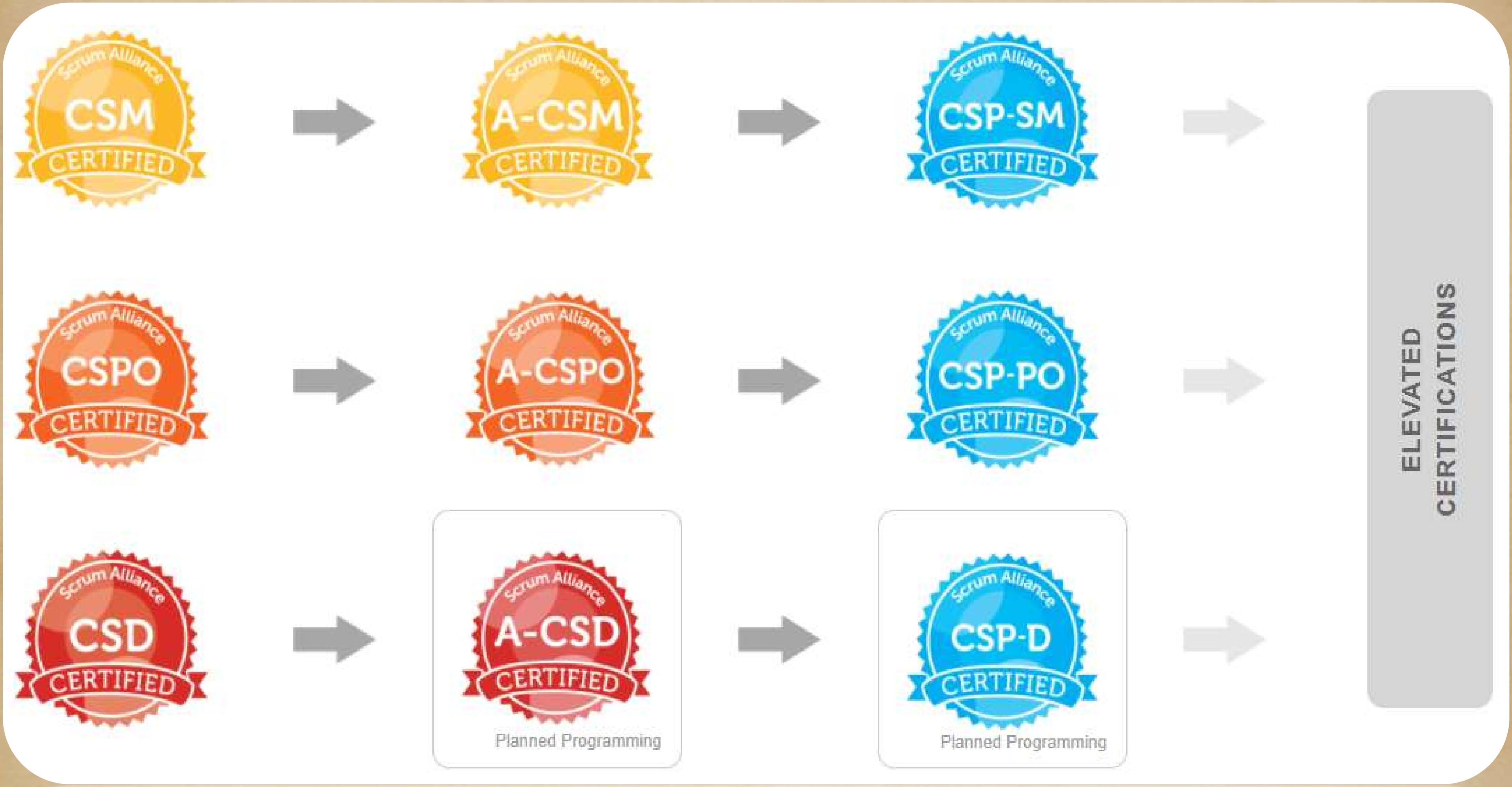
Right click any element for more options

NAVIGATE THE MURAL

Use your mouse to zoom

Click and drag to move

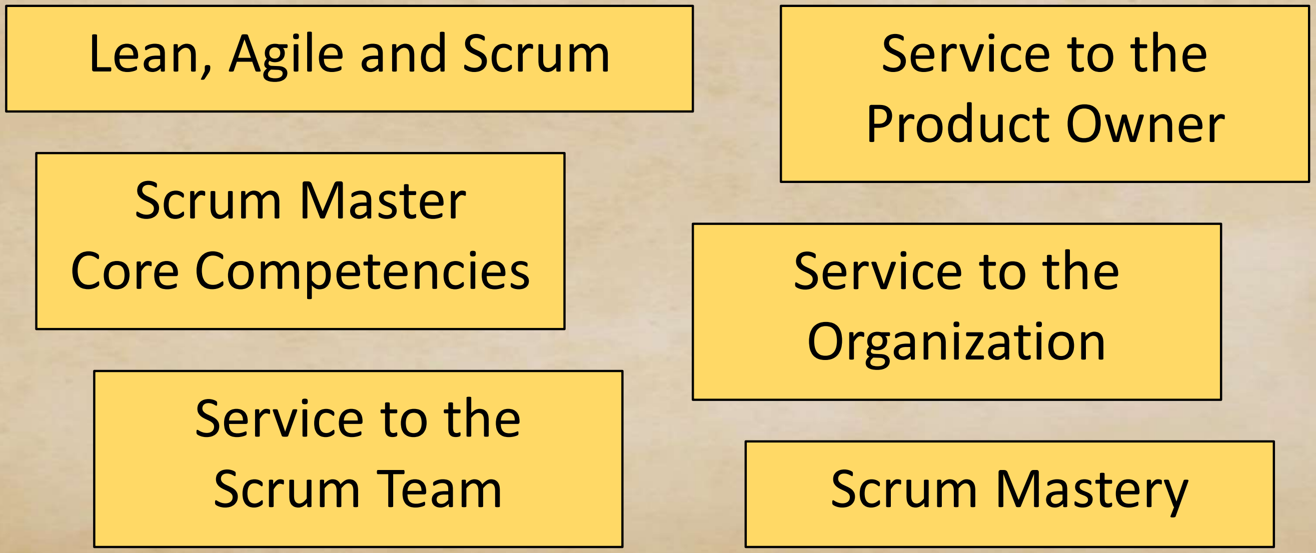
Scrum Alliance Certifications



Source: scrumalliance.org

Course Outline & Learning Objectives

- Visit the Scrum Alliance website for all current details for Advanced Certified ScrumMaster (A-CSM)
- The course Learning Objectives (LOs) are based on:
  - Scrum Guide: http://www.scrumguides.org/
  - Agile Manifesto, 4 values and 12 principles: http://agilemanifesto.org/
  - Scrum Values: https://www.scrumalliance.org/why-scrum/core-scrum-values-roles
- The A-CSM LOs fall into the following categories:



Lean, Agile and Scrum

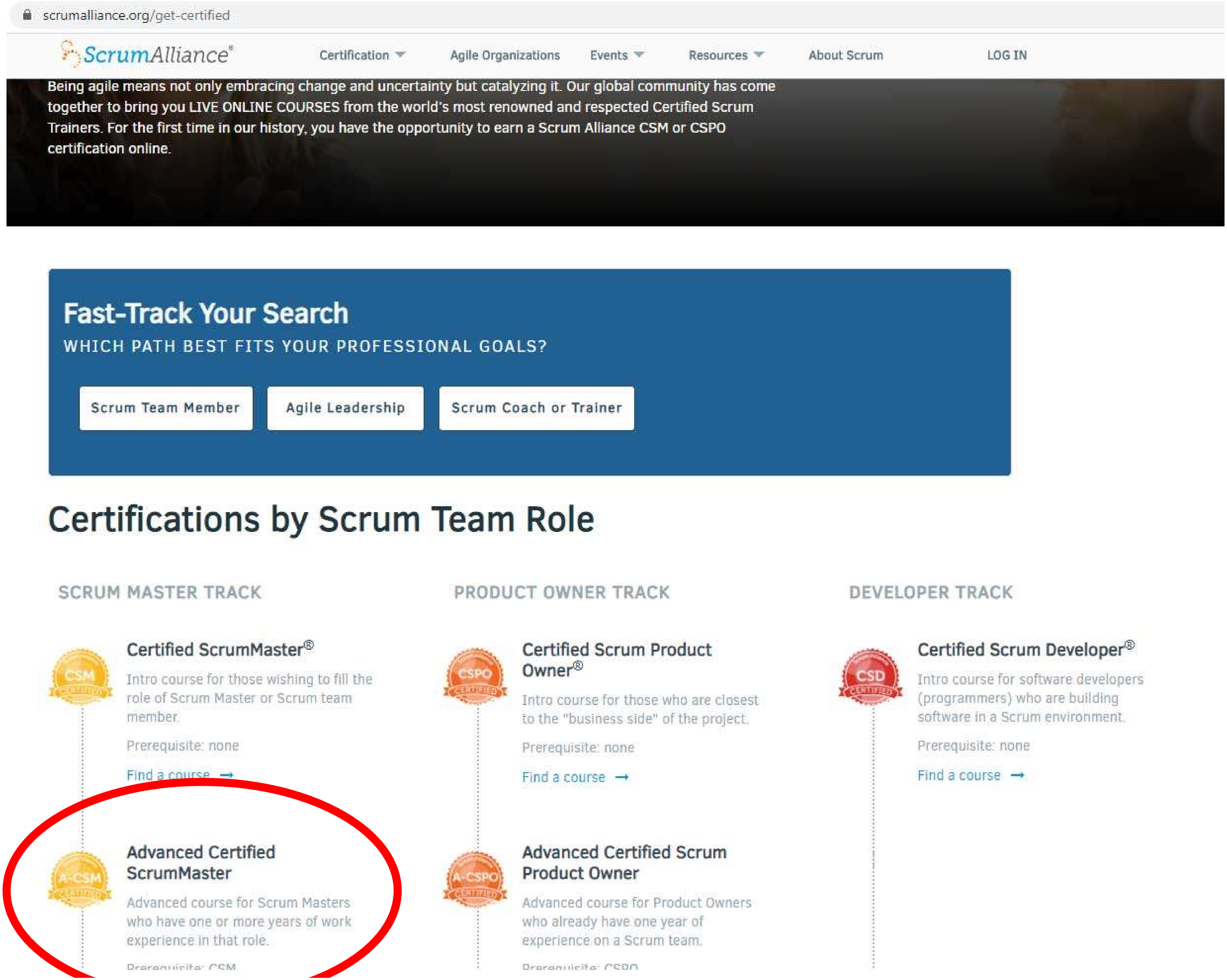
Service to the Product Owner

Scrum Master Core Competencies

Service to the Organization

Service to the Scrum Team

Scrum Mastery



Being agile means not only embracing change and uncertainty but catalyzing it. Our global community has come together to bring you LIVE ONLINE COURSES from the world's most renowned and respected Certified Scrum Trainers. For the first time in our history, you have the opportunity to earn a Scrum Alliance CSM or CSPO certification online.

Fast-Track Your Search

WHICH PATH BEST FITS YOUR PROFESSIONAL GOALS?

Certifications by Scrum Team Role

SCRUM MASTER TRACK

Certified ScrumMaster®

Intro course for those wishing to fill the role of Scrum Master or Scrum team member.

Prerequisite: none

Find a course →

Advanced Certified ScrumMaster

Advanced course for Scrum Masters who have one or more years of work experience in that role.

Prerequisite: CSM

PRODUCT OWNER TRACK

Certified Scrum Product Owner®

Intro course for those who are closest to the "business side" of the project.

Prerequisite: none

Find a course →

Advanced Certified Scrum Product Owner

Advanced course for Product Owners who already have one year of experience on a Scrum team.

Prerequisite: CSPO

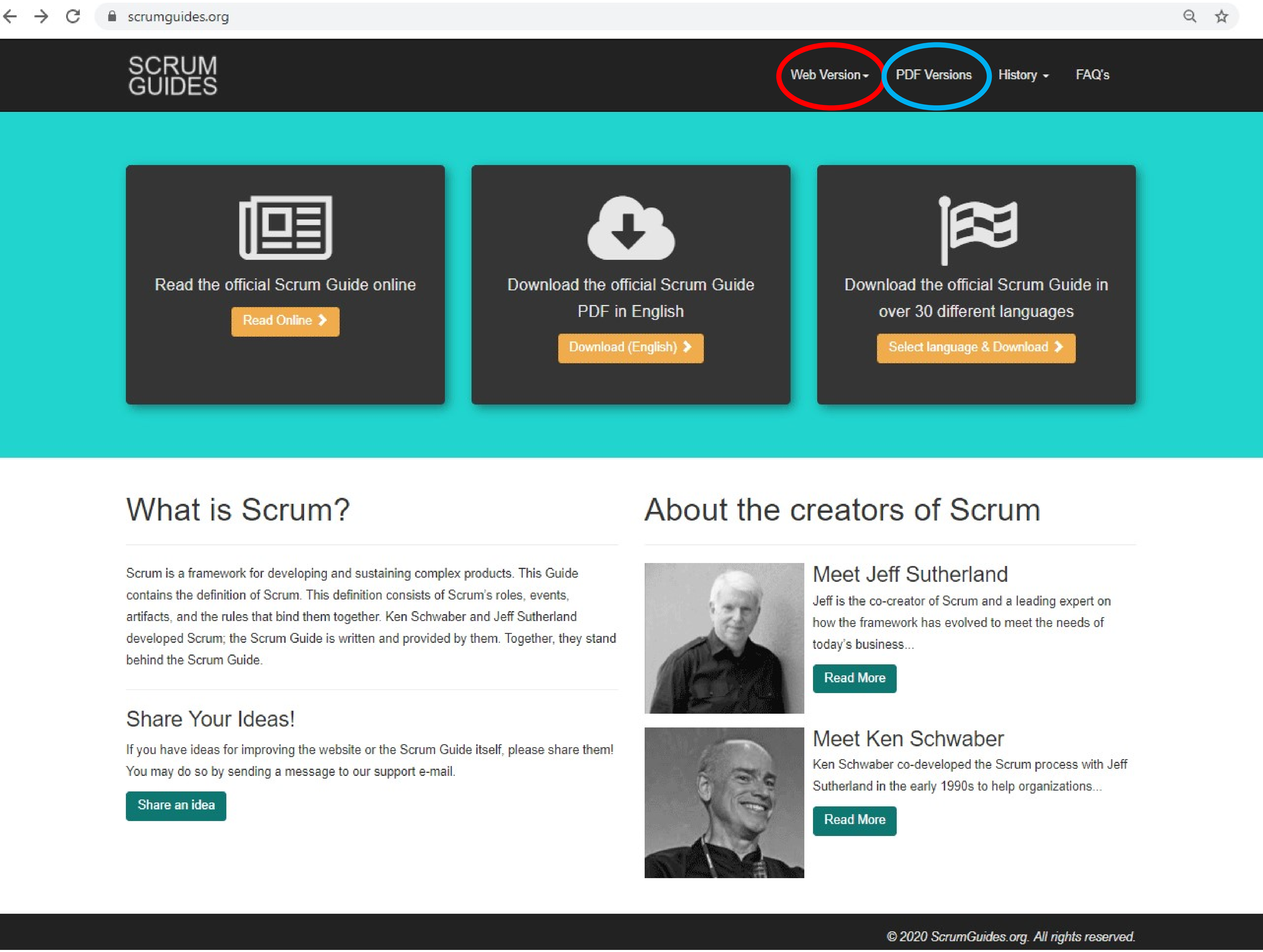
DEVELOPER TRACK

Certified Scrum Developer®

Intro course for software developers (programmers) who are building software in a Scrum environment.

Prerequisite: none

Find a course →



Read the official Scrum Guide online

Download the official Scrum Guide PDF in English

Download the official Scrum Guide in over 30 different languages

What is Scrum?

Scrum is a framework for developing and sustaining complex products. This Guide contains the definition of Scrum. This definition consists of Scrum's roles, events, artifacts, and the rules that bind them together. Ken Schwaber and Jeff Sutherland developed Scrum; the Scrum Guide is written and provided by them. Together, they stand behind the Scrum Guide.

Share Your Ideas!

If you have ideas for improving the website or the Scrum Guide itself, please share them! You may do so by sending a message to our support e-mail.

About the creators of Scrum

Meet Jeff Sutherland

Jeff is the co-creator of Scrum and a leading expert on how the framework has evolved to meet the needs of today's business...

Meet Ken Schwaber

Ken Schwaber co-developed the Scrum process with Jeff Sutherland in the early 1990s to help organizations...

Even Less Prescriptive

Over the years, the Scrum Guide started getting a bit more prescriptive.

The 2020 version aimed to bring Scrum back to being a minimally sufficient framework by removing or softening prescriptive language.

- removed Daily Scrum questions
- soften language around PBI attributes
- soften language around retro items in Sprint Backlog
- shortened Sprint cancellation section
- and more

Overall Simplification of Language for a Wider Audience

The 2020 Scrum Guide has placed an emphasis on eliminating redundant and complex statements as well as removing any remaining inference to IT work (e.g. testing, system, design, requirement, etc.)

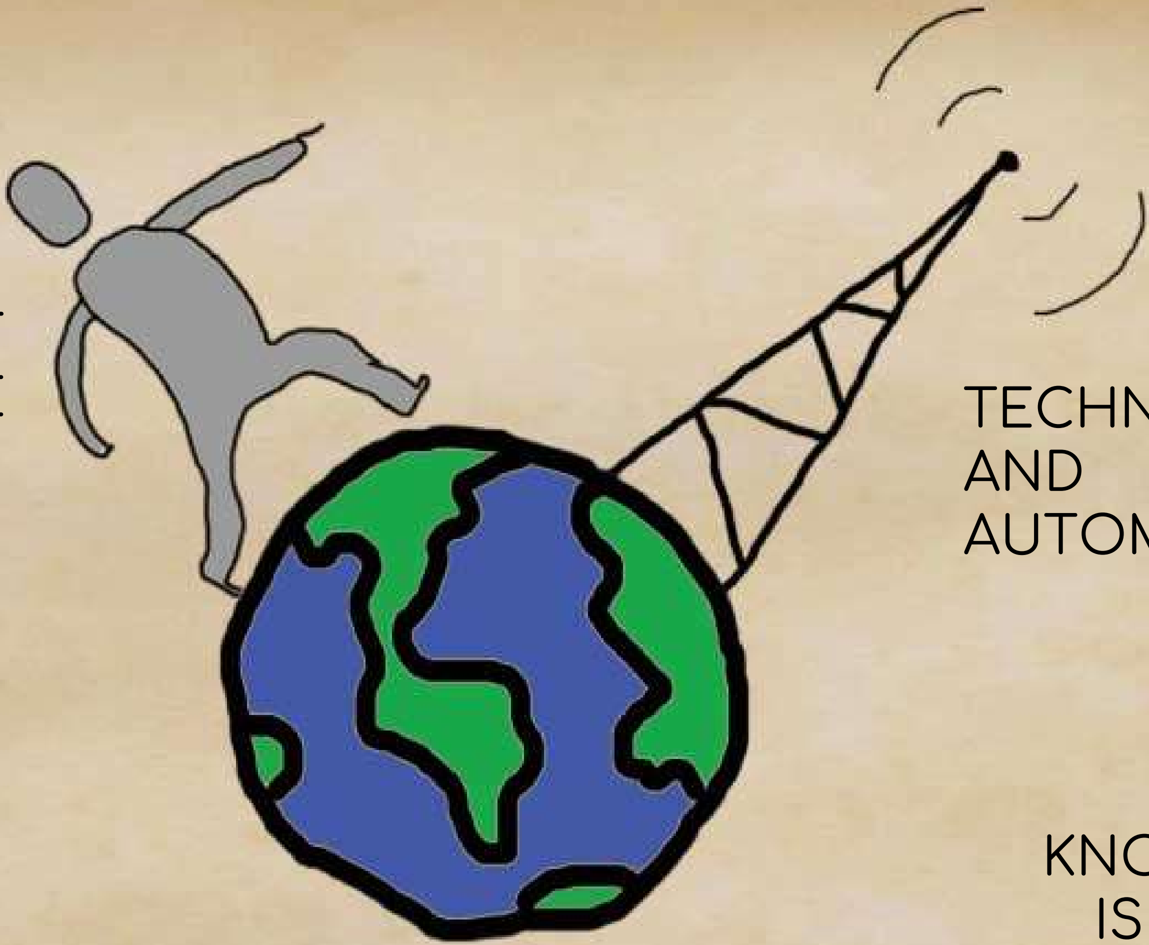
The Scrum Guide is now less than 13 pages.

What are the potential impacts to your team or organization?

As a group, imagine at least three potential impacts due to the adoption of the most recent definition of Scrum.

NEW SCRUM GUIDE

TO DO	DOING	DONE
Scrum Leadership	Why & What of Agile and Scrum	Introductions
Facilitation		
Coaching		
Organizational Change		
Self-managing Teams		
XP & Kanban		
Agile Product Development		
Teaching		
Scaling Scrum / Agile Organization		
Scrum Leadership		



PACE OF CHANGE

TECHNOLOGY AND AUTOMATION

CUSTOMER EXPECTATIONS

KNOWLEDGE IS NOW A COMMODITY

GLOBALIZATION

PRODUCT COMPLEXITY

HYPER-COMPETITION

INCREASED REGULATIONS

illustration inspired by Pete Behrens, Trail Ridge Consulting

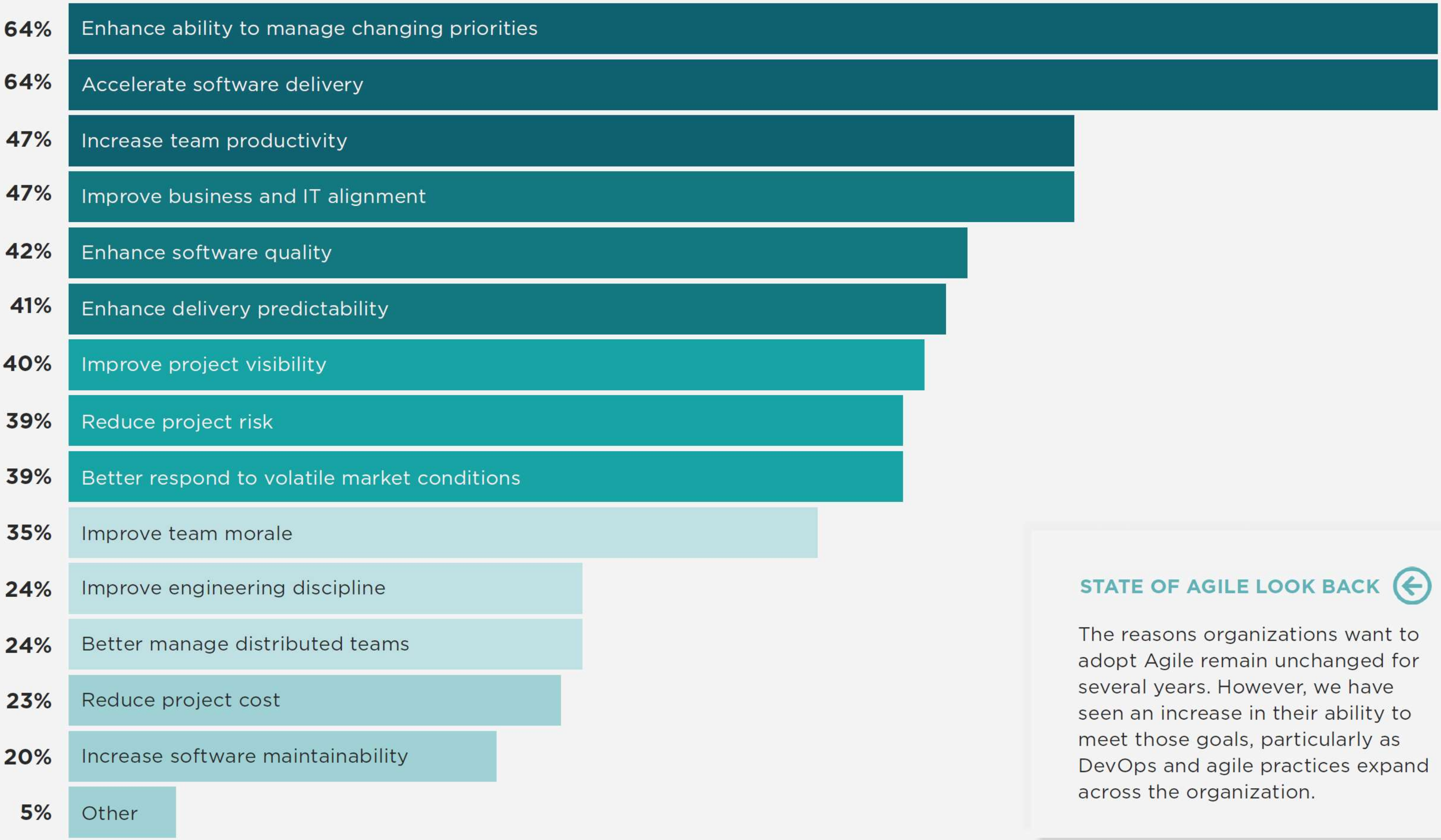
Agile Adoption

15th State of Agile Report (2021)

stateofagile.com

What were the most important reasons for adopting Agile within your team or organization?

**Respondents were able to select multiple responses to this question.*



STATE OF AGILE LOOK BACK

The reasons organizations want to adopt Agile remain unchanged for several years. However, we have seen an increase in their ability to meet those goals, particularly as DevOps and agile practices expand across the organization.

em·pir·i·cism

/əmˈpirəˌsizəm/

the theory that all knowledge is derived from sense-experience. Stimulated by the rise of experimental science, it developed in the 17th and 18th centuries, expounded in particular by John Locke, George Berkeley, and David Hume.

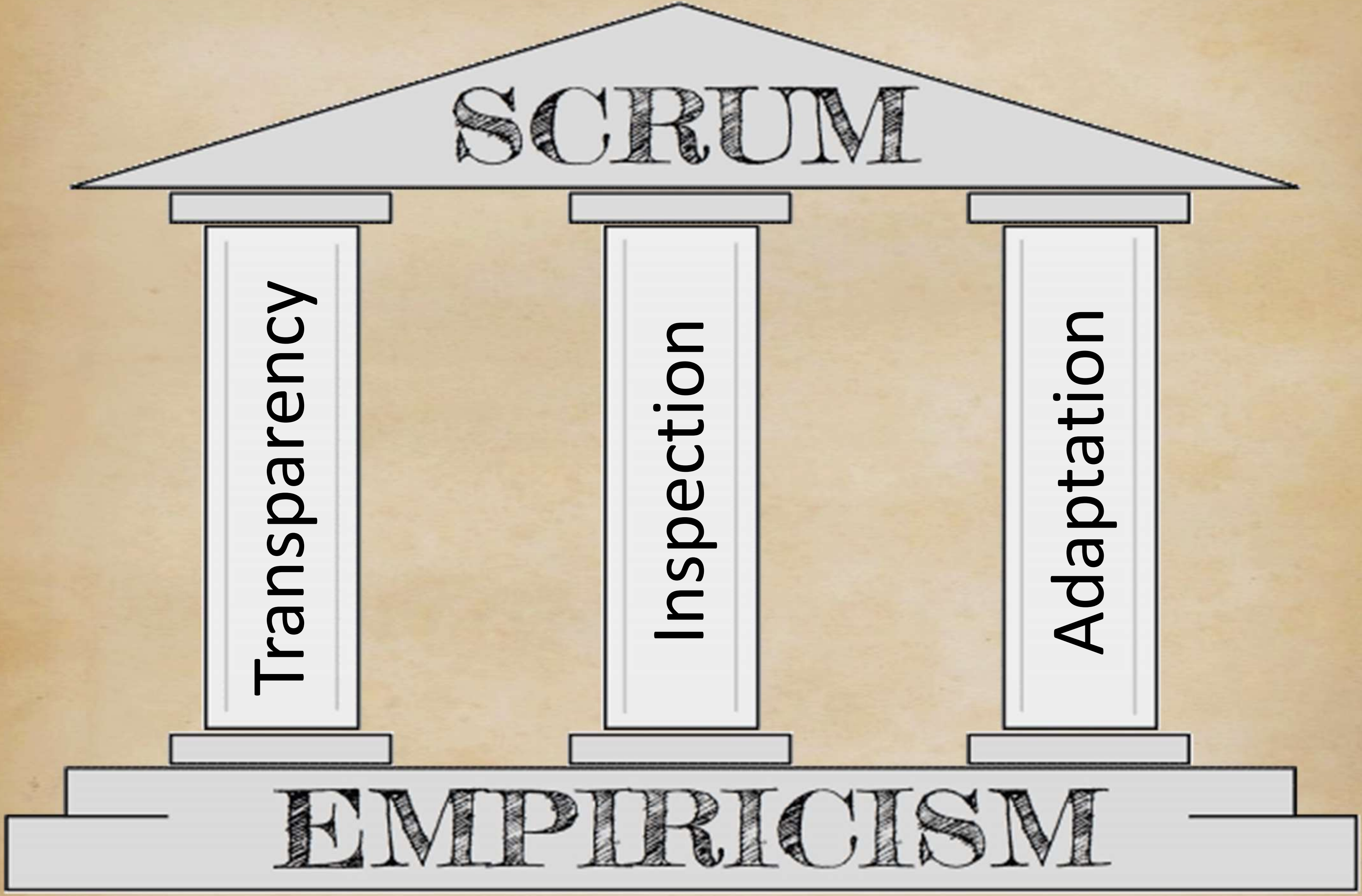
Empirical Process

- Variable Inputs
- Adaptable Process
- Variable Outputs
- Plan-Do-Study-Act

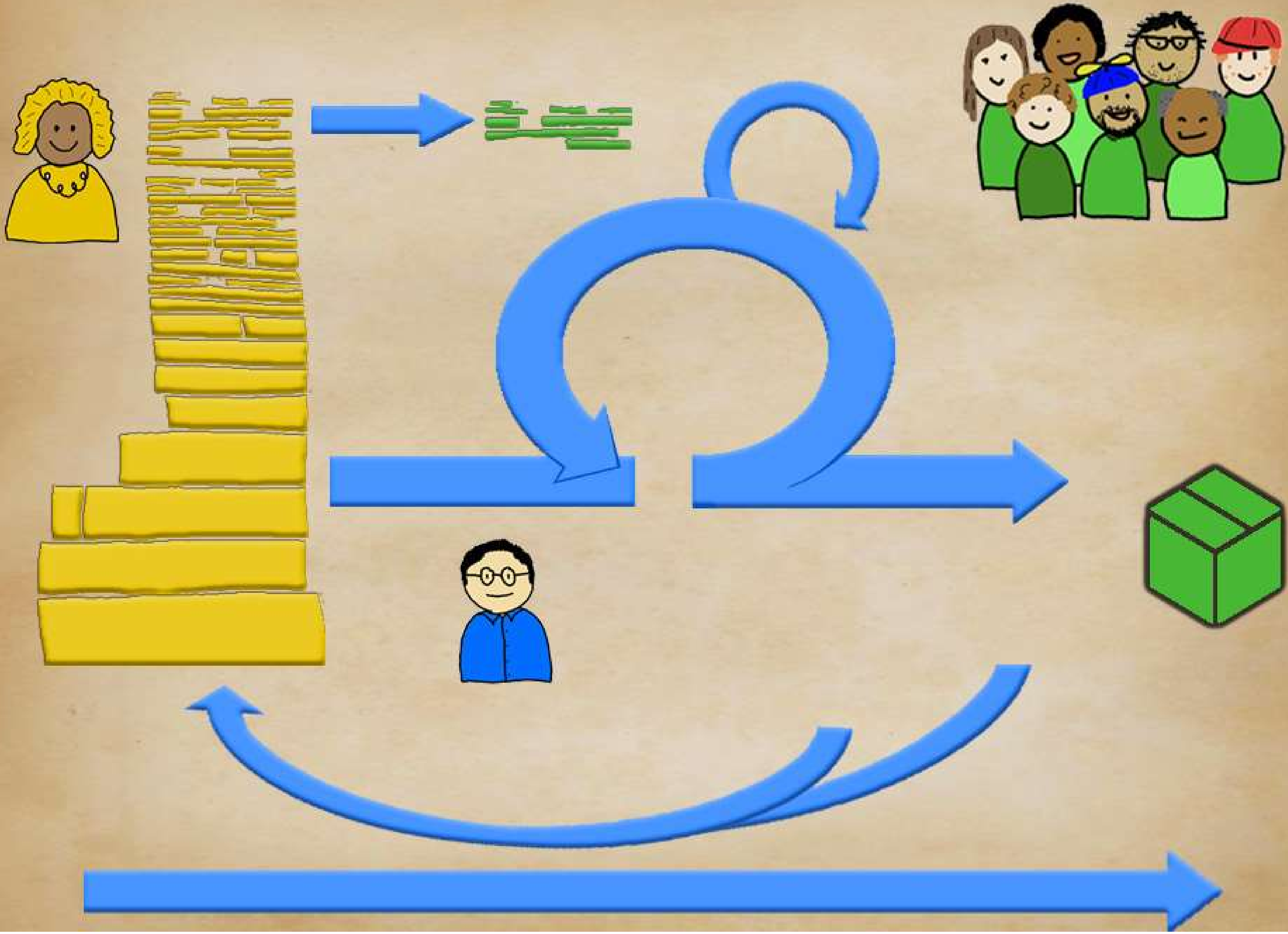
vs

Defined Process

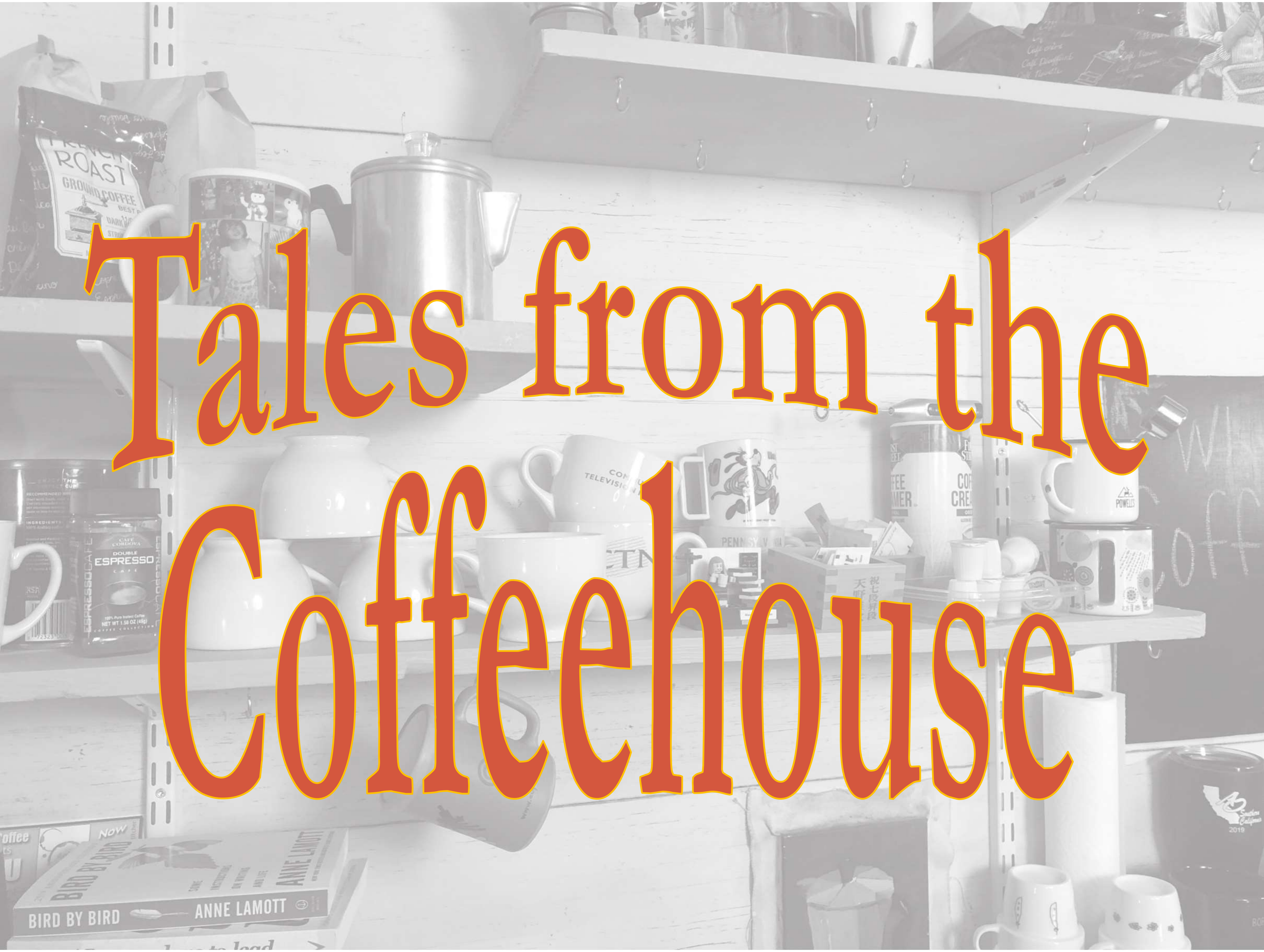
- Known Inputs
- Repeatable Process
- Expected Outputs
- Assumptions


SCRUM
Transparency
Inspection
Adaptation
EMPIRICISM

Scrum Framework



Tales from the Coffeehouse



Review your Scenario.
As a group, discuss whether Transparency, Inspection and Adaptation are working effectively for the Scrum Team and how they can be improved.

Empiricism: Scrum Events

Sprint:

Sprint Planning:

Daily Scrum:

Sprint Review:

Sprint Retrospective:

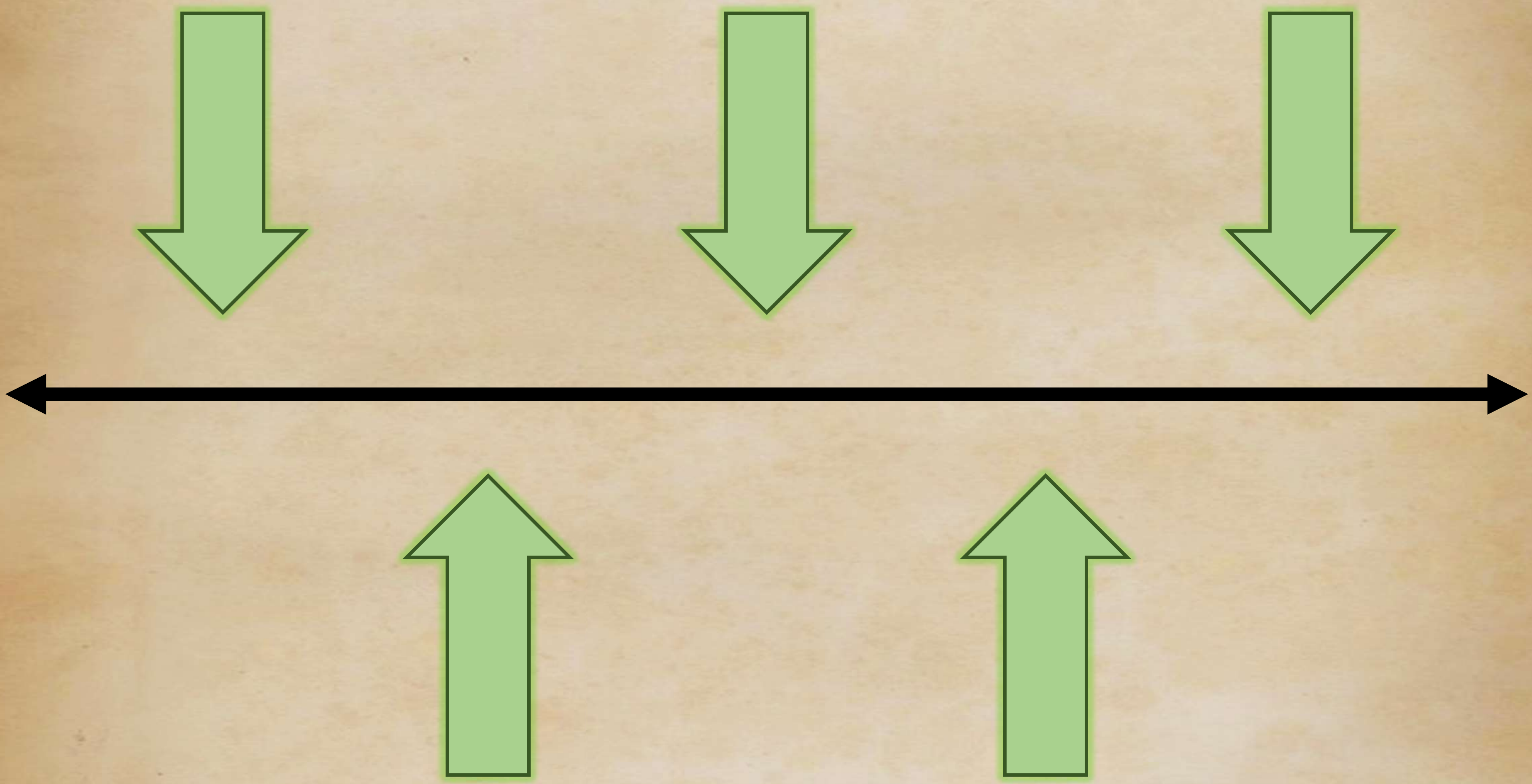
Empiricism: Scrum Artifacts

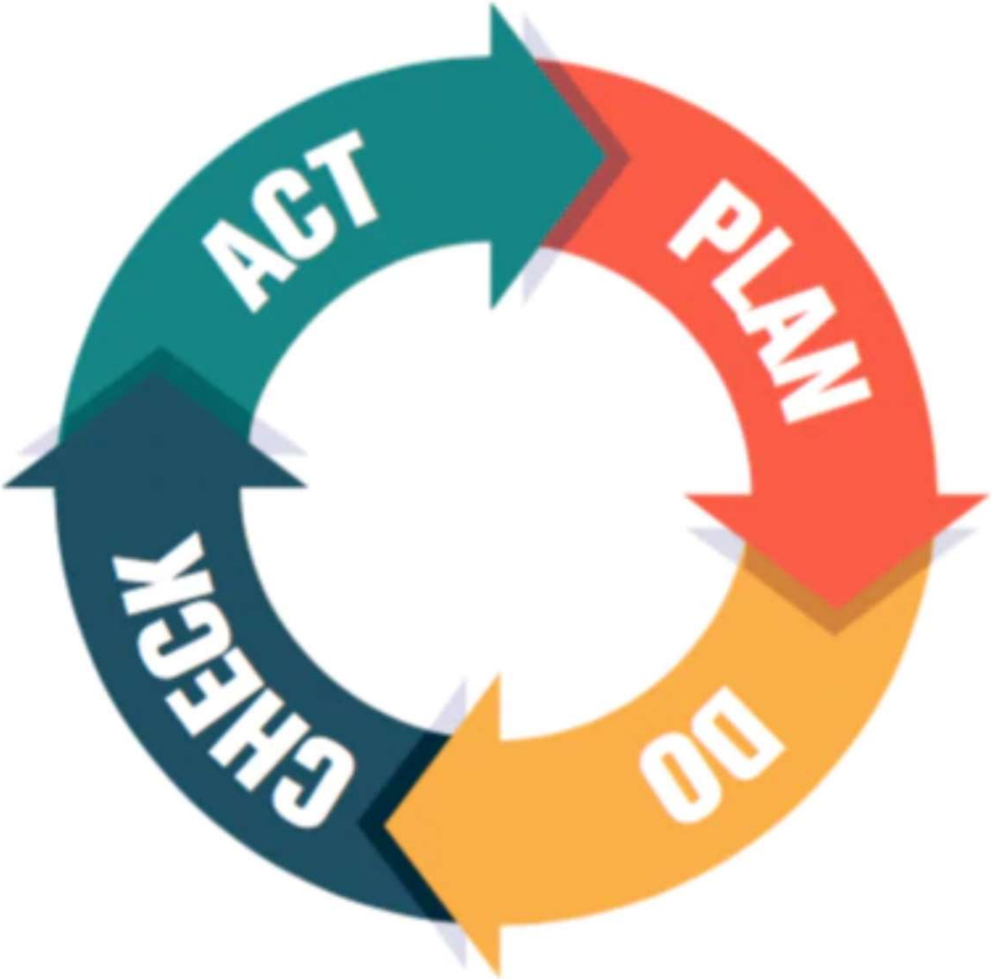
Product Backlog:

Sprint Backlog:

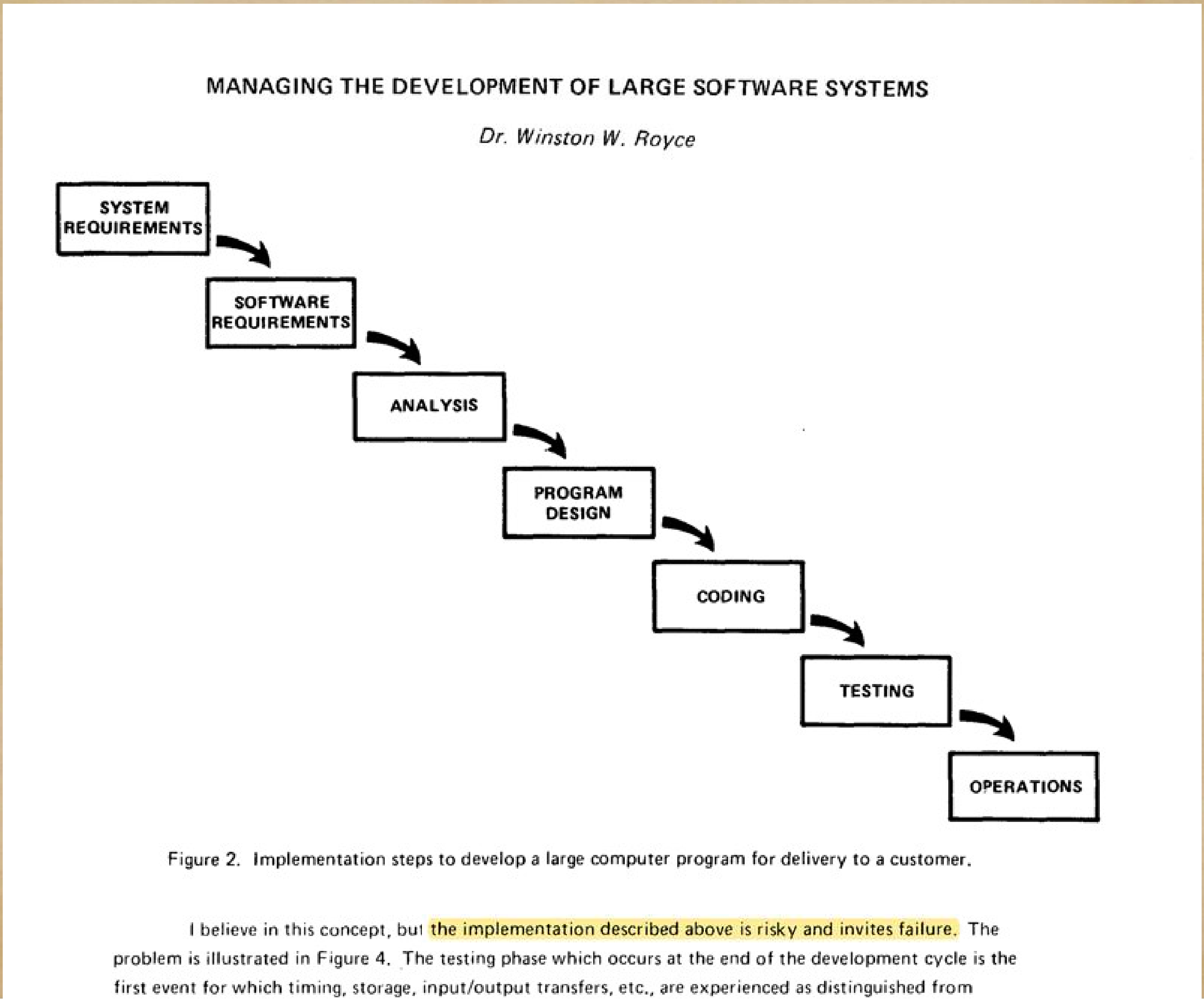
Increment(s):

Timeline: Agile and Scrum





Increase Quality by Reducing Waste Lean Manufacturing



MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS

Dr. Winston W. Royce

SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
OPERATIONS

Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

I believe in this concept, but the implementation described above is risky and invites failure. The problem is illustrated in Figure 4. The testing phase which occurs at the end of the development cycle is the first event for which timing, storage, input/output transfers, etc., are experienced as distinguished from

New New Product Development Game (1986)

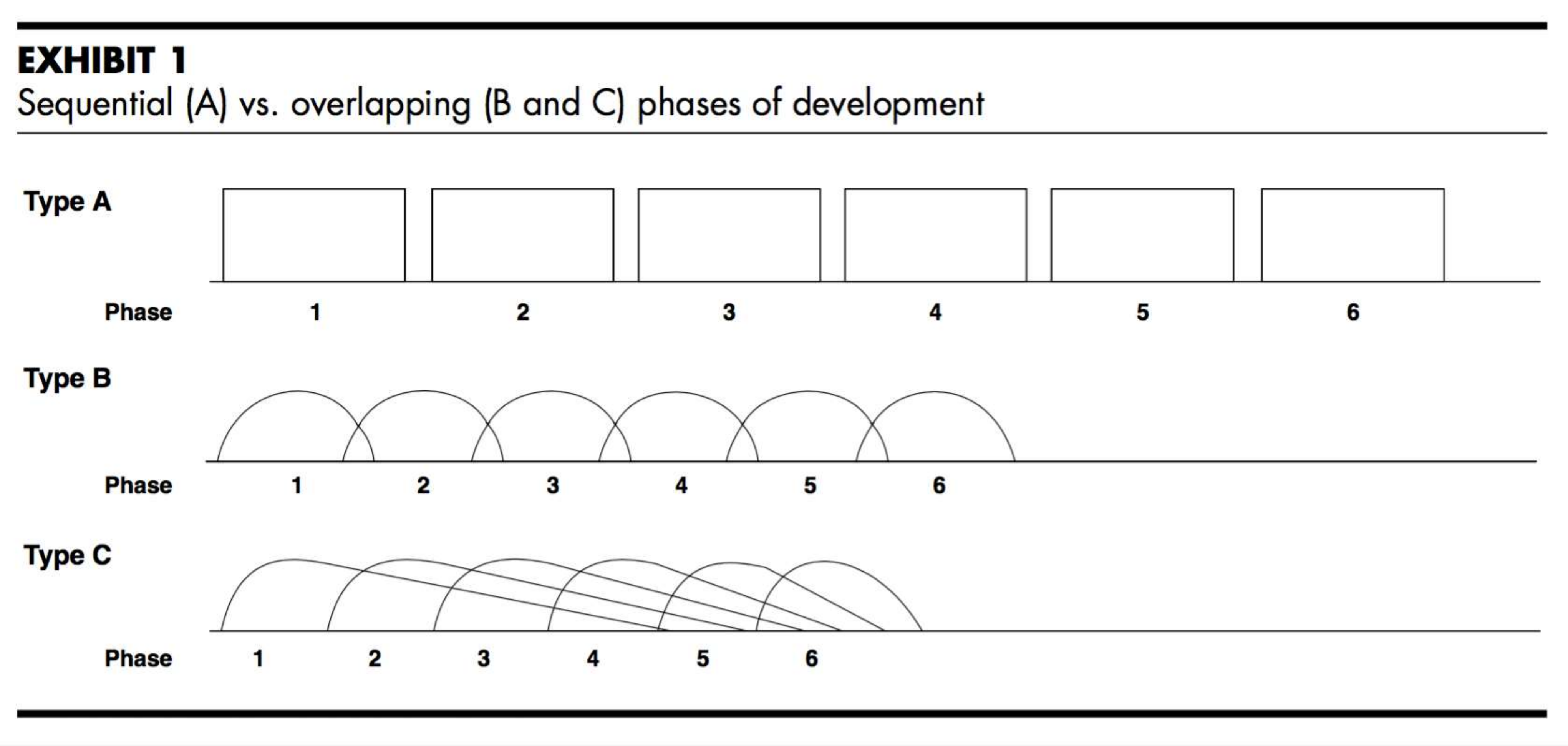


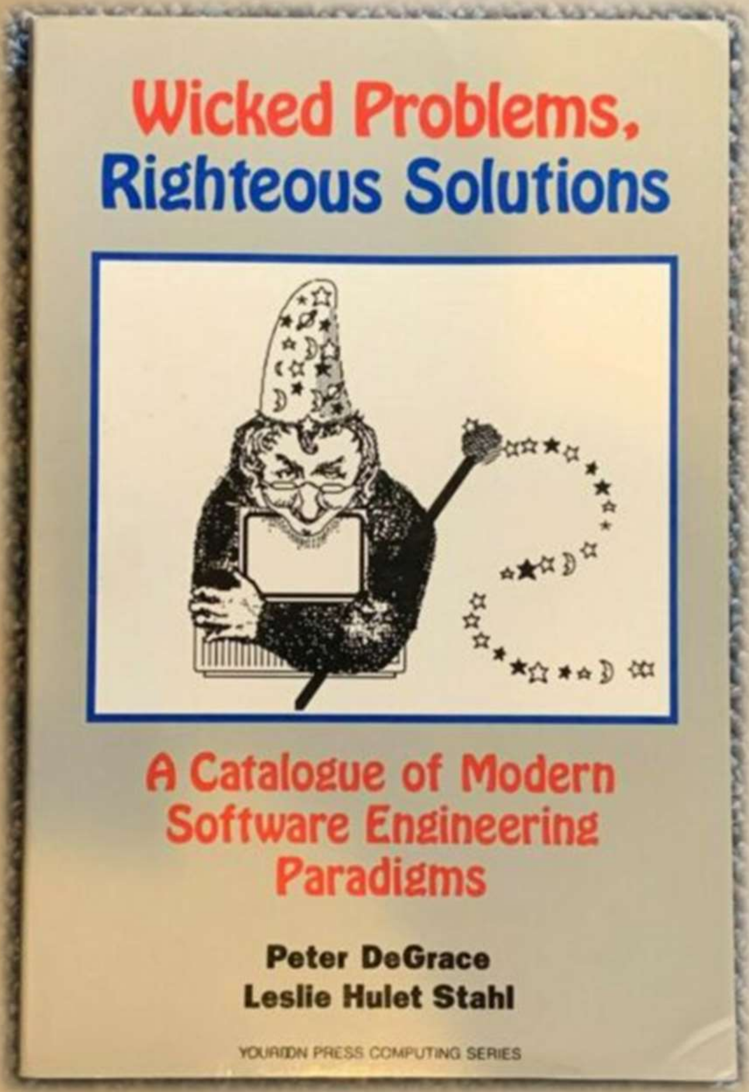
EXHIBIT 1
Sequential (A) vs. overlapping (B and C) phases of development

https://hbr.org/1986/01/the-new-new-product-development-game

6

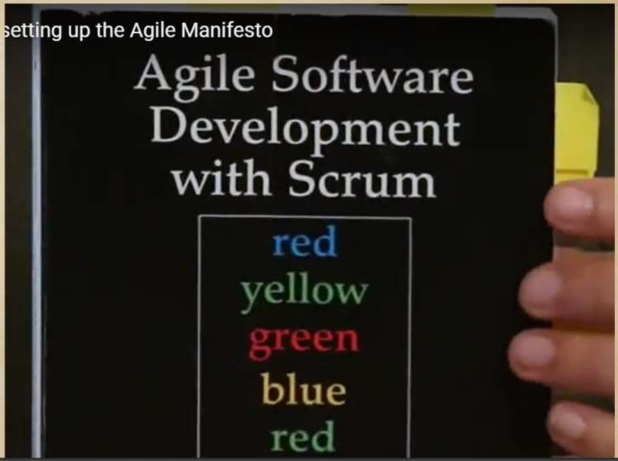
World Cup USA 94
The 1990's
HARRY POTTER
J.K. ROWLING
HOGWARTS EXPRESS
Seinfeld
George
intel pentium
eBay
YAHOO!
2370
62
ROSS PEROT
The 1

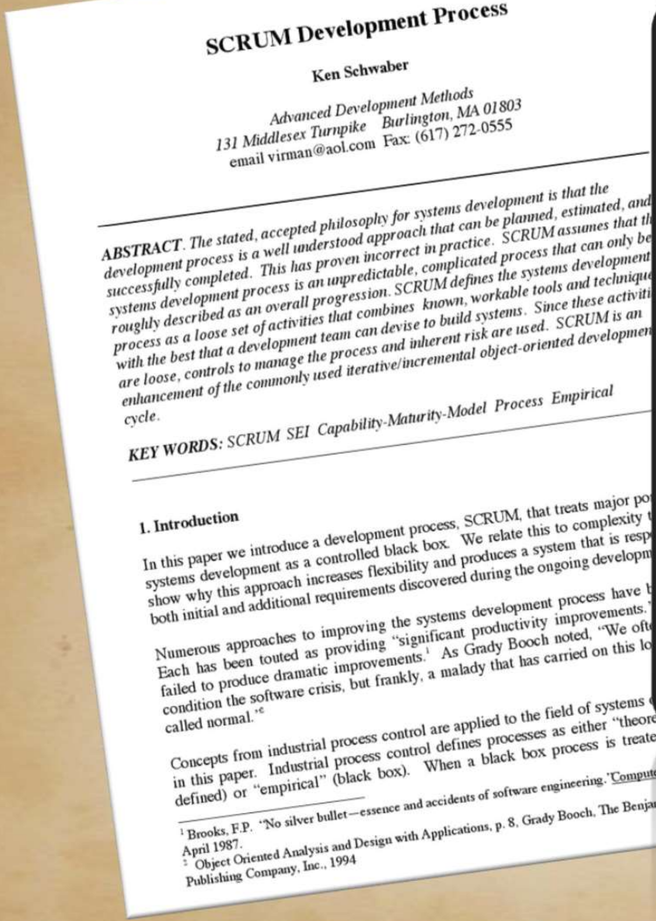
Wicked Problems, Righteous Solutions (1990)


Wicked Problems,
Righteous Solutions
A Catalogue of Modern Software Engineering Paradigms
Peter DeGrace
Leslie Hulet Stahl
YOURDON PRESS COMPUTING SERIES

"This book is a critique of the waterfall model, and a description of a number of alternative models. It has an early description of Sashimi and Scrum applied to software. (Sashimi draws overlapping phases like slices of fish; Scrum gives a team a goal around which it can self-organize.) This book is mostly of historical and background interest."

Review by Bill Wake - https://xp123.com/articles/review-wicked-problems-righteous-solutions/





SCRUM Development Process

Ken Schwaber

Advanced Development Methods
131 Middlesex Turnpike Burlington, MA 01803
email virman@aol.com Fax: (617) 272-0555

ABSTRACT. The stated, accepted philosophy for systems development is that the development process is a well understood approach that can be planned, estimated, and successfully completed. This has proven incorrect in practice. SCRUM assumes that the systems development process is an unpredictable, complicated process that can only be roughly described as an overall progression. SCRUM defines the systems development process as a loose set of activities that combines known, workable tools and techniques with the best that a development team can devise to build systems. Since these activiti are loose, controls to manage the process and inherent risk are used. SCRUM is an enhancement of the commonly used iterative/incremental object-oriented developmen cycle.

KEY WORDS: SCRUM SEI Capability-Maturity-Model Process Empirical

1. Introduction

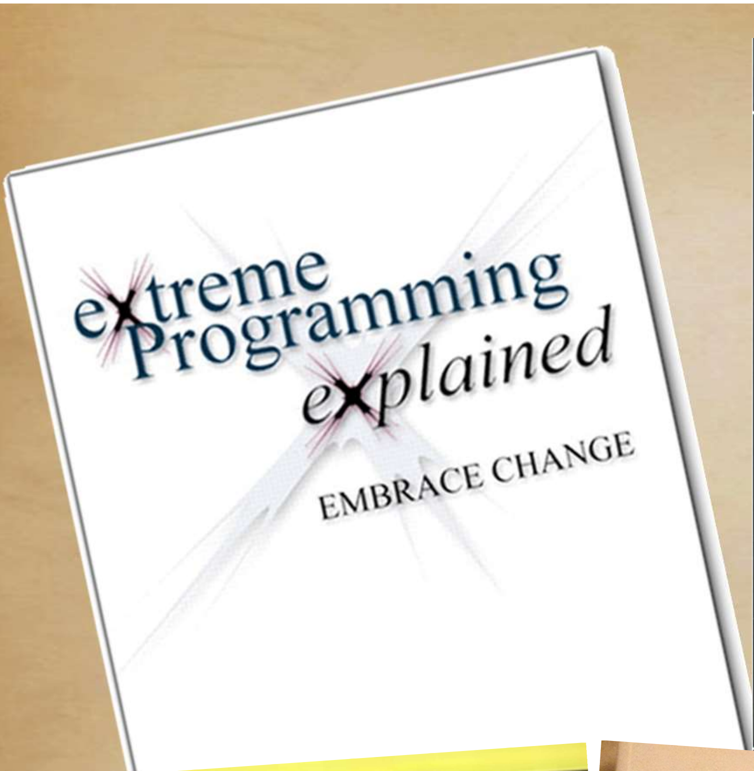
In this paper we introduce a development process, SCRUM, that treats major po systems development as a controlled black box. We relate this to complexity t show why this approach increases flexibility and produces a system that is resp both initial and additional requirements discovered during the ongoing developm

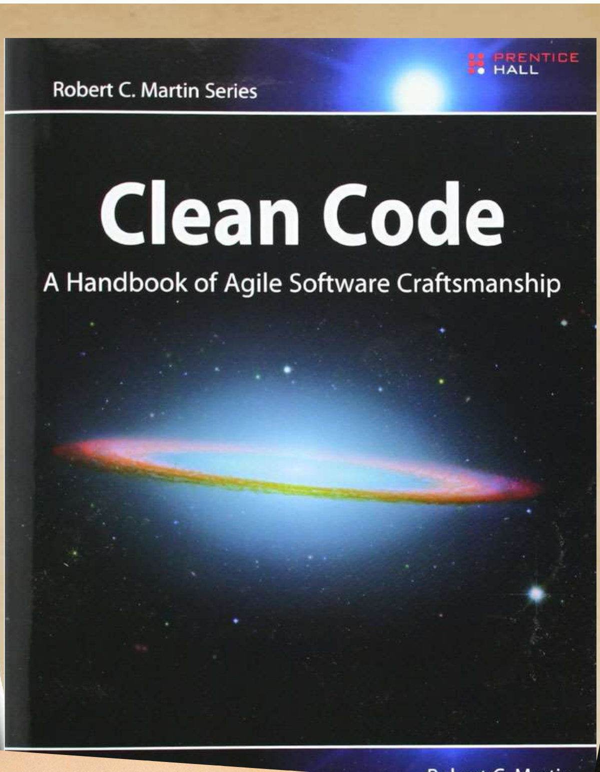
Numerous approaches to improving the systems development process have b Each has been touted as providing "significant productivity improvements." failed to produce dramatic improvements.[1] As Grady Booch noted, "We oft condition the software crisis, but frankly, a malady that has carried on this lo called normal."[2]

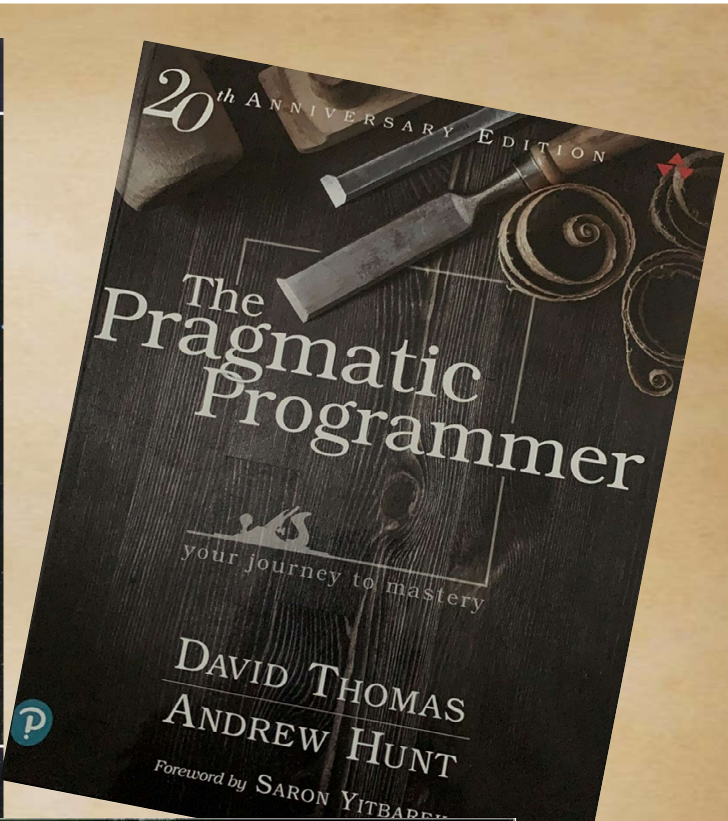
Concepts from industrial process control are applied to the field of systems in this paper. Industrial process control defines processes as either "theore defined) or "empirical" (black box). When a black box process is treated as

[1] Brooks, F.P. "No silver bullet—essence and accidents of software engineering." Computer 20:4:10-19, April 1987.
[2] Object Oriented Analysis and Design with Applications, p. 8, Grady Booch, The Benjamin/Cummings Publishing Company, Inc., 1994

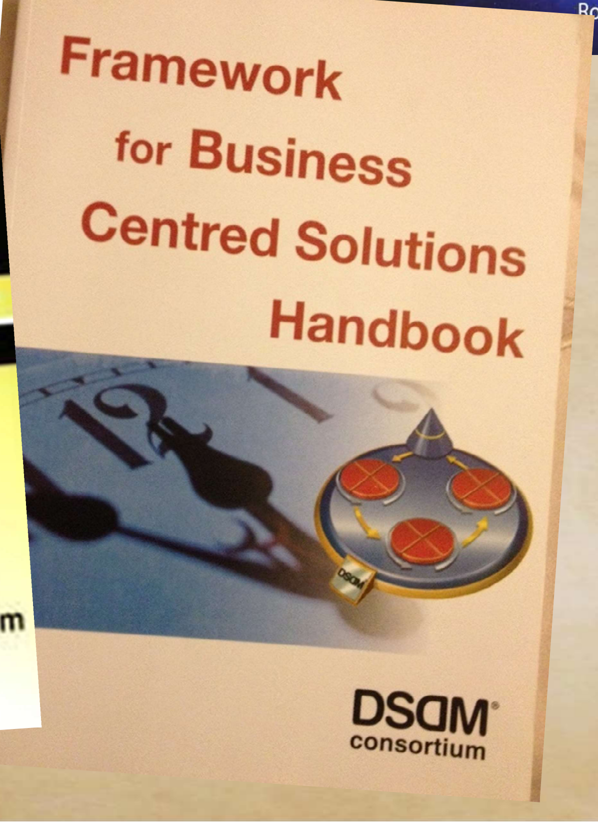
Story of Scrum ~22:00

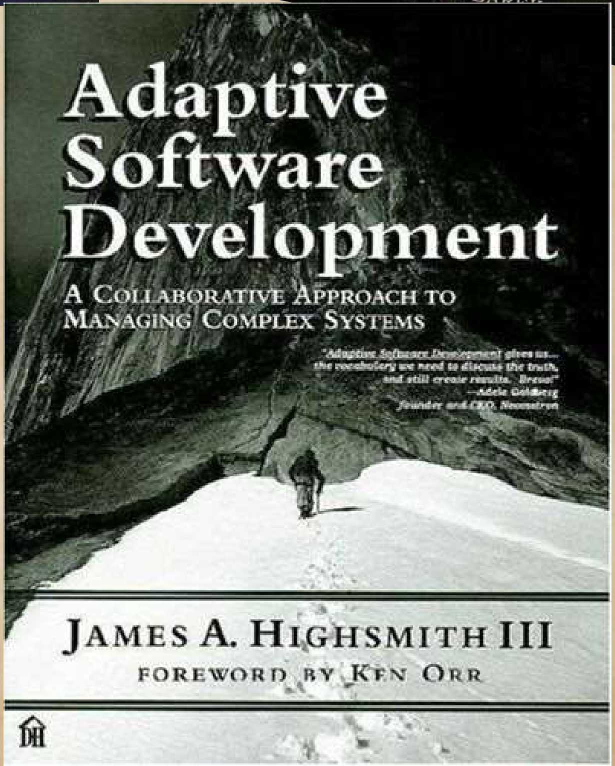

extreme Programming explained
EMBRACE CHANGE

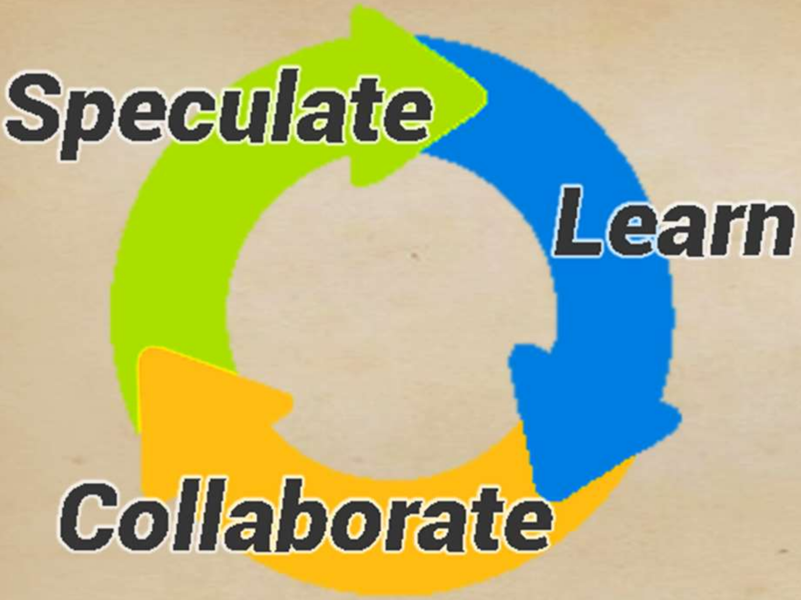

Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Robert C. Martin

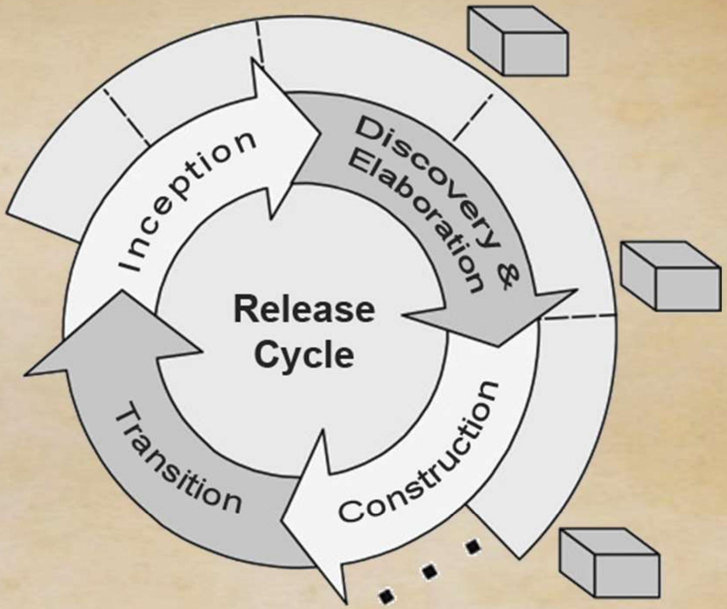

20th Anniversary Edition
The Pragmatic Programmer
your journey to mastery
David Thomas
Andrew Hunt
Foreword by Saron Yitbarek

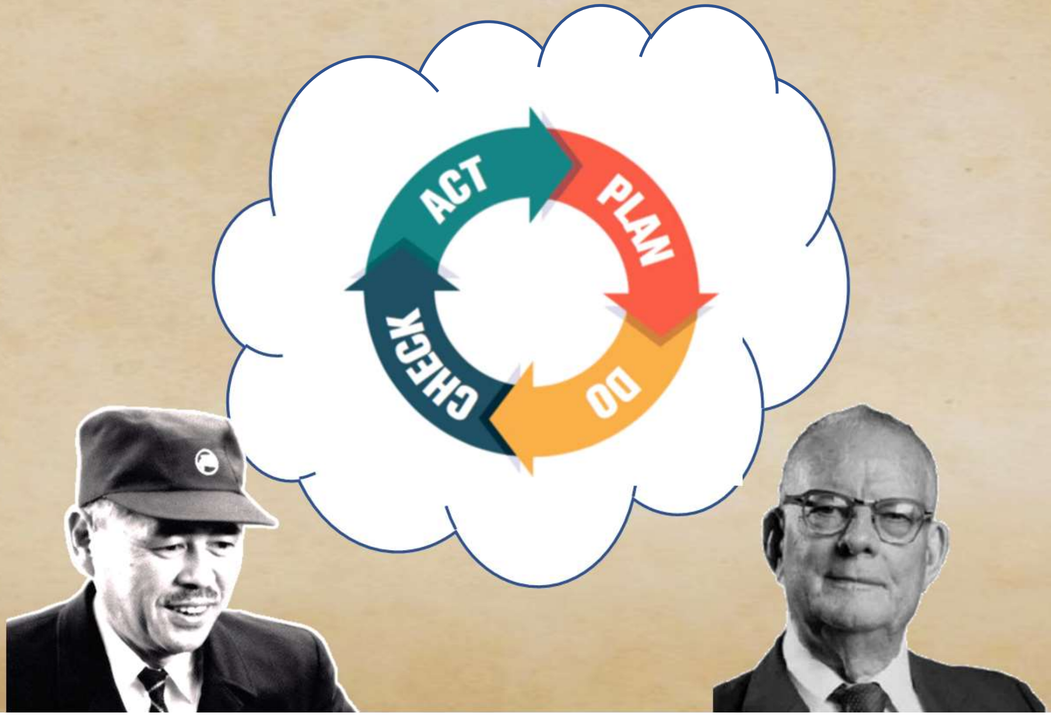
DSDM
Business Focused Development
Second Edition
Cockburn • Highsmith Series Editors
DSDM Consortium
Edited by
Jennifer Stapleton
DSDM consortium


Framework for Business Centred Solutions Handbook
DSDM
DSDM consortium


Adaptive Software Development
A Collaborative Approach to Managing Complex Systems
"Adaptive Software Development gives us... the vocabulary we need to discuss the truth, and still create results. Bravo!"
—Adele Goldberg
founder and CEO, Neometron
James A. Highsmith III
Foreword by Ken Orr
DH


Speculate
Learn
Collaborate


Inception
Discovery & Elaboration
Release Cycle
Construction
Transition
. . .


ACT
PLAN
DO
CHECK

Snowbird, Utah
February 2001

Purpose:

uncovering

We are ~~figuring out~~ better ways of developing sofware by doing it and helping others to do it.

interactions

1) individuals and ~~social networks~~ over process + tools

4 ~~2~~) responding to change over following a plan

3 ~~3~~) customer collobaration over contract negotiation

Manifesto for Agile Software Development

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

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.

Manifesto for Agile Software Development, www.agilemanifesto.org

Principles of the Agile Manifesto

http://agilemanifesto.org/principles

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

2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

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

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

5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

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

7 Working software is the primary measure of progress.

8 Agile processes promote sustainable development.
The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

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

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

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

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

How is Scrum aligned with the values and principles of *the Manifesto for Agile Software Development*?

Principles of the Agile Manifesto

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

2 Wel
deve
for t

3 Deli
coup
pref

4 Bus
toge

5 Buil
Giv
need

6 The
conv
deve

http://agilemanifesto.org/principles

7 Working software is the primary measure of progress.

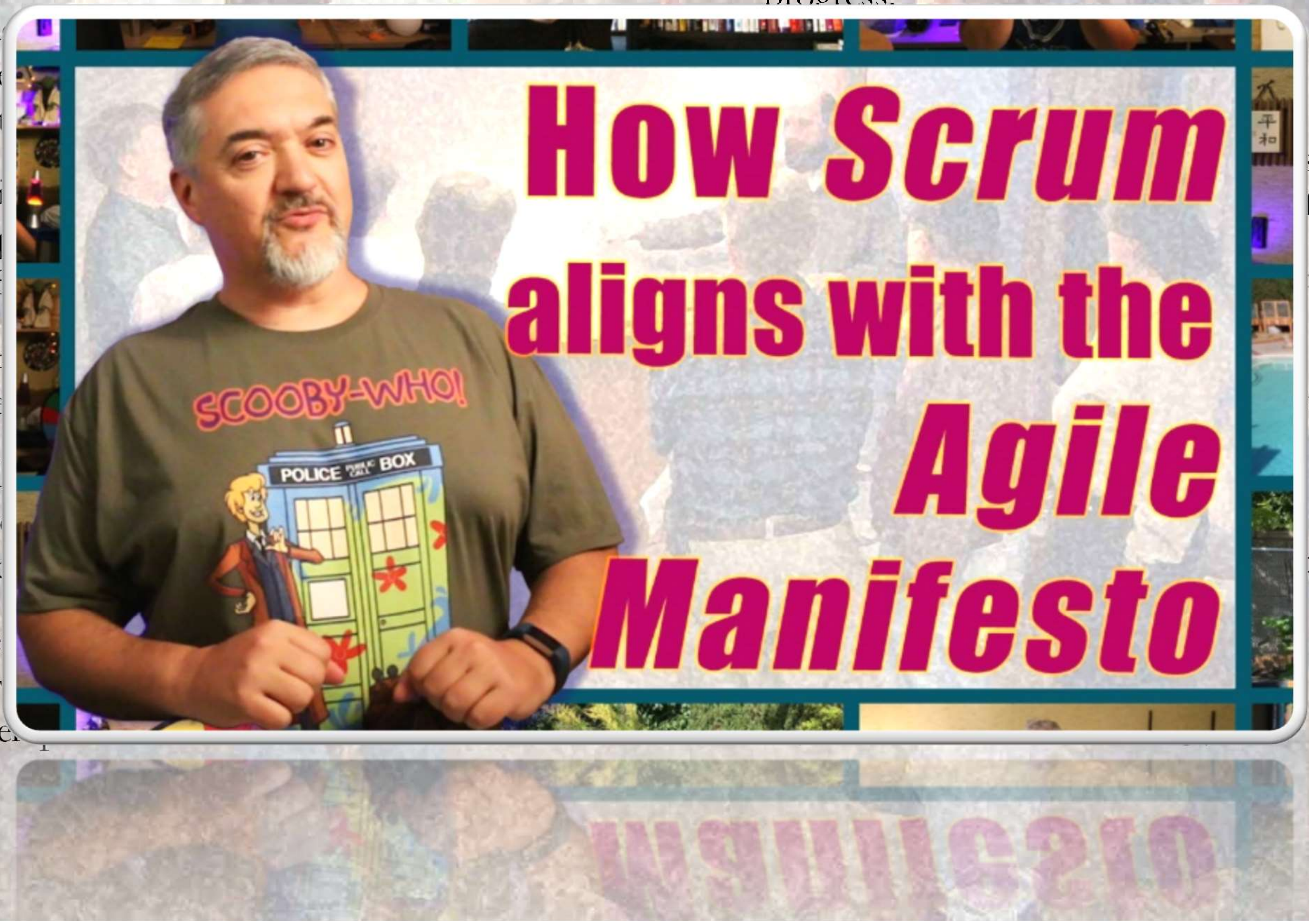
should be
finitely.

ellence

amount

and
teams.

on how
and



TO DO

- Facilitation
- Coaching
- Organizational Change

- Self-managing Teams
- XP & Kanban
- Agile Product Development
- Teaching
- Scaling Scrum / Agile Organization
- Scrum Leadership

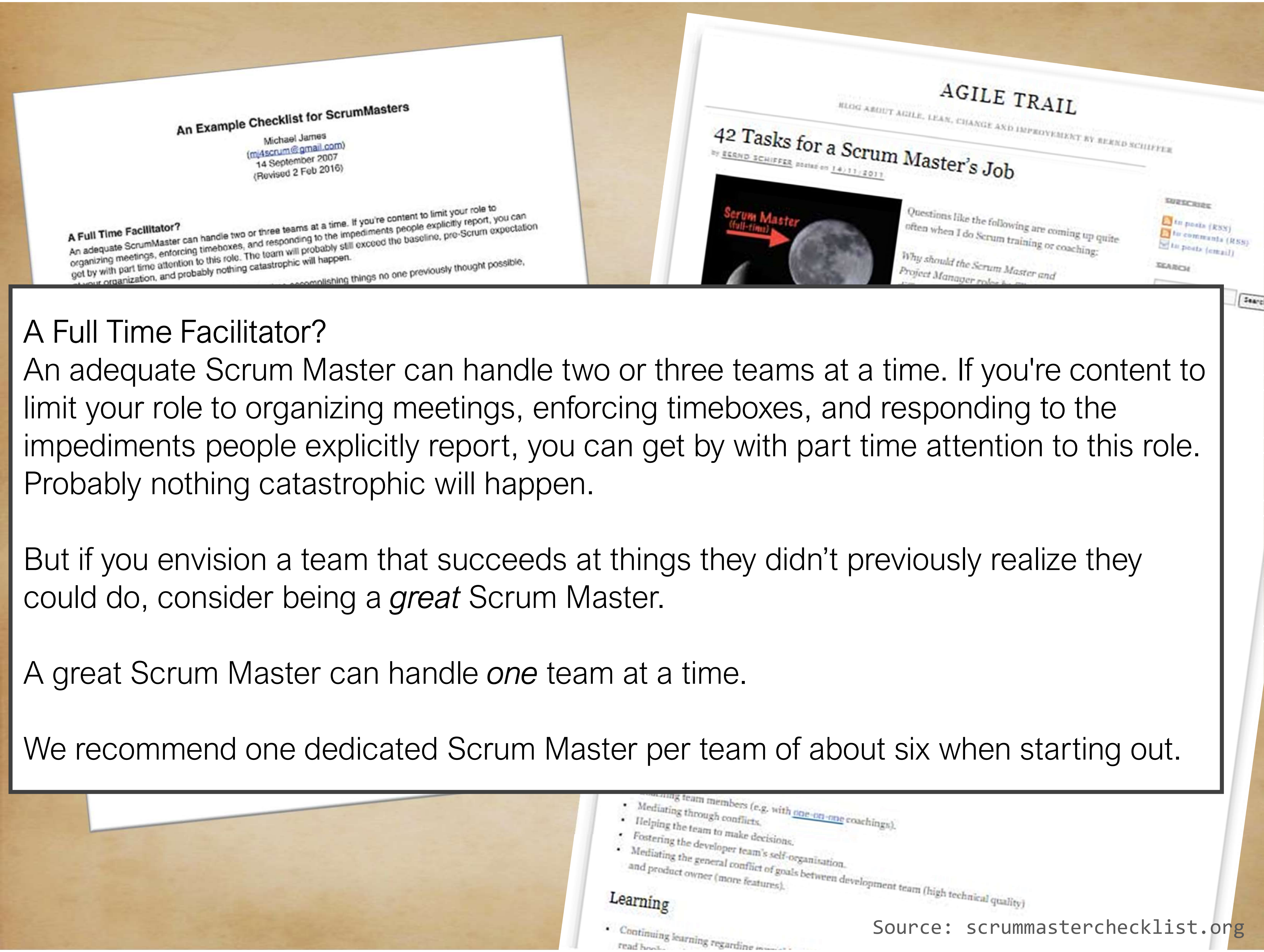
DOING

- Scrum Leadership:
  Scrum Mastery &
  Agile Leadership

DONE

- Introductions
- Why & What of Agile and Scrum

So you’re a Scrum Master …



An Example Checklist for ScrumMasters

Michael James
(mj4scrum@gmail.com)
14 September 2007
(Revised 2 Feb 2016)

A Full Time Facilitator?

An adequate ScrumMaster can handle two or three teams at a time. If you're content to limit your role to organizing meetings, enforcing timeboxes, and responding to the impediments people explicitly report, you can get by with part time attention to this role. The team will probably still exceed the baseline, pre-Scrum expectation at your organization, and probably nothing catastrophic will happen.

accomplishing things no one previously thought possible,

AGILE TRAIL

BLOG ABOUT AGILE, LEAN, CHANGE AND IMPROVEMENT BY BERND SCHIFFER

42 Tasks for a Scrum Master's Job

by BERND SCHIFFER posted on 14.11.2011

Questions like the following are coming up quite often when I do Scrum training or coaching:

Why should the Scrum Master and Project Manager roles be

SUBSCRIBE
to posts (RSS)
to comments (RSS)
to posts (email)

SEARCH

Search

team members (e.g. with one-on-one coachings).
- Mediating through conflicts.
- Helping the team to make decisions.
- Fostering the developer team's self-organisation.
- Mediating the general conflict of goals between development team (high technical quality) and product owner (more features).

Learning

- Continuing learning regarding

A Full Time Facilitator?

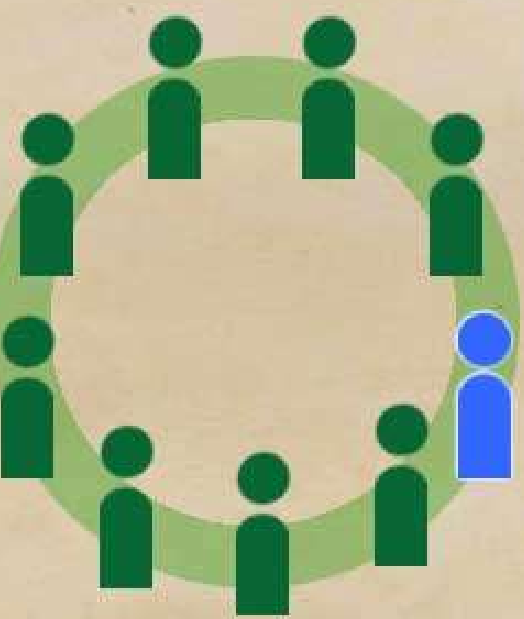
An adequate Scrum Master can handle two or three teams at a time. If you're content to limit your role to organizing meetings, enforcing timeboxes, and responding to the impediments people explicitly report, you can get by with part time attention to this role. Probably nothing catastrophic will happen.

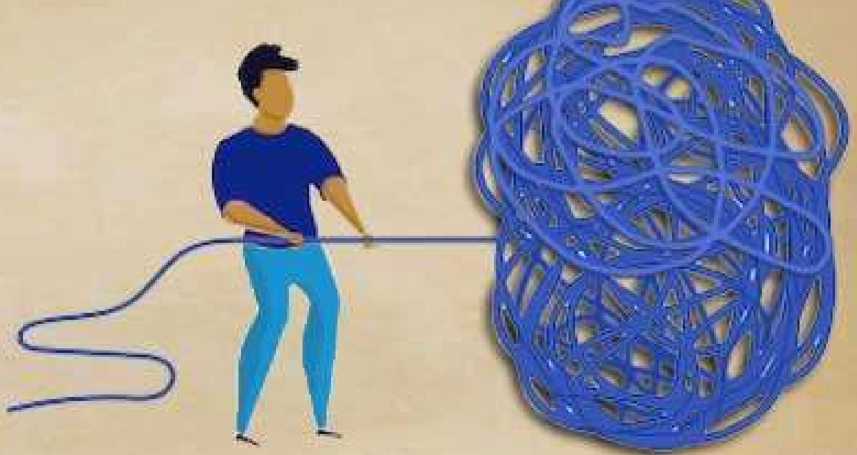
But if you envision a team that succeeds at things they didn't previously realize they could do, consider being a *great* Scrum Master.

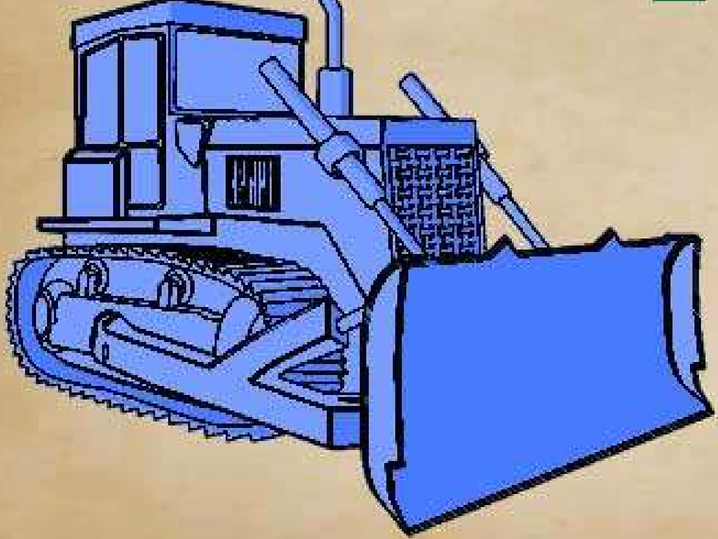
A great Scrum Master can handle *one* team at a time.

We recommend one dedicated Scrum Master per team of about six when starting out.

Source: scrummasterchecklist.org







Serving the Organization

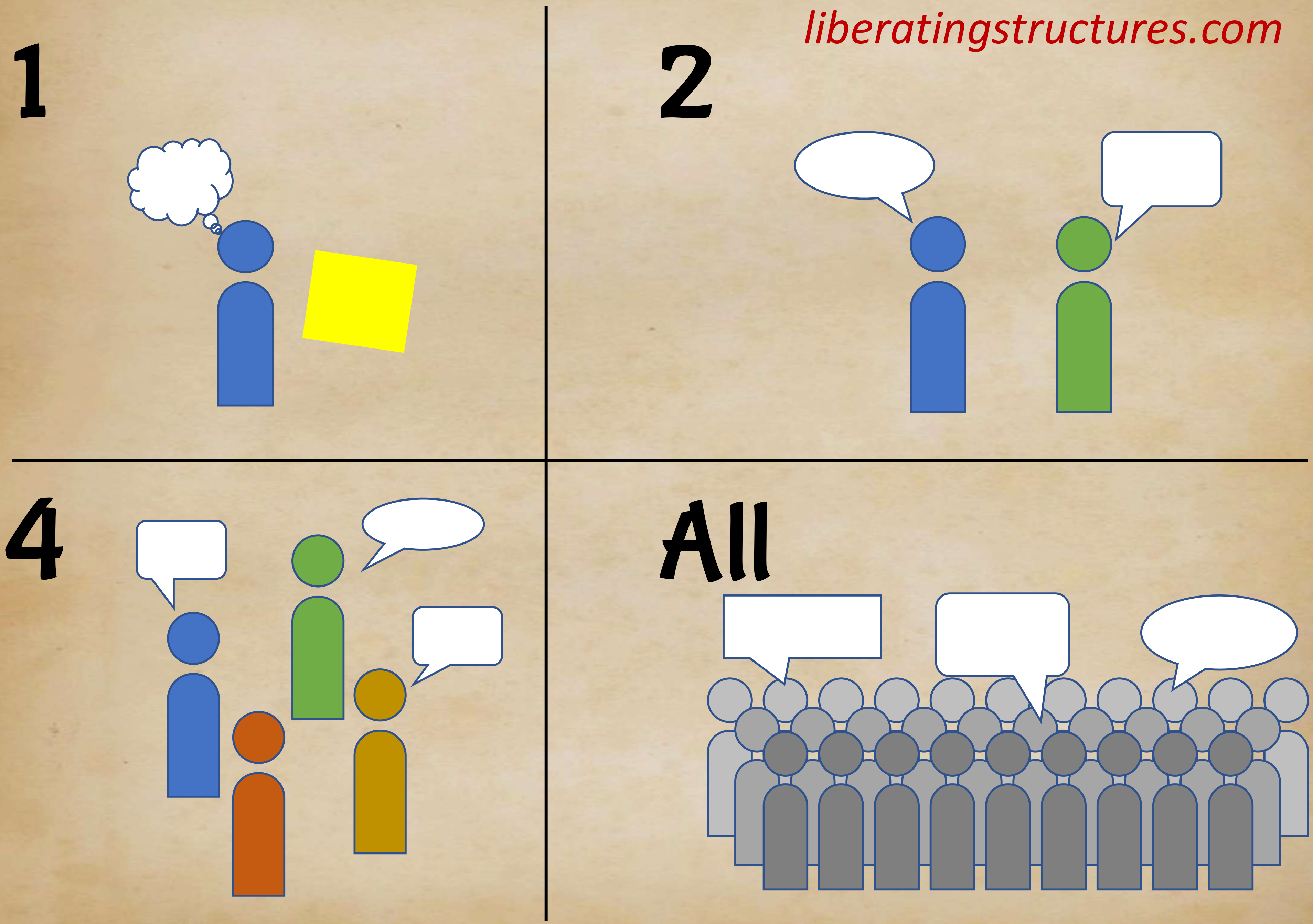
- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

Source: Scrum Guide

Scrum Master

Time Box: 6 Min.

What are the personality traits of an excellent Scrum Master?



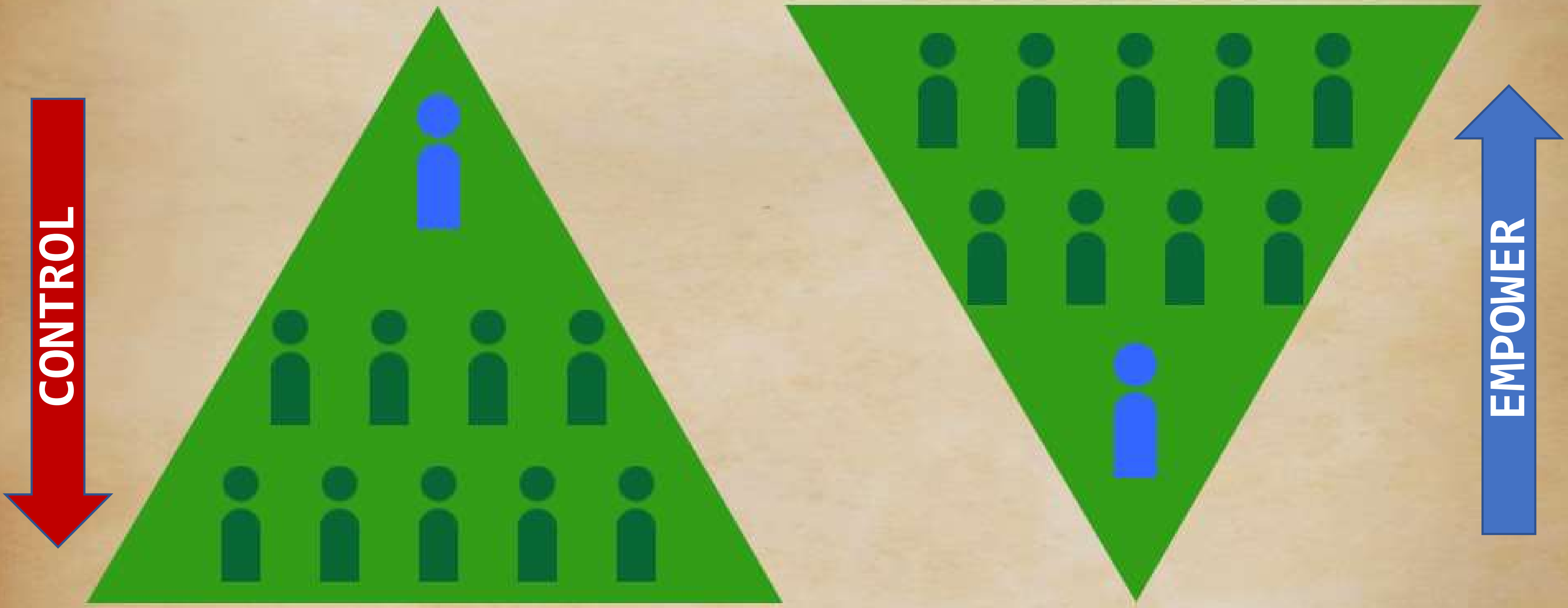
liberatingstructures.com

How would you describe Effective Leadership?

What are some key attributes
of an effective leader?

Traditional Leadership
(hierarchical teams)

True Leadership
(Agile teams)

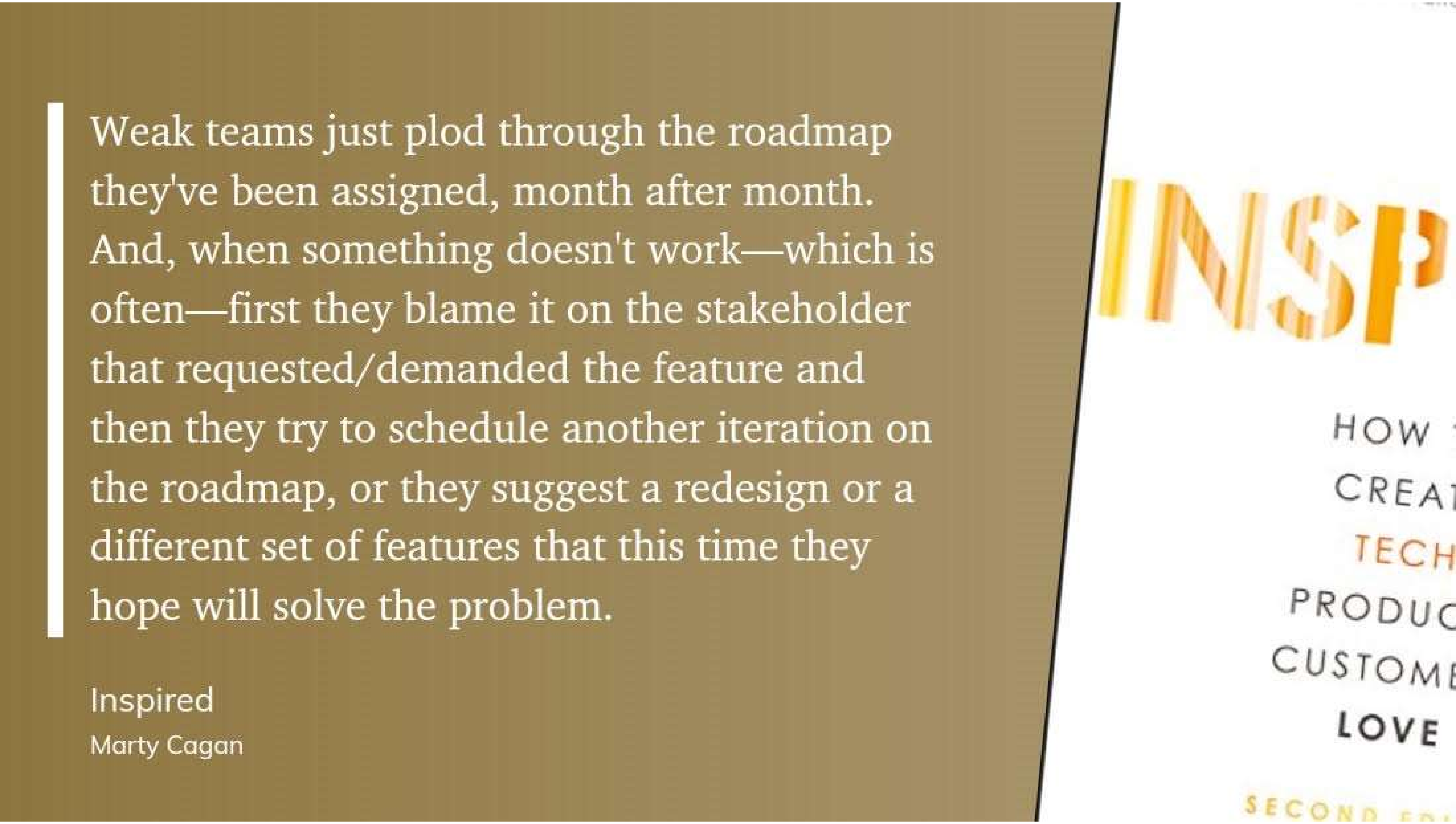


CONTROL

EMPOWER

Dysfunctional Players Theater

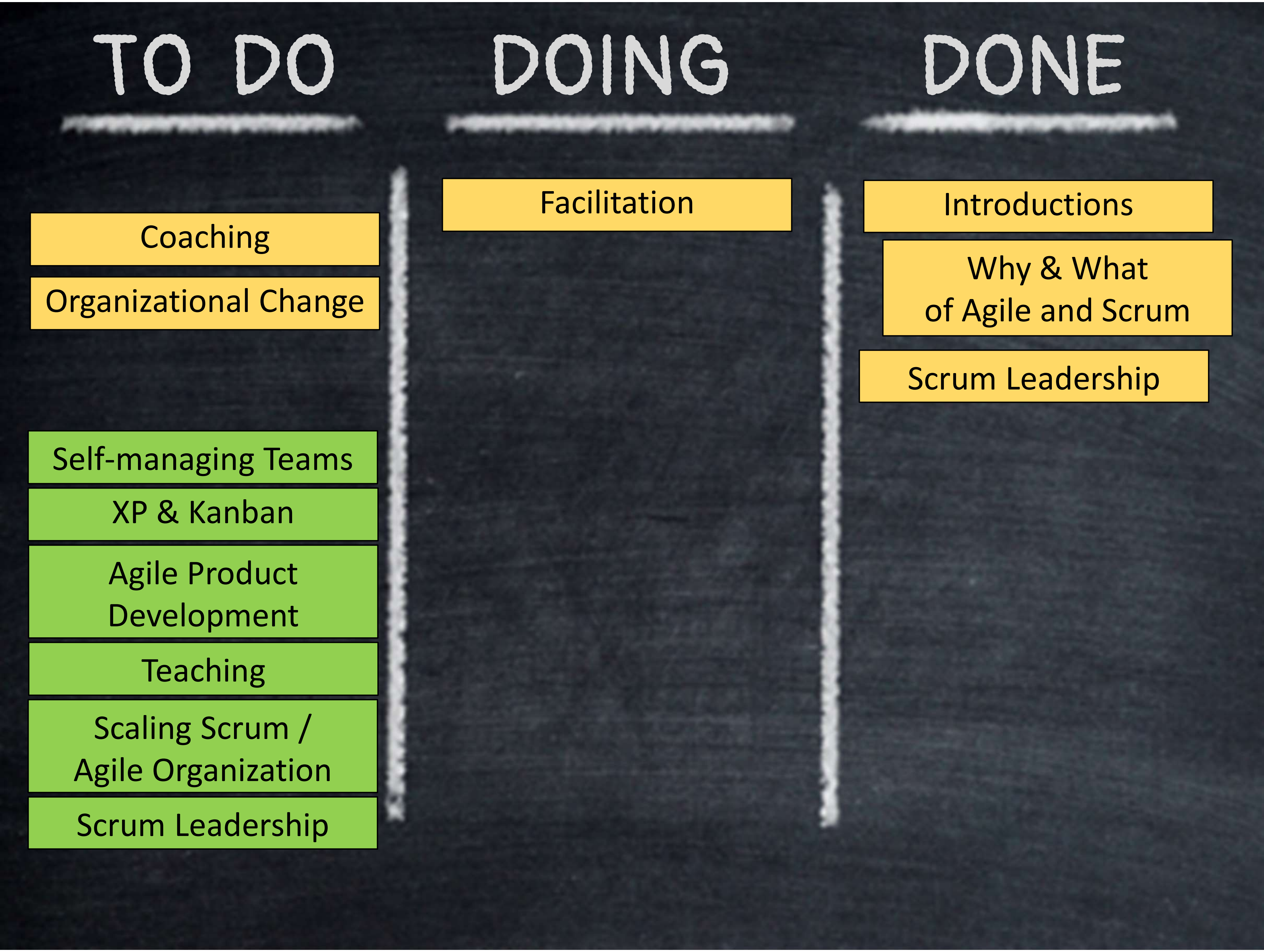
- ☑ form groups of 2-4
- ☑ create a scenario to demonstrate a common dysfunction
- ☑ demonstrate a leadership attribute to remove the dysfunction
- ☑ after 8 minutes, return here
- □ SHOW TIME



> Weak teams just plod through the roadmap they've been assigned, month after month. And, when something doesn't work—which is often—first they blame it on the stakeholder that requested/demanded the feature and then they try to schedule another iteration on the roadmap, or they suggest a redesign or a different set of features that this time they hope will solve the problem.

Inspired

Marty Cagan



TO DO

- Coaching
- Organizational Change
- Self-managing Teams
- XP & Kanban
- Agile Product Development
- Teaching
- Scaling Scrum / Agile Organization
- Scrum Leadership

DOING

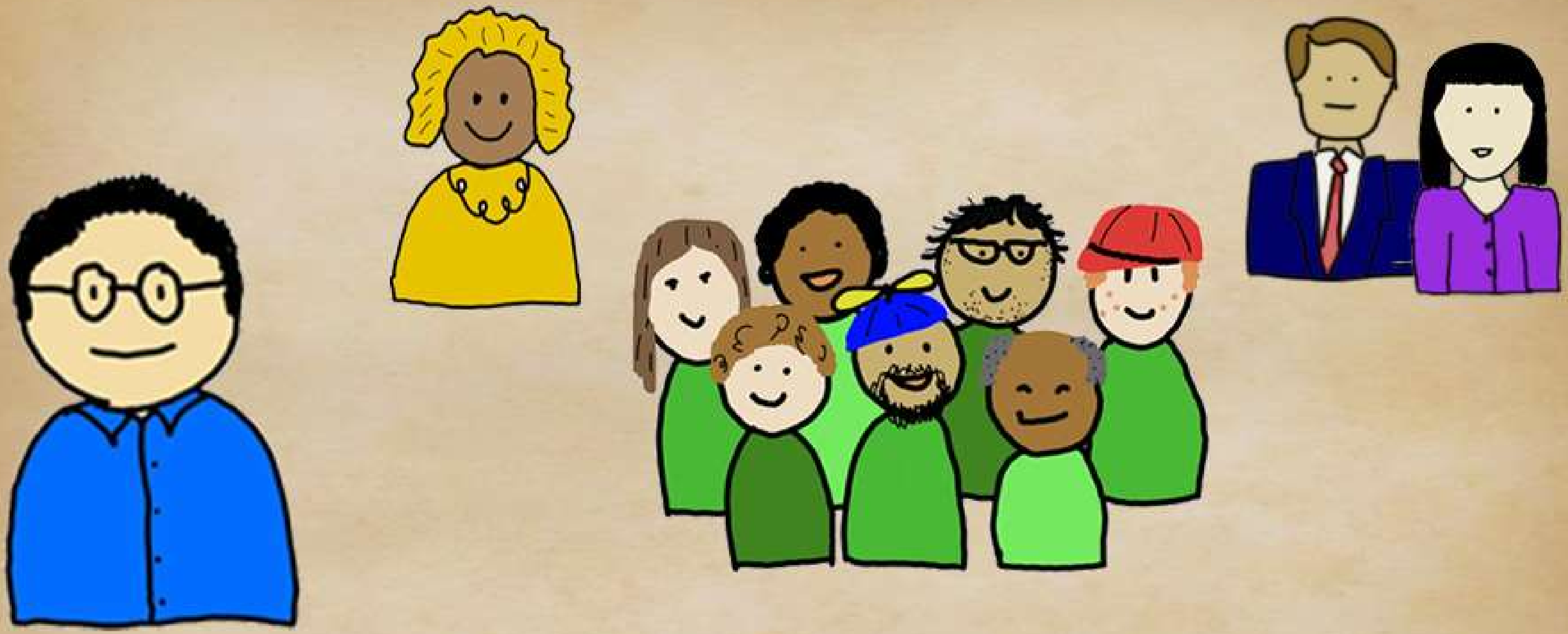
- Facilitation

DONE

- Introductions
- Why & What of Agile and Scrum
- Scrum Leadership

ScrumMaster as Facilitator

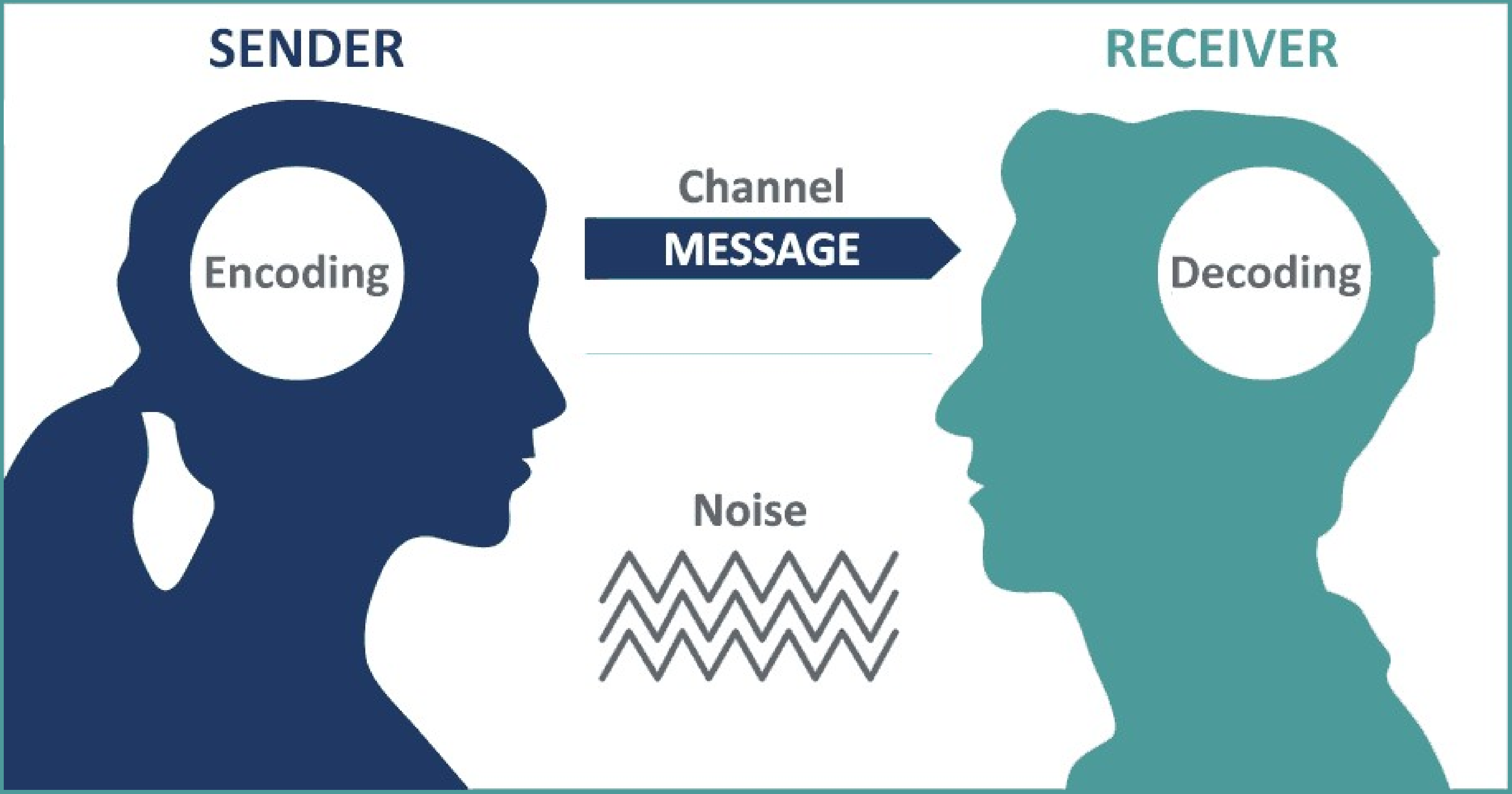
- Facilitation: “make easier; help bring about”
  - Clear Communication and Teamwork
- ScrumMaster facilitates:
  - Scrum events as requested or as needed
  - other conversations and meetings as necessary
- Facilitating takes preparation and focus
  - Not merely writing on whiteboard / taking notes
  - Requires paying attention to many *non-verbal cues* (e.g. facial expressions and body language)
  - Requires *actively listening (facilitative listening)*



- Neutral Facilitator - not advocating for any one solution
- Active Facilitator - not tuning out, typing, etc.
- Help the Scrum Team (Product Owner and Developers) and the Organization reach their goals and continue to improve the delivery of business value with each Sprint

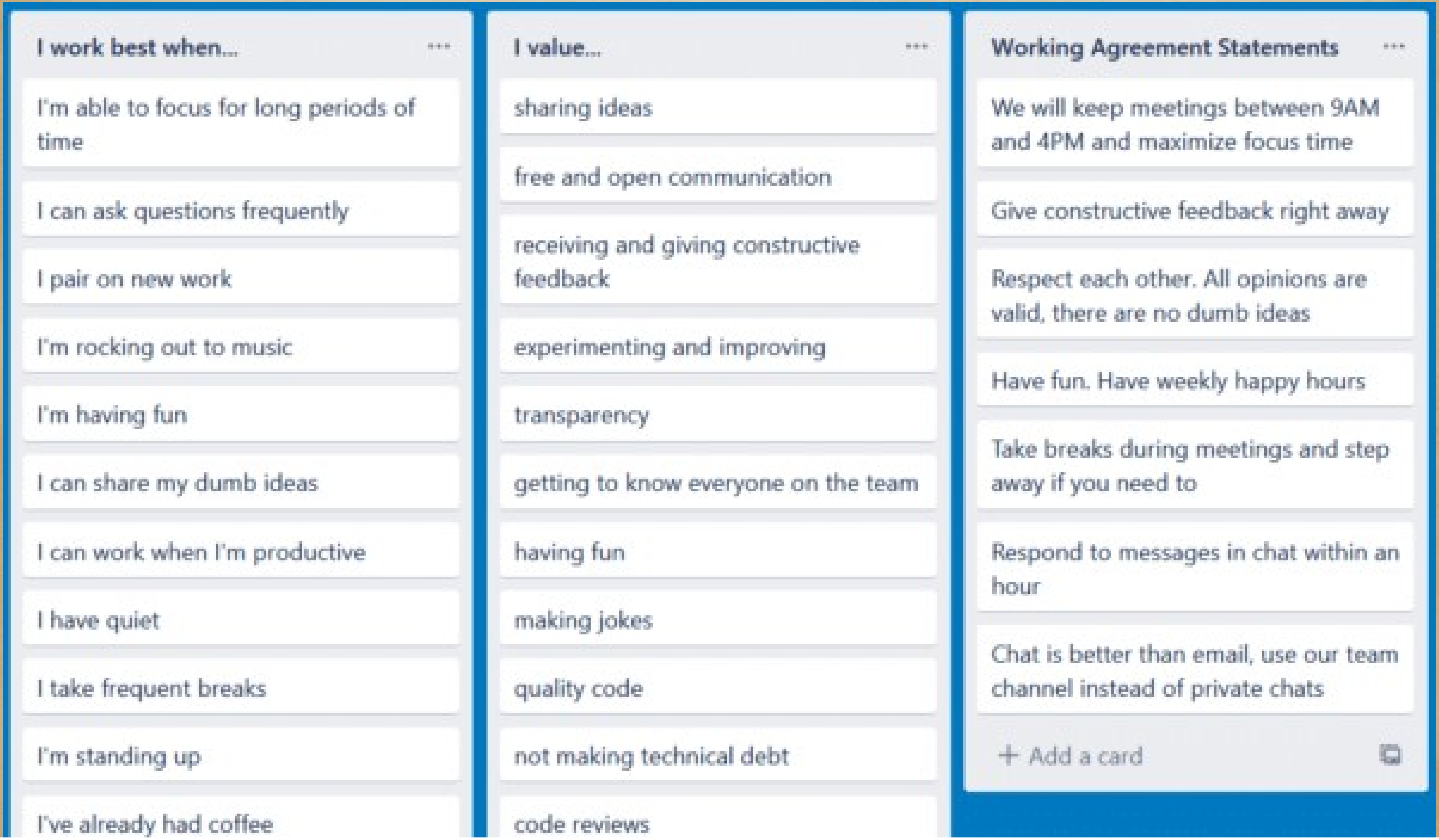
Sender, Receiver, Noise

COMMUNICATION MODEL



Working Agreements foster Clear Communication and Teamwork

Source: https://www.ruralsourcing.com/blogs/five-steps-for-improving-team-communication-with-working-agreements/



I work best when...

- I'm able to focus for long periods of time
- I can ask questions frequently
- I pair on new work
- I'm rocking out to music
- I'm having fun
- I can share my dumb ideas
- I can work when I'm productive
- I have quiet
- I take frequent breaks
- I'm standing up
- I've already had coffee

I value...

- sharing ideas
- free and open communication
- receiving and giving constructive feedback
- experimenting and improving
- transparency
- getting to know everyone on the team
- having fun
- making jokes
- quality code
- not making technical debt
- code reviews

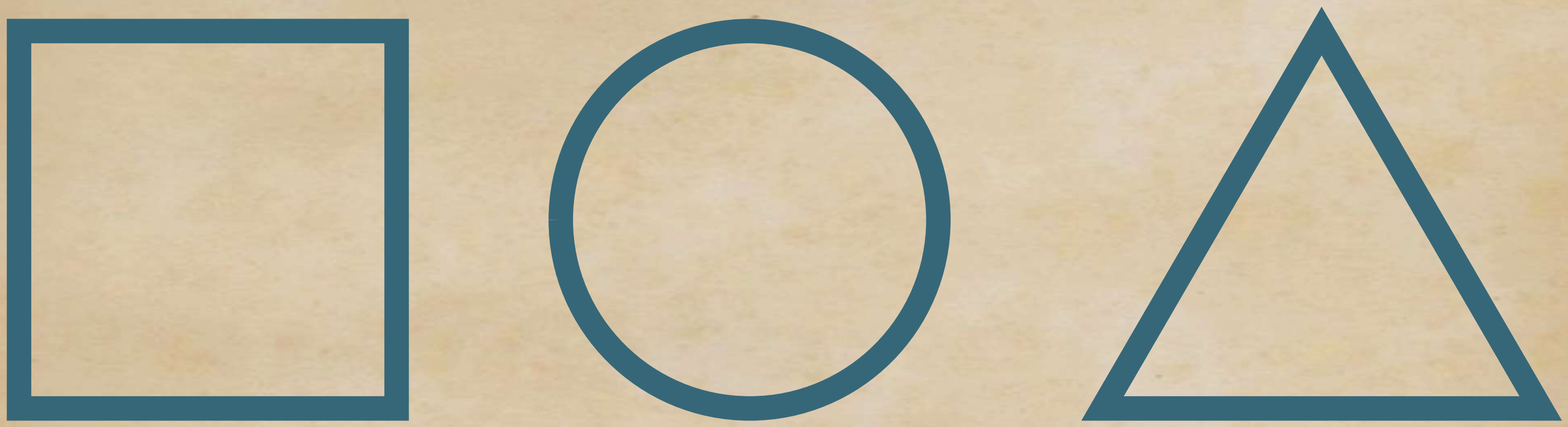
Working Agreement Statements

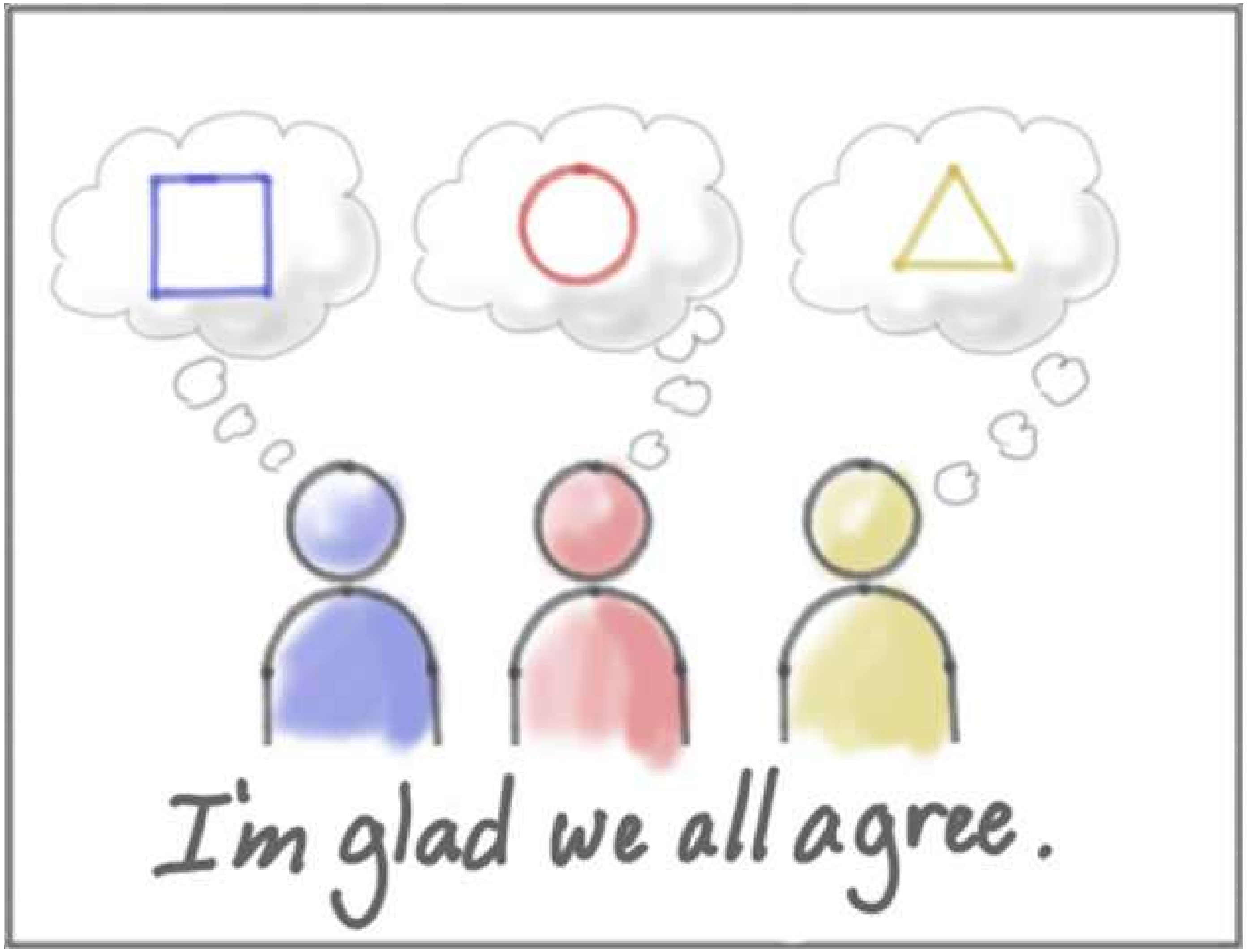
- We will keep meetings between 9AM and 4PM and maximize focus time
- Give constructive feedback right away
- Respect each other. All opinions are valid, there are no dumb ideas
- Have fun. Have weekly happy hours
- Take breaks during meetings and step away if you need to
- Respond to messages in chat within an hour
- Chat is better than email, use our team channel instead of private chats

How do we resolve obstacles to Clear Communication?

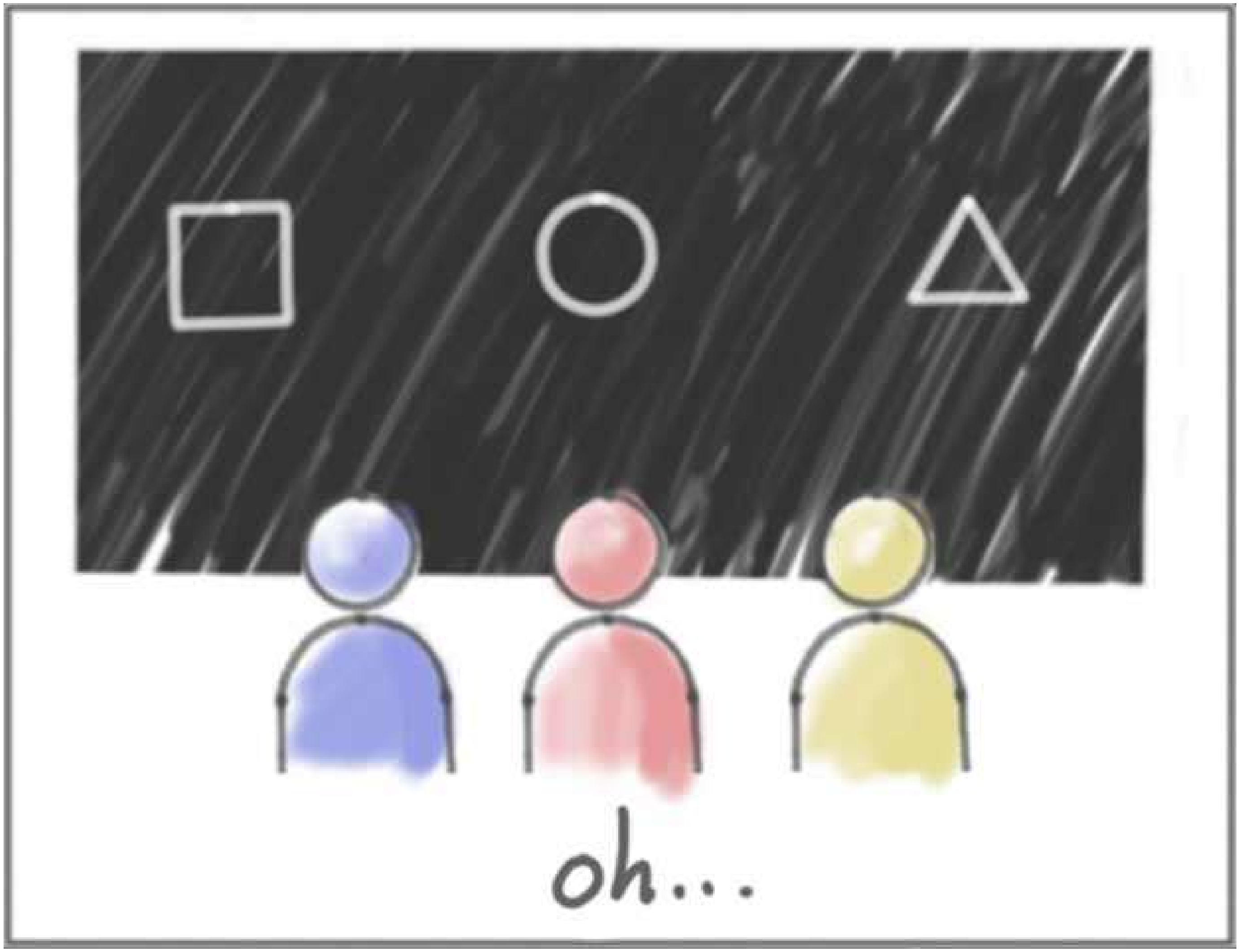
Verbal	Jargon, Slang, Regional language Disorganized messages, Ambiguity Information overload, Tone
Cultural	Attitudinal differences No shared experiences Demographic differences
External	Noise Technology Physiological (health, vision, hearing)

How do we integrate from Multiple Frames of Reference?



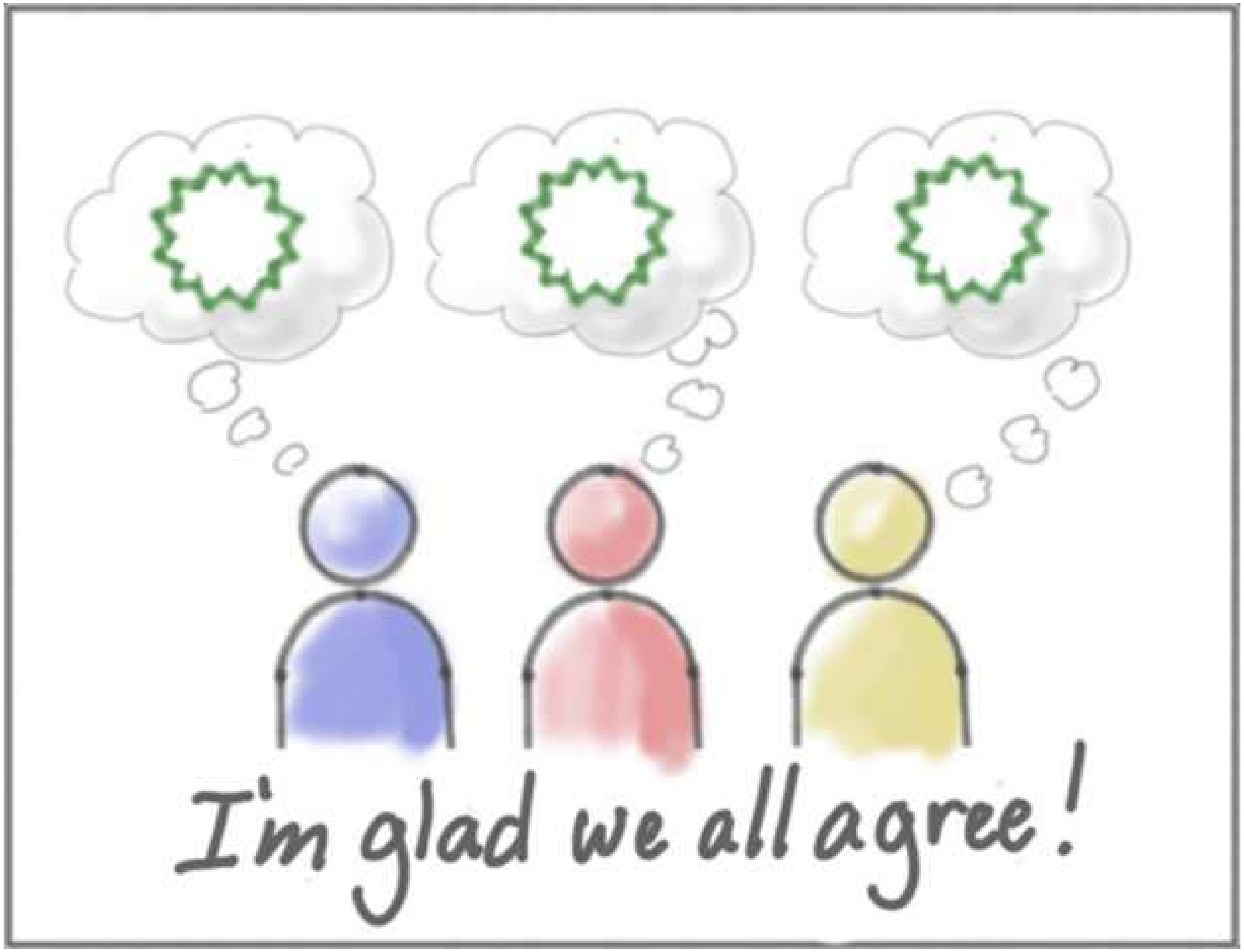


With a shallow discussion,
we each might take away something different.



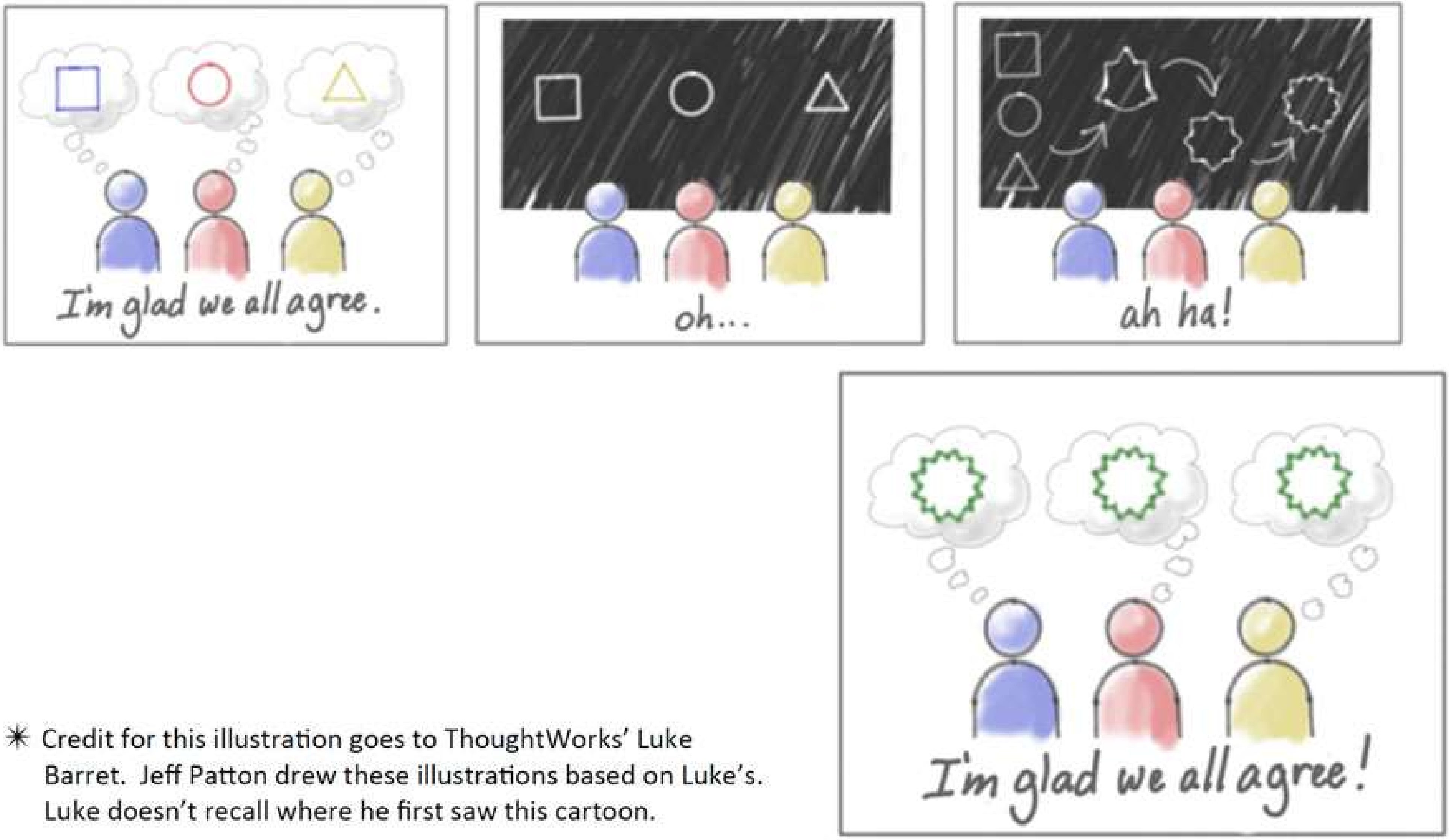
When we externalize our thinking
with words and pictures, we detect differences.

When we combine and refine,
we arrive at something better.



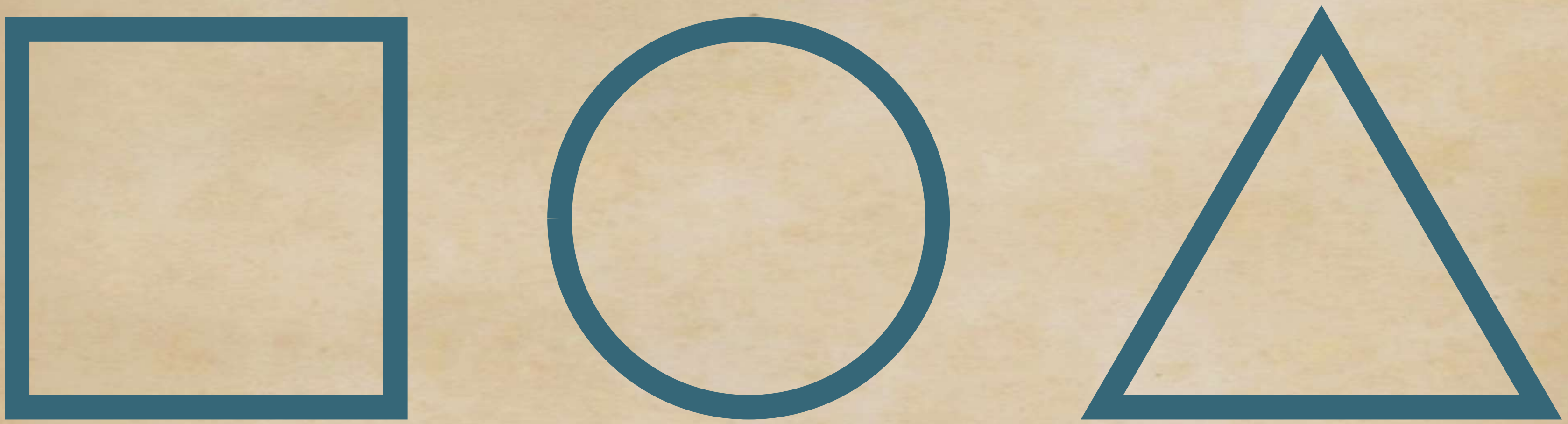
Afterwards, when we say the same thing, we actually mean it.

Shared understanding and alignment are the objectives of collaborative work.



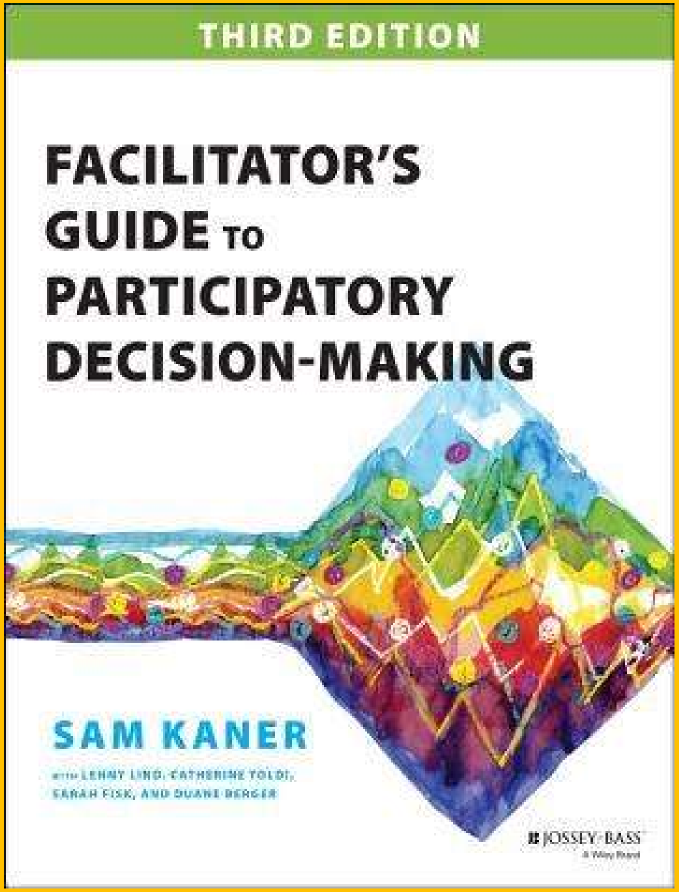
✳ Credit for this illustration goes to ThoughtWorks' Luke Barret. Jeff Patton drew these illustrations based on Luke's. Luke doesn't recall where he first saw this cartoon.

What challenges exist in integrating multiple perspectives?



Facilitation Reference: Sam Kaner

- Active listening is also referred to as ***facilitative listening***
- The facilitator’s goal is to get participants to listen to each other and collaborate to achieve the session’s goal
- It is important for the facilitator to remind participants of their roles and to establish any ground rules or agreements necessary to achieve the goal of the session



THIRD EDITION

FACILITATOR’S GUIDE TO PARTICIPATORY DECISION-MAKING

SAM KANER

WITH LENNY LIND, CATHERINE TOLDI, SARAH FISK, AND DUANE BERGER

JOSSEY-BASS
A Wiley Brand

RECOMMENDED READING

Facilitative Listening

Kaner offers many Facilitative Listening techniques:

- ***<u>Paraphrasing</u>***: supporting a speaker when thinking out loud and allowing them to review the contribution. An example would begin by saying something like “Let me see if I am understanding you” – then “play back” what was said – close with “Is that right”?

- ***<u>Mirroring</u>***: This can help establish or re-establish your neutrality as a facilitator and is just restating the speaker’s words verbatim.

- ***<u>Making Space</u>*** for a Quiet Person: Offers people who are more introverted, quiet or who need to take information in before speaking the chance to do so. Calling on them by name, ask if they have anything to add or if they would like to say something.

Facilitative Listening

- ***Drawing People Out***: When someone is having trouble expressing their idea, is being vague or confusing, helping to gain clarity. Try asking “Can you say more about that?” or “Can you give me an example?”

- ***Using the Clock***: To provide a subtle clue to quieter participants that if they want to contribute the time is now. “There are 5 minutes left in our timebox. Does anyone have anything else to add or to bring up?”

- ***Validating***: Legitimizes and accepts speaker’s opinion or feeling without agreeing that the opinion is correct. First, paraphrase, then assess if the speaker needs support (and offer support if needed). “I heard you say ___. What’s next?”

The Breakout Bunch

Have you used any of the Facilitative (Active) Listening techniques referenced from Kaner?

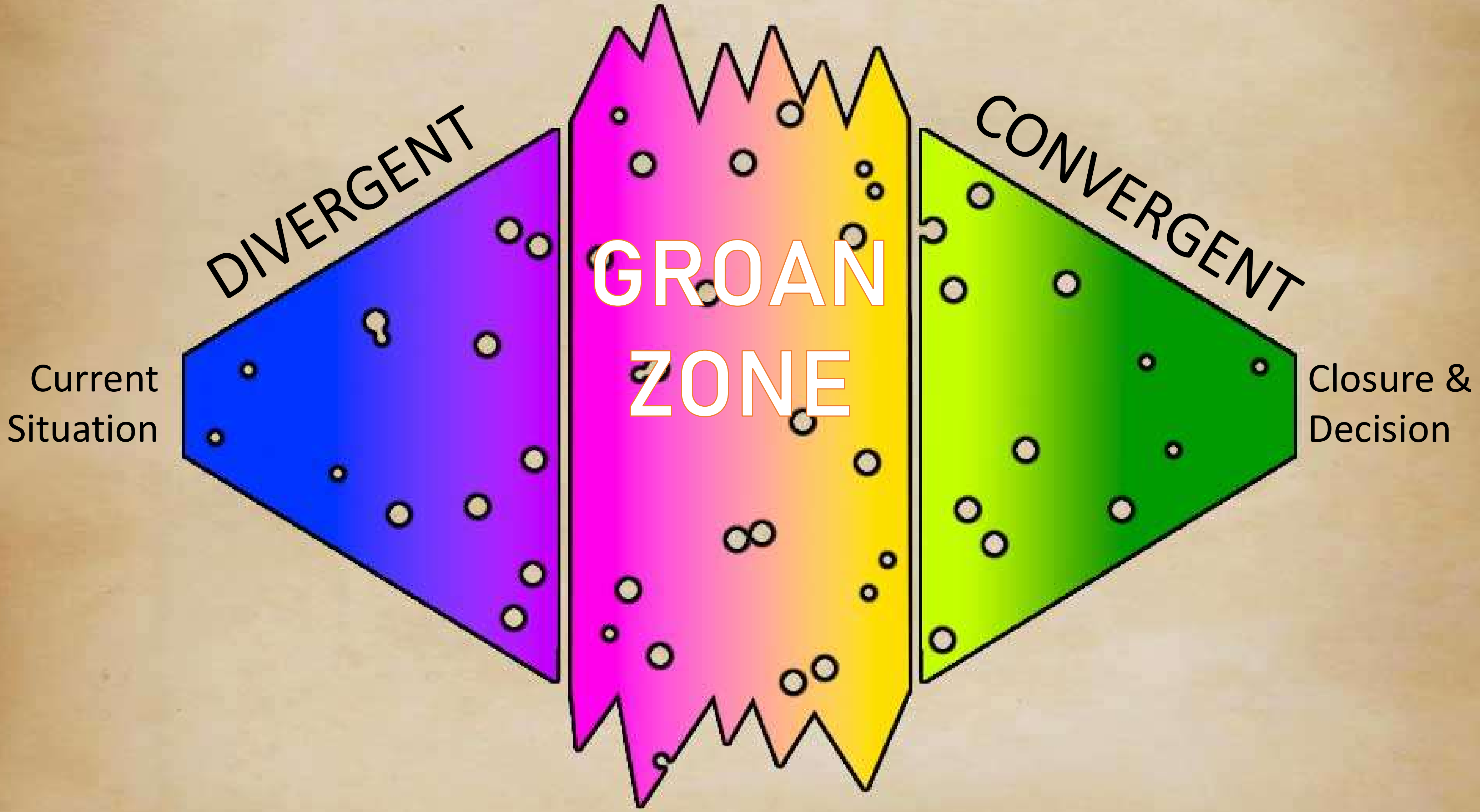
- If so, which ones?
- Which Scrum Events or what types of meetings and activities have you used these in?
- If you have not had an opportunity to use these yet, which ones do you plan on applying?

Assignment #1:

When you get back to a work environment, apply at least two facilitative listening techniques for effective meetings/events.

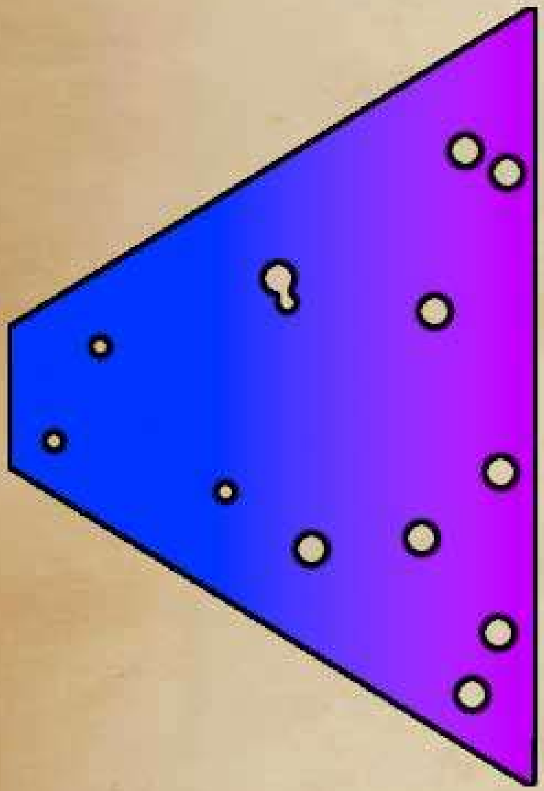
Document these (and any observations and learnings) in your personal journal.

Participatory Decision Making



The Diamond of Participatory Decision-Making (Sam Kaner, et al. 2001)

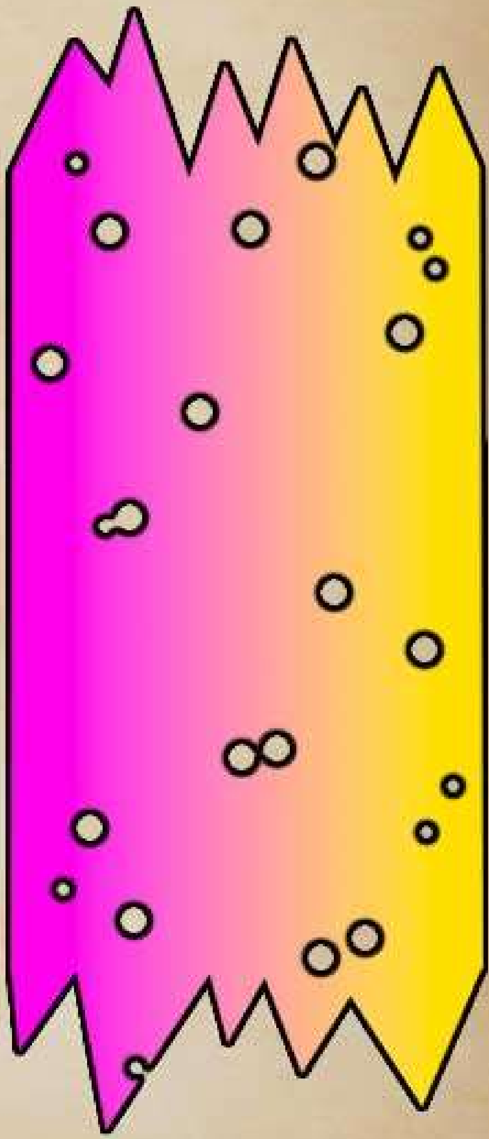
Divergent Zone

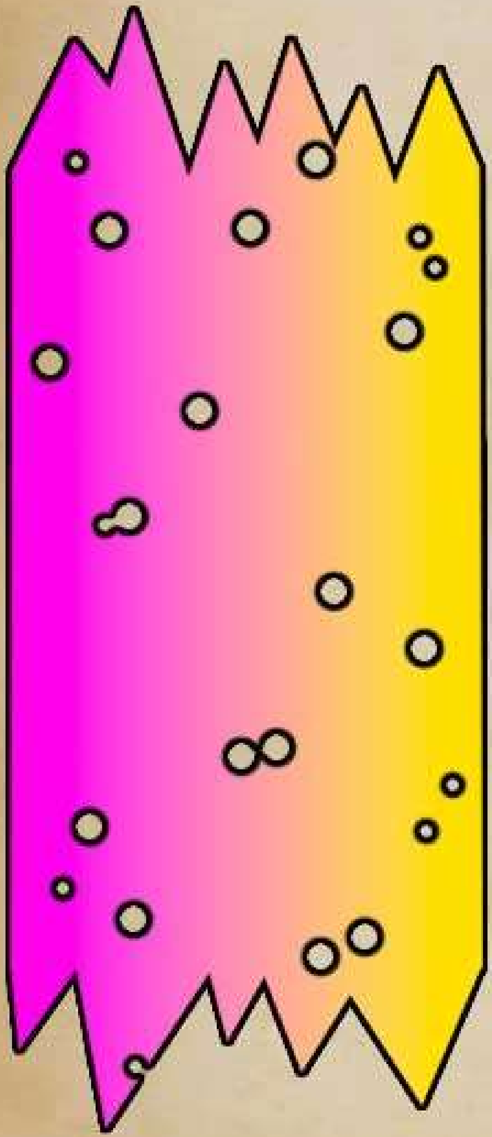


- Ideas are shared and explored
- Diversity and possibilities are celebrated
- Openness to any ideas
- Ideas are generated in a free-flowing manner and are building upon one another
- Ideas are generating in an “emergent” way

The Groan Zone

- People struggle with where to take ideas next and can feel stuck
- Effort is needed to understand differing perspectives; everyone has their own frame of reference
- Participants can become impatient or get frustrated as communication breaks down
- Participants feel uncomfortable as they struggle with how to gain clarity

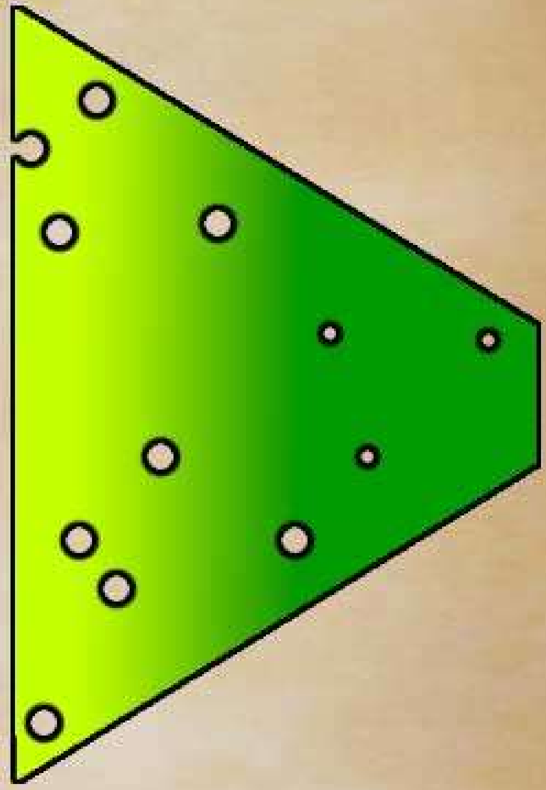




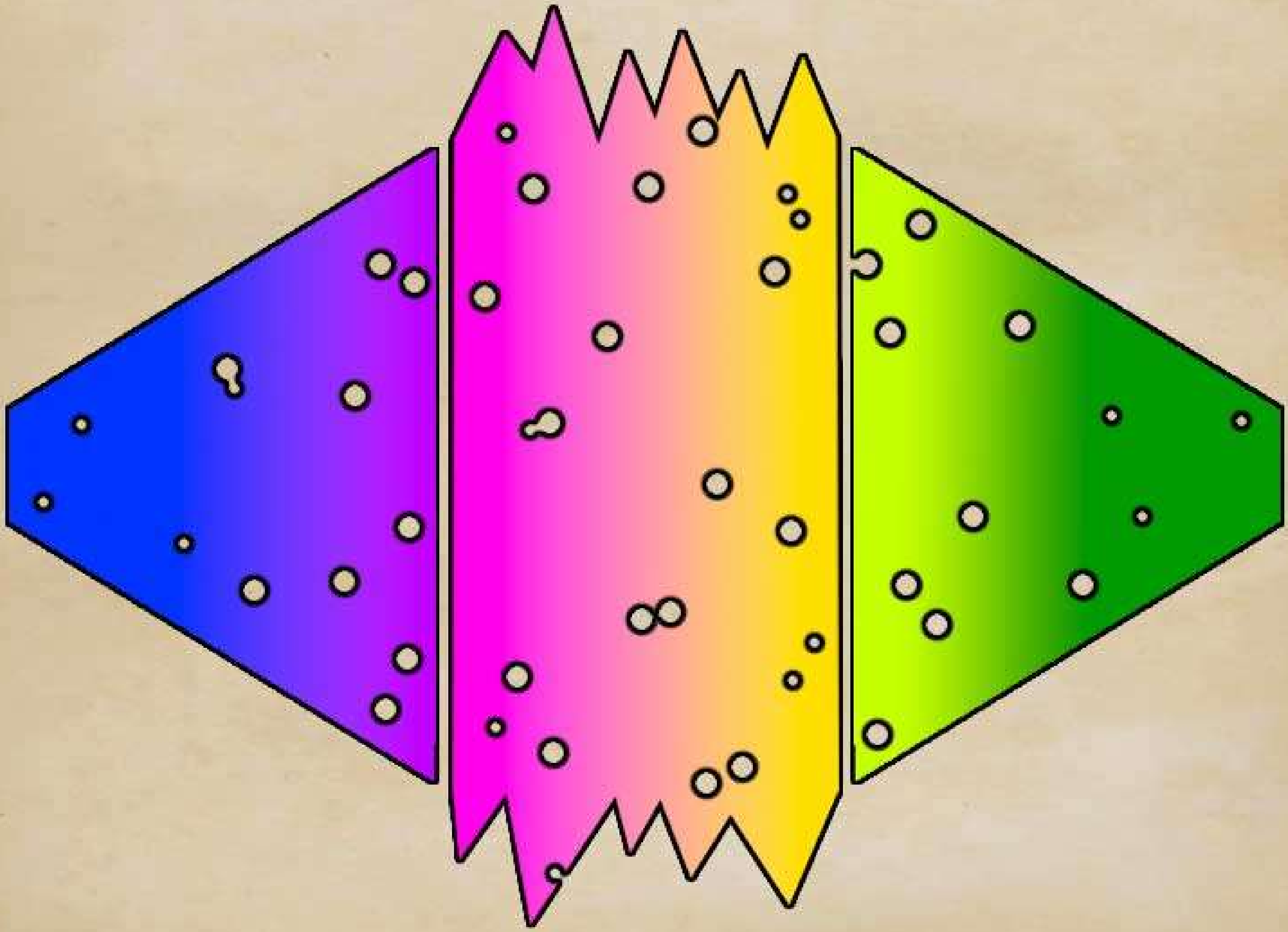
- Groups can remain stuck in the ***Groan Zone*** if there is no shared understanding between members
- There can be organizational cultural differences or heritage-based cultural differences
- Language can pose communication challenges – even if everyone is using the same one
- Competition or jockeying for “power” is another challenge that may be experienced in this zone

Convergent Zone

- Participants can get excited as they gain clarity and move towards agreement or consensus
- Different ideas come together
- Decisions are made about the best solutions or ways to proceed
- Ideas are refined and might be sorted into categories
- Key points are summarized as agreement is reached



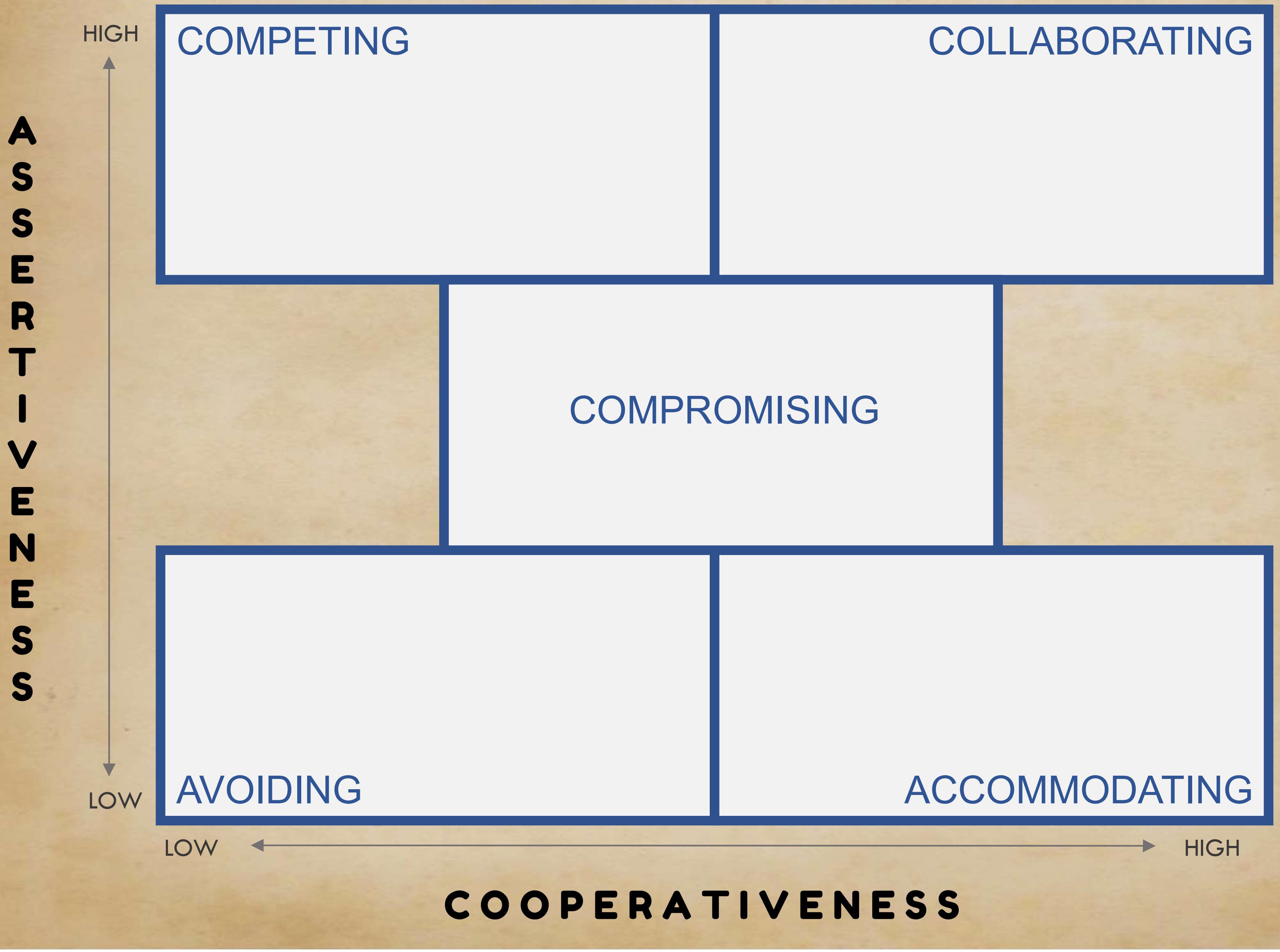
Where is the team?



What are some indicators of where you're at?

Recognize Conflict with TKI

(Thomas-Kilmann conflict mode instrument)



Recognizing your Conflict Mode

Think about the last time you were in a conflict.

- It doesn’t have to be from work; any conflict will do.
- What was your response mode?
  - Were you driven by a desire to assert your own needs or satisfy another’s concerns?

Go through the same process for a few more conflict scenarios. Try to find your native conflict response mode (natural tendency) *so that you can notice it when it arises*.

When constructive interaction becomes destructive conflict

Level 5: World War

- Destroy the Other!
- Little or No Language is Exchanged

Level 4: Crusade

- Protecting One's Own Group becomes the Focus
- Language is Ideological

Level 3: Contest

- Winning trumps Resolving
- Language includes Personal Attacks

Level 2: Disagreement

- Personal Protection trumps Collaboration
- Language is Guarded and Open to Interpretation

Level 1: Problem to Solve

- Information Sharing and Collaboration
- Language is Open and Fact-based

Source: Speed Leas (1985); Lyssa Adkins (2010)

What is Open Discussion?

unstructured dialogue within a meeting

allows for the greatest freedom in the conversation, which also makes it the approach most likely to result in dysfunctional meeting interactions

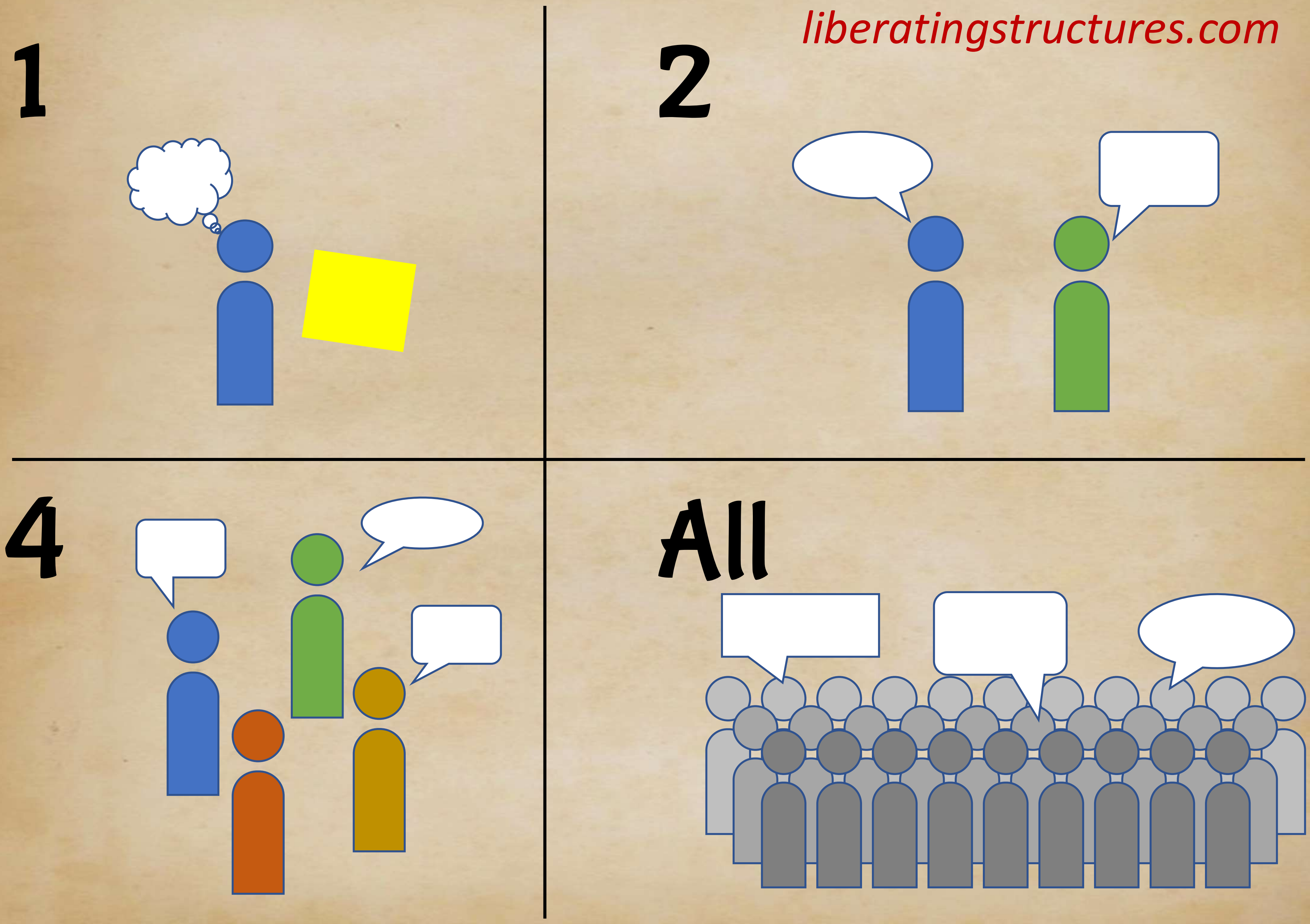
Alternatives to Open Discussion

Listing Ideas
Small Groups
Gallery Walk
Roleplays / Skits
Fishbowls
Ask The Expert

Individual Writing
Debate Mode
Presentations / Reports
Trade Show / Science Fair
Rotating Breakout Groups
Structured Go-Arounds

1-2-4-All is a *Liberating Structure*

liberatingstructures.com



1

2

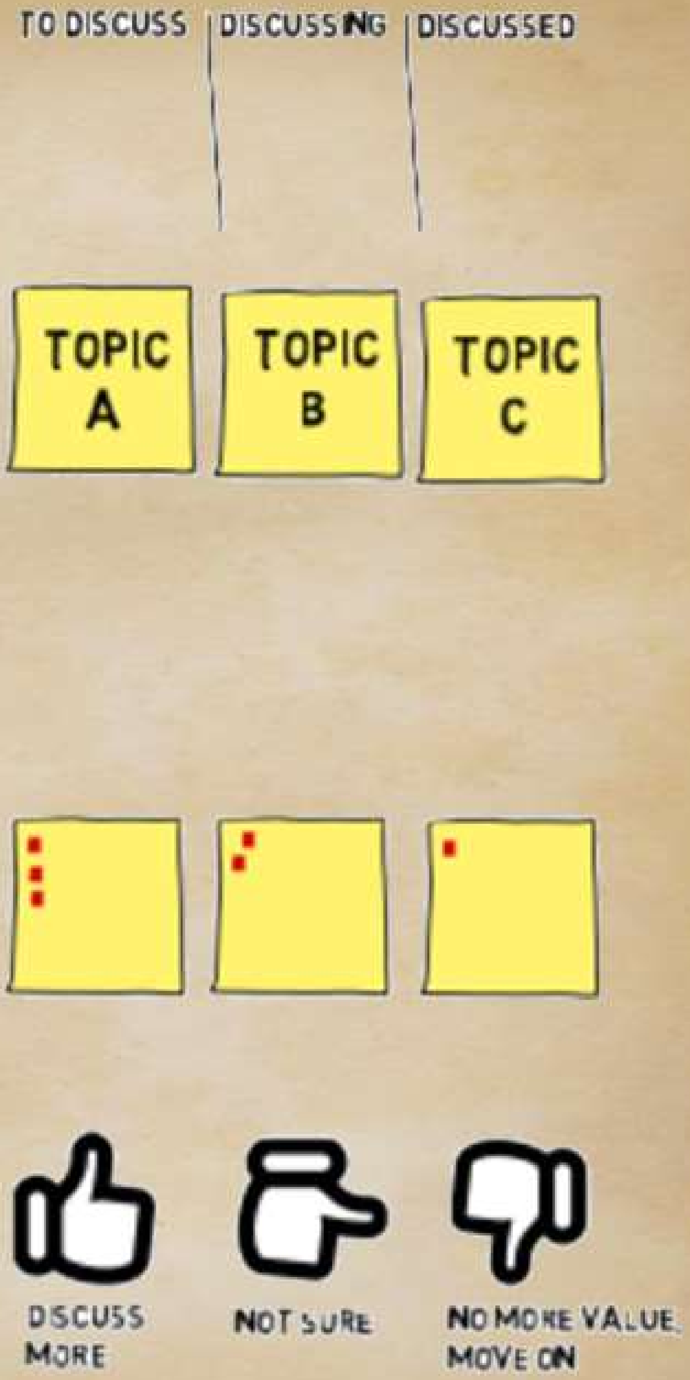
4

All

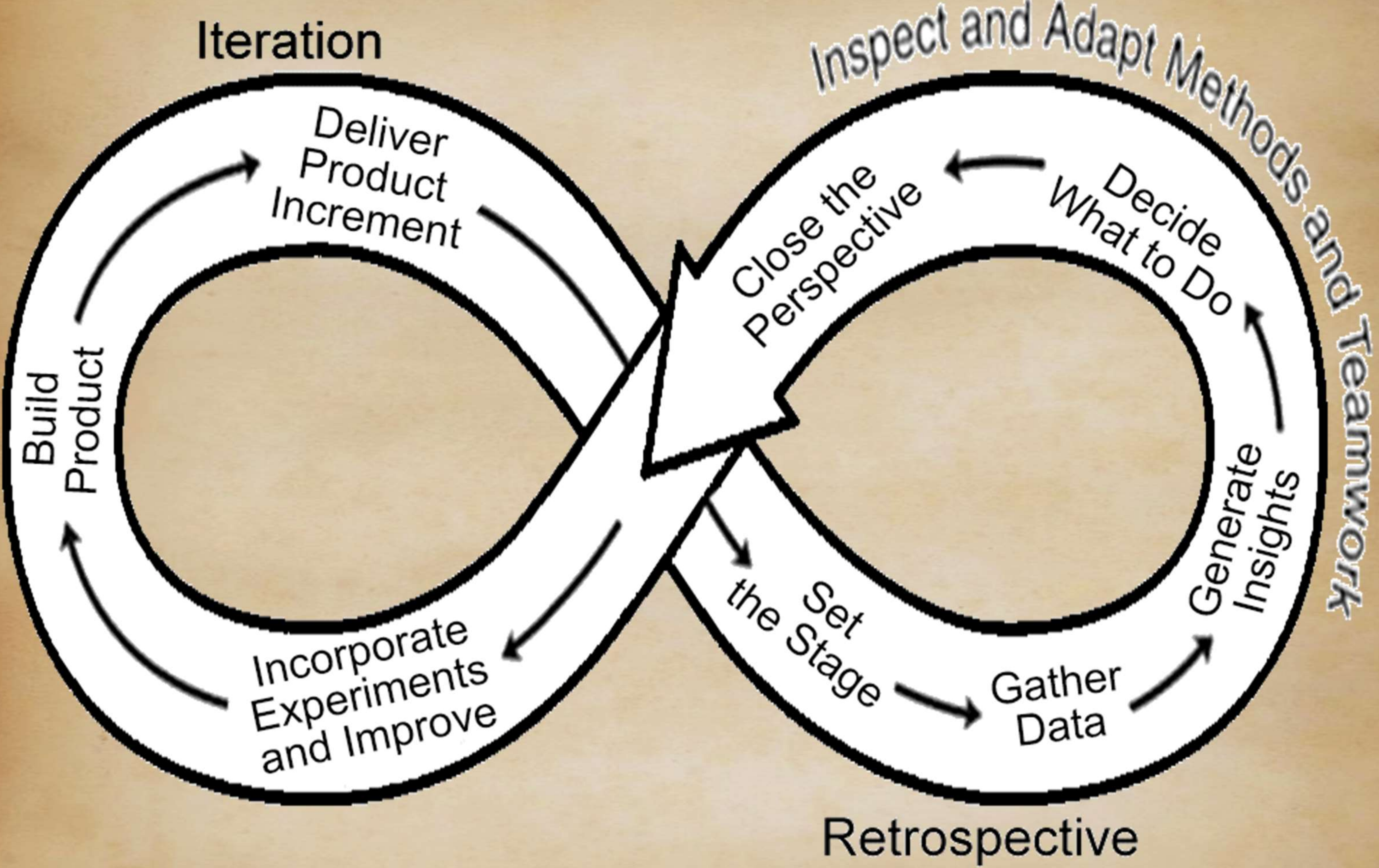
Lean Coffee

1. create a simple kanban board
2. brainstorm topics
3. pitch your topic(s)
4. prioritize what to discuss
5. manage flow of conversation
6. Discuss!

LeanCoffee.org



2009 - Benson and Lightsmith - LeanCoffee.org



from Agile Retrospectives, 2006 – Derby and Larsen

Sprint Retrospective: References & Tools

- Agile Retrospectives: Esther Derby & Diana Larsen
- Sites:
  - RetroMat: retromat.org
  - Explore Retro Templates: thevirtualagilecoach.co.uk/retros
  - Retro Resource Wiki: retrospectivewiki.org
  - Idea Boardz: www.ideaboardz.com
- Any other tools you have used for either co-located or distributed teams?

Assignment #2:

When you get back to a work environment, design and facilitate TWO Collaborative Events.

Document these (and any reactions and learnings) in your personal journal.

Scrum Master as Facilitator

- Consider your experience with Scrum and what we have covered regarding facilitation techniques
- In your group, discuss times when the ScrumMaster should **not** act as the facilitator for the Scrum Team
- How would you, as the facilitator, help the Scrum Team address issues that are identified in a Sprint Retrospective

TO DO

- Organizational Change

DAY 2

- Self-managing Teams
- XP & Kanban
- Agile Product Development
- Teaching
- Scaling Scrum / Agile Organization
- Scrum Leadership

DOING

- Coaching

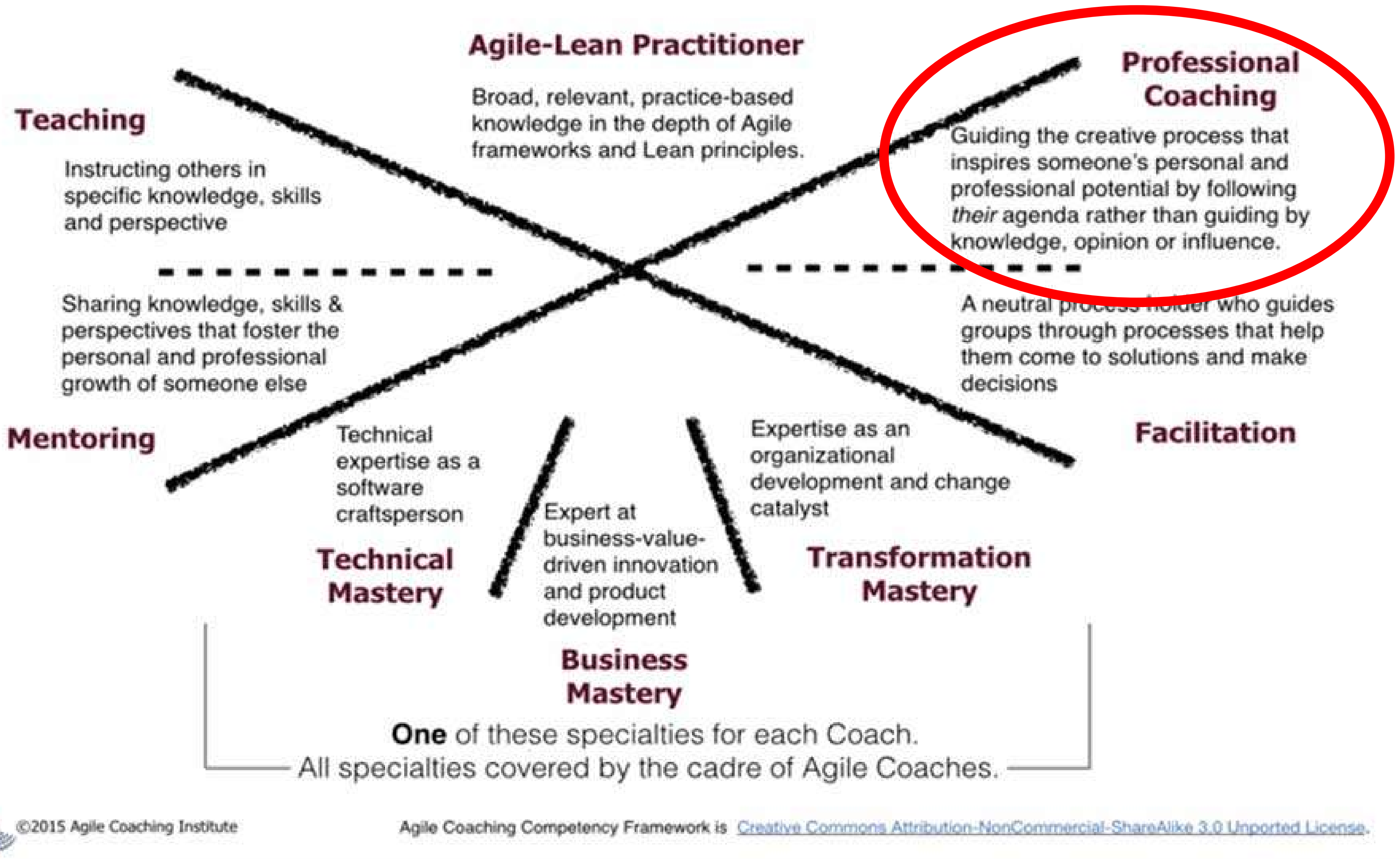
DONE

- Introductions
- Why & What of Agile and Scrum
- Scrum Leadership
- Facilitation

Scrum Master as Coach

What is a Coach?

Agile Coaching Competency Framework



Agile-Lean Practitioner

Broad, relevant, practice-based knowledge in the depth of Agile frameworks and Lean principles.

Teaching

Instructing others in specific knowledge, skills and perspective

Mentoring

Sharing knowledge, skills & perspectives that foster the personal and professional growth of someone else

Professional Coaching

Guiding the creative process that inspires someone's personal and professional potential by following *their* agenda rather than guiding by knowledge, opinion or influence.

Facilitation

A neutral process holder who guides groups through processes that help them come to solutions and make decisions

Technical Mastery

Technical expertise as a software craftsperson

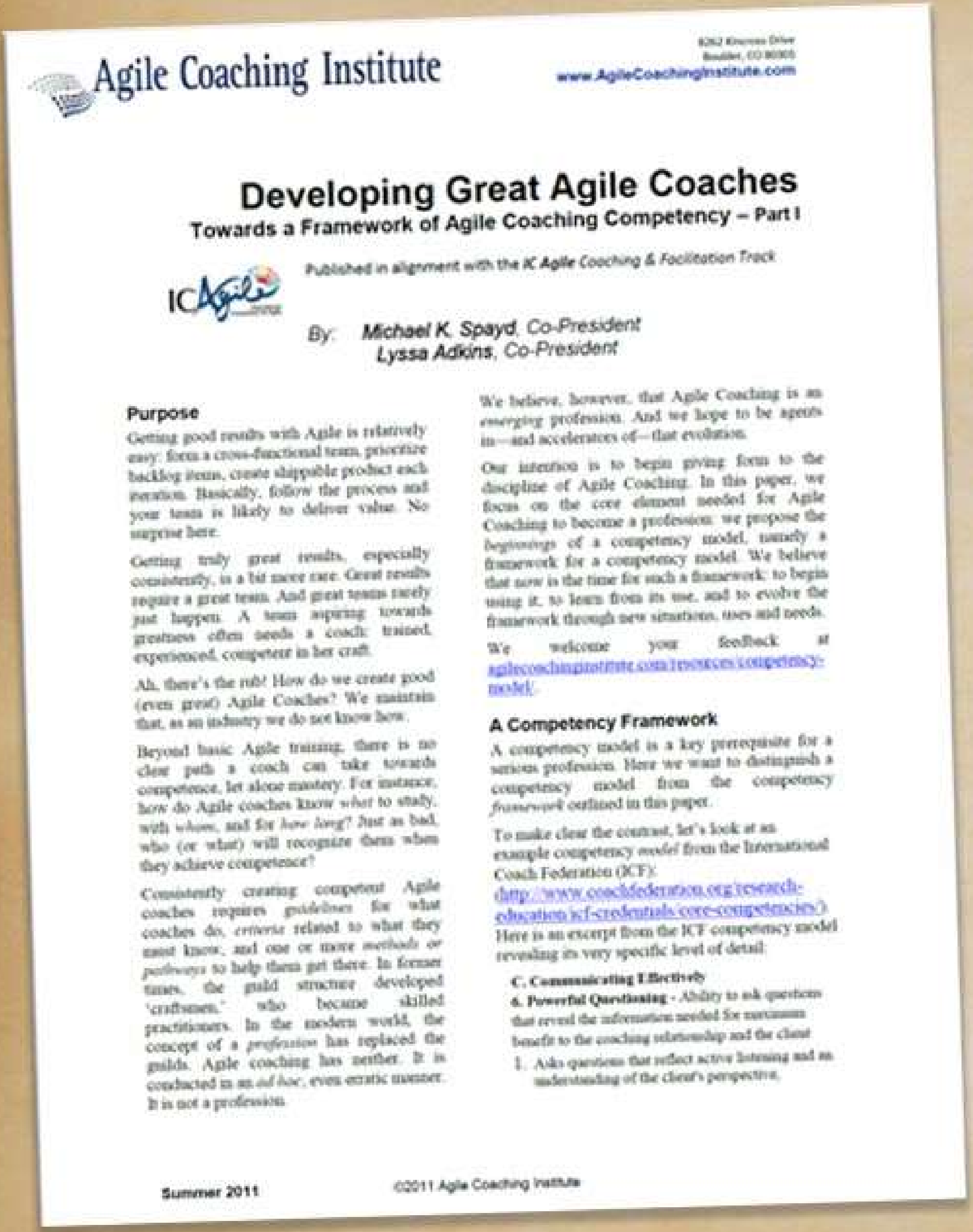
Business Mastery

Expert at business-value-driven innovation and product development

Transformation Mastery

Expertise as an organizational development and change catalyst

One of these specialties for each Coach.
All specialties covered by the cadre of Agile Coaches.



Agile Coaching Institute

6262 Kincross Drive
Boulder, CO 80301
www.AgileCoachingInstitute.com

Developing Great Agile Coaches

Towards a Framework of Agile Coaching Competency – Part I

Published in alignment with the IC Agile Coaching & Facilitation Track

ICAgile

By: ***Michael K. Spayd****, Co-President*
Lyssa Adkins*, Co-President*

Purpose

Getting good results with Agile is relatively easy: form a cross-functional team, prioritize backlog items, create shippable product each iteration. Basically, follow the process and your team is likely to deliver value. No surprise here.

Getting truly great results, especially consistently, is a bit more rare. Great results require a great team. And great teams rarely just happen. A team aspiring towards greatness often needs a coach: trained, experienced, competent in her craft.

Ah, there's the rub! How do we create good (even great) Agile Coaches? We maintain that, as an industry we do not know how.

Beyond basic Agile training, there is no clear path a coach can take towards competence, let alone mastery. For instance, how do Agile coaches know *what* to study, with *whom*, and for *how long*? Just as bad, who (or what) will recognize them when they achieve competence?

Consistently creating competent Agile coaches requires *guidelines* for what coaches do, *criteria* related to what they must know, and one or more *methods or pathways* to help them get there. In former times, the guild structure developed 'craftsmen,' who became skilled practitioners. In the modern world, the concept of a *profession* has replaced the guilds. Agile coaching has neither. It is conducted in an *ad hoc*, even erratic manner. It is not a profession.

We believe, however, that Agile Coaching is an *emerging* profession. And we hope to be agents in—and accelerators of—that evolution.

Our intention is to begin giving form to the discipline of Agile Coaching. In this paper, we focus on the core element needed for Agile Coaching to become a profession: we propose the *beginnings* of a competency model, namely a framework for a competency model. We believe that *now* is the time for such a framework: to begin using it, to learn from its use, and to evolve the framework through new situations, uses and needs.

We welcome your feedback at agilecoachinginstitute.com/resources/competency-model/.

A Competency Framework

A competency model is a key prerequisite for a serious profession. Here we want to distinguish a competency model from the competency *framework* outlined in this paper.

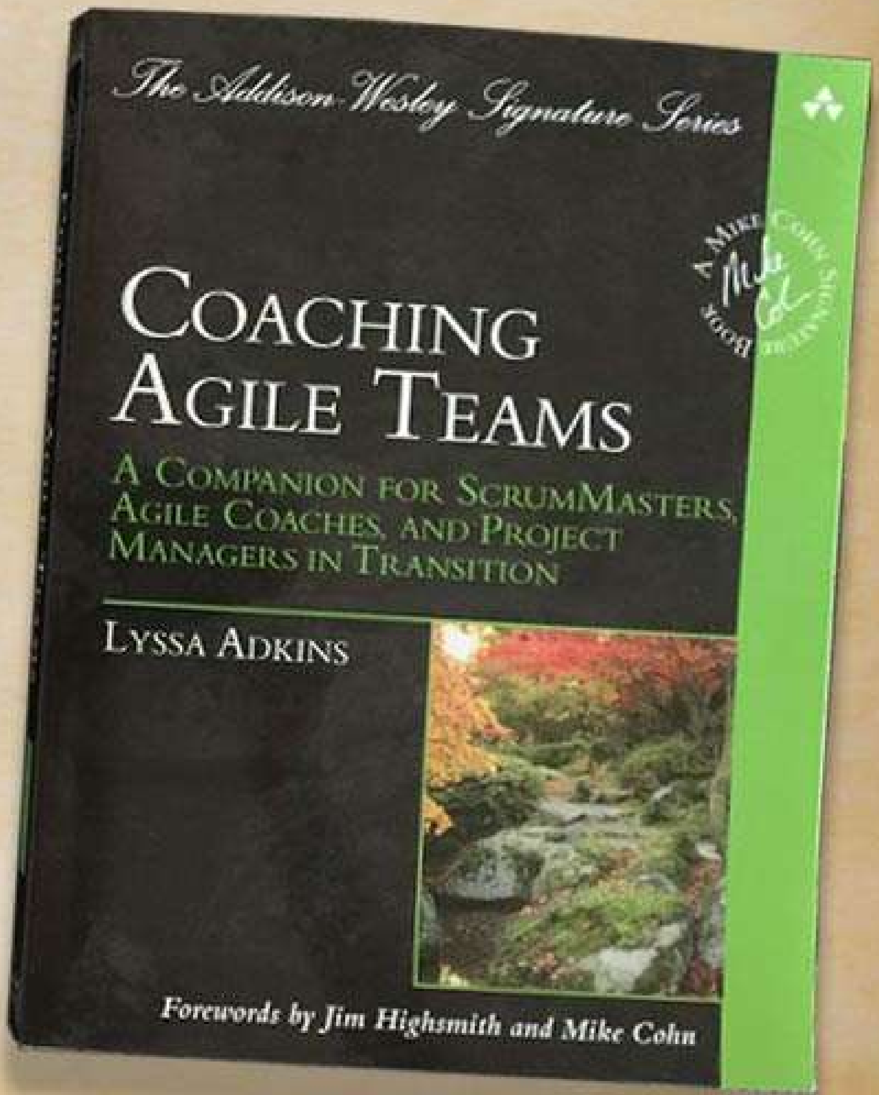
To make clear the contrast, let's look at an example competency *model* from the International Coach Federation (ICF): (http://www.coachfederation.org/research-education/icf-credentials/core-competencies/). Here is an excerpt from the ICF competency model revealing its very specific level of detail:

C. Communicating Effectively

6. Powerful Questioning - Ability to ask questions that reveal the information needed for maximum benefit to the coaching relationship and the client

1. Asks questions that reflect active listening and an understanding of the client's perspective.

Summer 2011


The Addison-Wesley Signature Series
A Mike Cohn Signature Book
Coaching Agile Teams
A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition
Lyssa Adkins
Forewords by Jim Highsmith and Mike Cohn

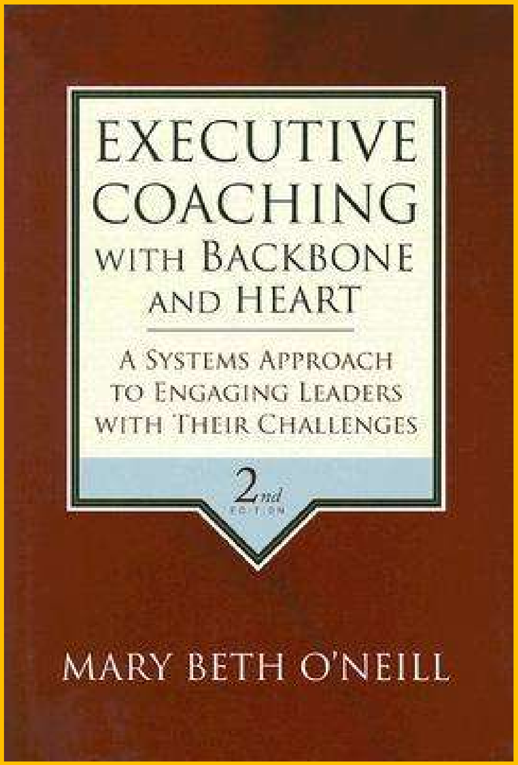
The ‘Coaching Stance’: The Heart of the Competency Framework

What we call the “coaching stance” is for us the heart of the competency framework because it informs all the other competencies. The coaching stance is the place we start from, the place we return, and ideally the place we include when using any of the other competencies.

> Without the coaching stance, the work of an Agile Coach may become manipulative, ideological, and driven by the coach’s own desires—even when those desires are seemingly benign and believed to be in the ‘best interest’ of the client.

Watch-words of the Coaching Stance:

- Maintaining neutrality
- Serving the client's agenda
- Reducing client dependence
- Not colluding
- Signature presence



EXECUTIVE COACHING WITH BACKBONE AND HEART

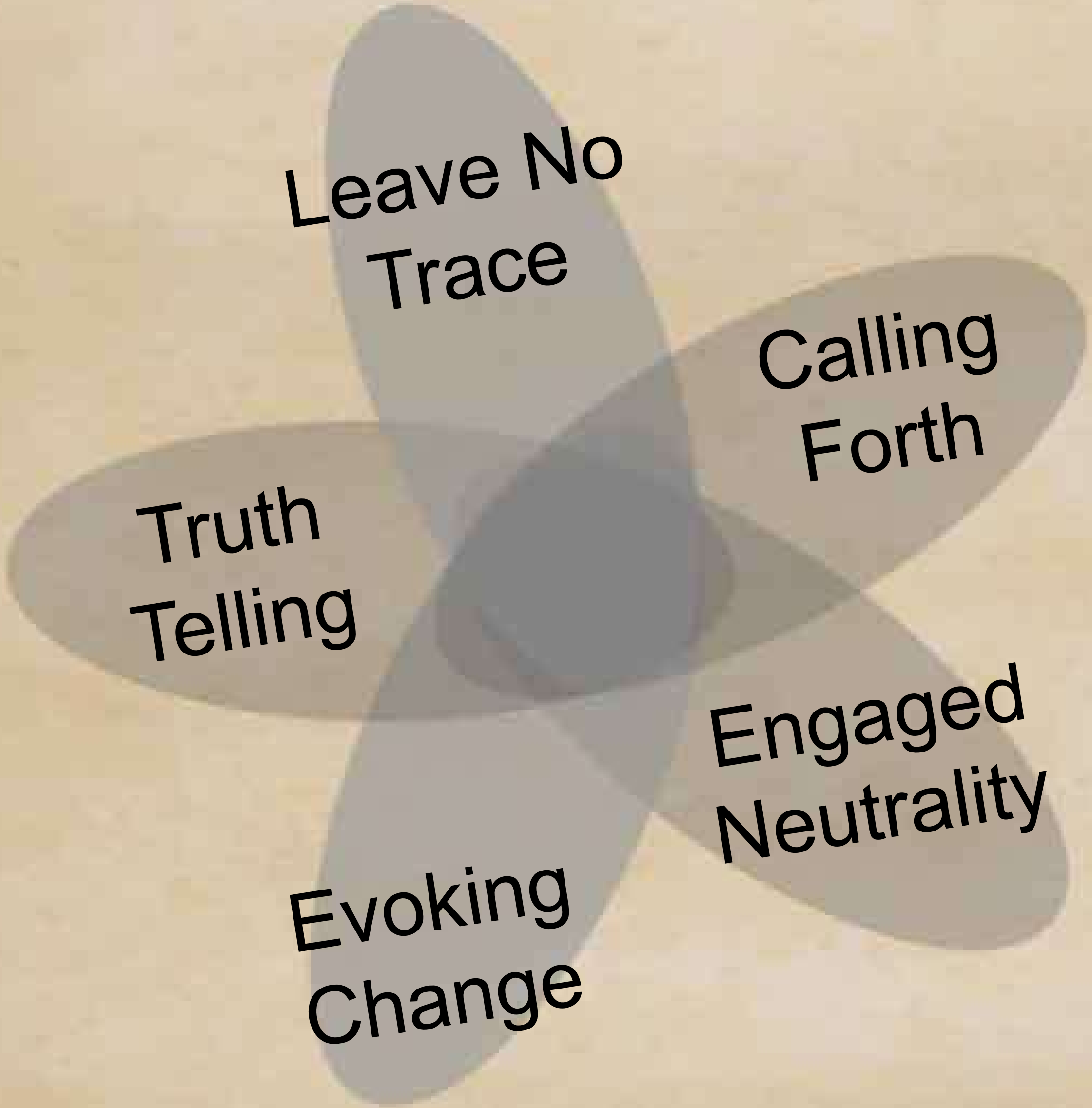
A SYSTEMS APPROACH TO ENGAGING LEADERS WITH THEIR CHALLENGES

2nd EDITION

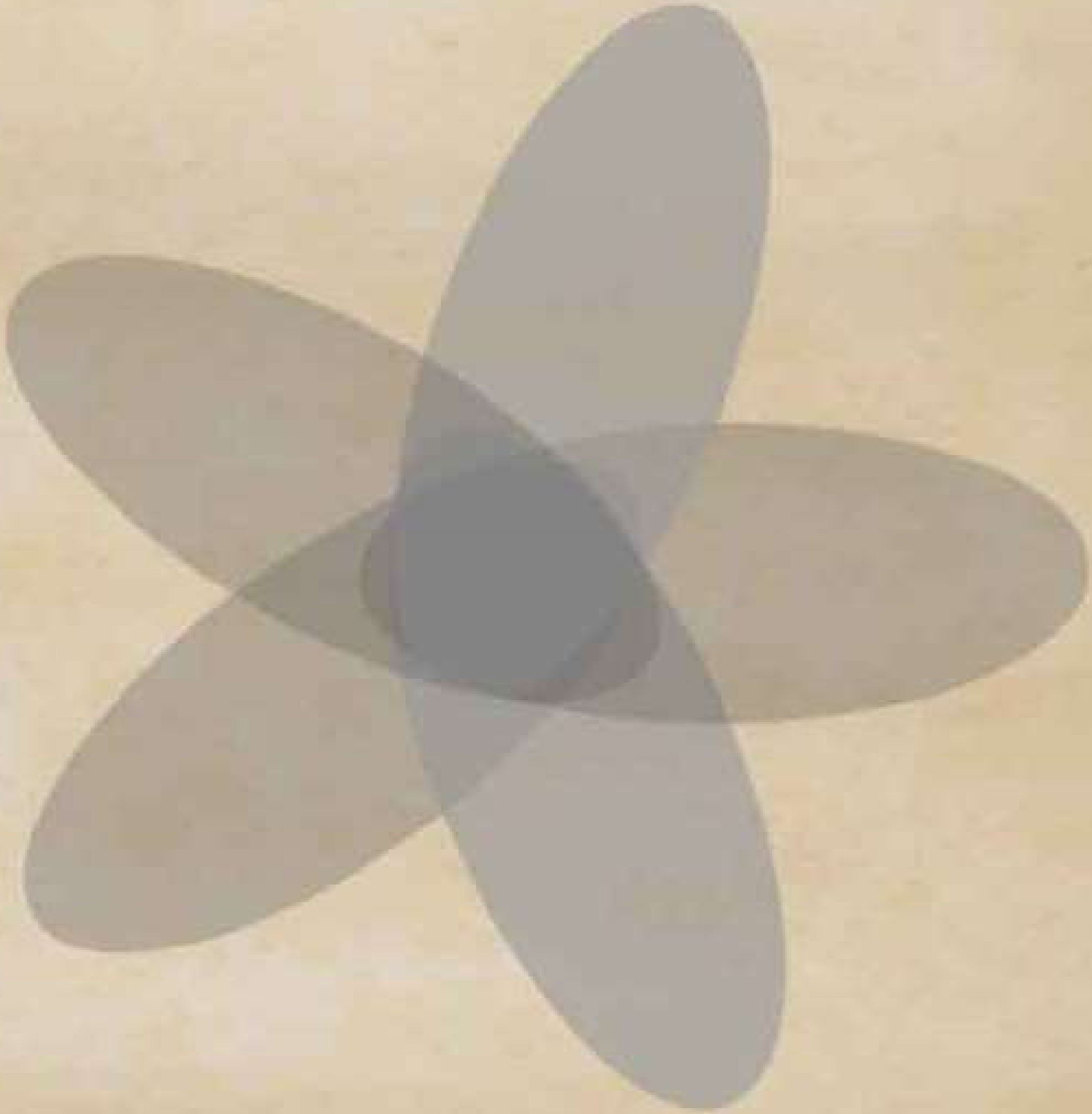
MARY BETH O'NEILL

RECOMMENDED READING

One Possible Stance



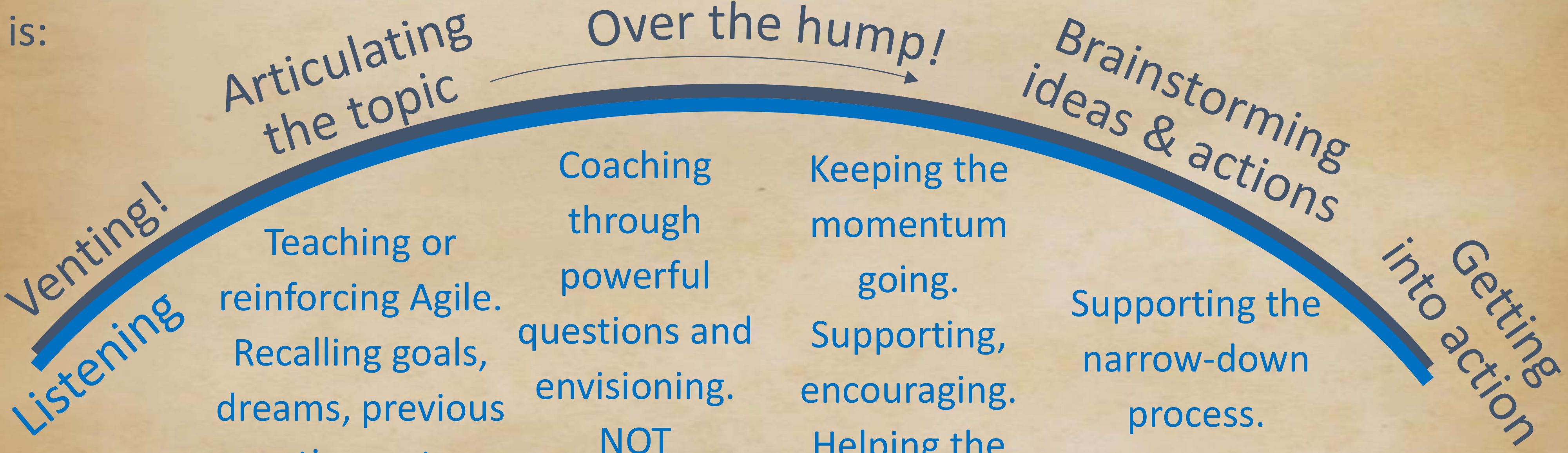
What’s Your Stance?



Coaching Uses Several Techniques

The Arc of the Coaching Conversation

COACHEE is:



Venting!

Articulating the topic

Over the hump!

Brainstorming ideas & actions

Getting into action

Listening

Teaching or reinforcing Agile. Recalling goals, dreams, previous actions, etc.

Coaching through powerful questions and envisioning. NOT problem-solving.

Keeping the momentum going. Supporting, encouraging. Helping the coachee get BIG!

Supporting the narrow-down process. Creating an accountability for the action chosen.

Acknowledge the coachee for who they are being.

COACH is:

Coaching Skills: Listening

Level 1: **Internal Listening** (*Listening to Me*)

- Listening to own, inner voice and thoughts
- Making judgements and assumptions

Level 2: **Focused Listening** (*Listening to You*)

- Laser-focused on client; inner voice is ignored
- Empty mind before the conversation
- Beware writing while the other person is talking

Level 3: **Global Listening** (*Listening to Us*)

- Tuned in to non-verbal gestures and facial expressions
- Picking up on mood from tones used within the group

WAIT

Why

Am

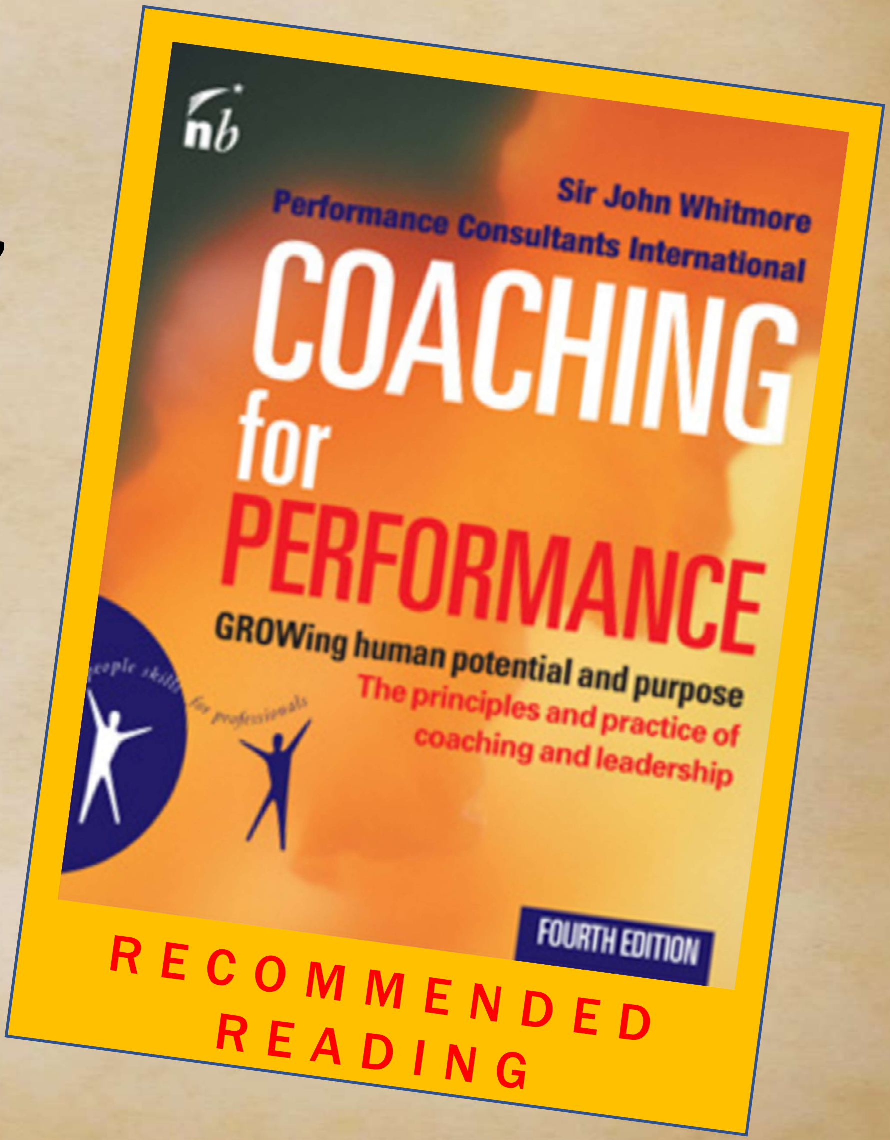
I

Talking

?

The GROW Model

- We are using the version made popular in the book Coaching for Performance, by Sir John Whitmore
- This approach does not just focus on growing performance but also on growing learning and enjoyment



nb

Sir John Whitmore
Performance Consultants International

COACHING for PERFORMANCE

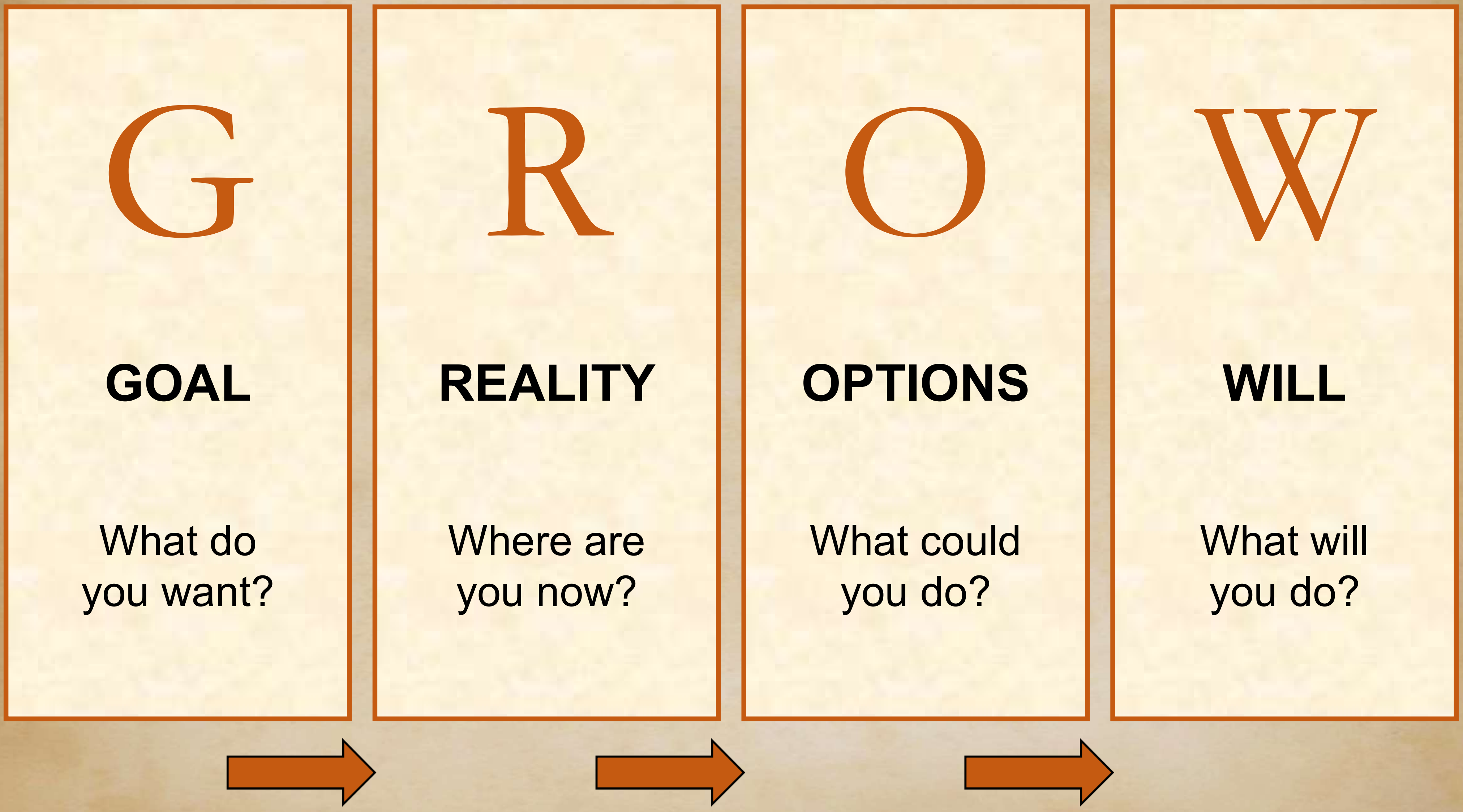
GROWing human potential and purpose
The principles and practice of coaching and leadership

people skills for professionals

FOURTH EDITION

RECOMMENDED READING

The GROW Model



G

GOAL

What do you want?

R

REALITY

Where are you now?

O

OPTIONS

What could you do?

W

WILL

What will you do?

Grow Model: **Goal**

- What would you like to have by the end of the session?
- Imagine a year from now. What’s the ideal state?
- What elements are you most drawn to?
- How important is each? (Rank, i.e. 1-10)
- So your goal would be what? By when?
- Deep down, what do you really want?
- What would be some stepping stones?
- What would be a good first step from where you are now?

G

Grow Model: **Reality**

- How much of this do you feel is within your control?
- On a scale of 1–10, if an ideal situation is 10, what number are you at now? What number would you like to be at?
- What gives the most dissatisfaction?
- What is the concern?
- What could meet that concern?
- What sort of people and activities do you dislike?
- What gives you most satisfaction?
- What sort of people and activities do you like?

R

Grow Model: **Options** (or ***Obstacles***)

- What ideas or options do you have?
- What would the benefits be? Costs?
- What alternatives do you have? Is there anything else?
- If there were anything else, what would it be?
- What has worked in the past?
- What steps could you take?
- Who could help you with this?
- Where could you find out the information?
- How could you do that?

O

Grow Model: **Will** (or ***Way Forward***)

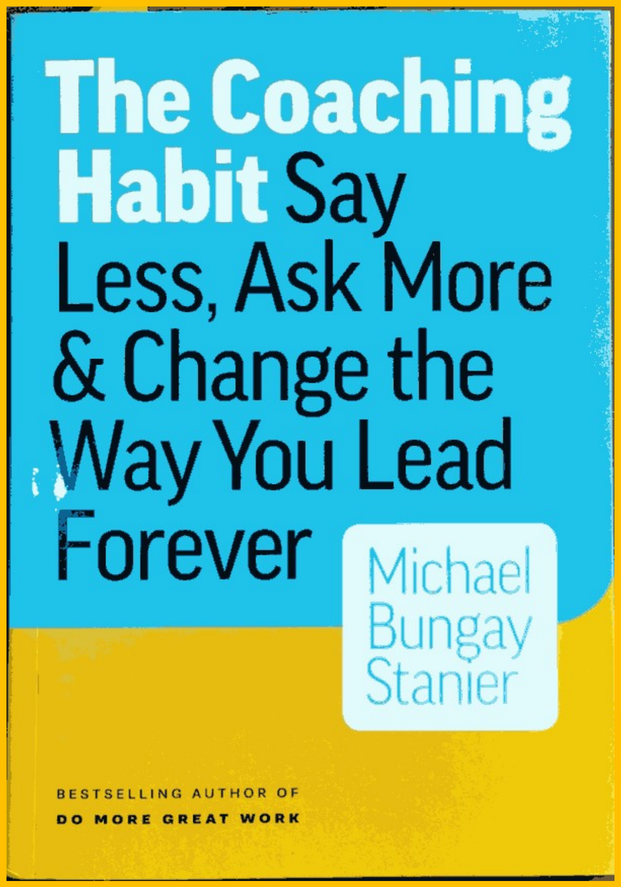
W

- What will you do?
- How will you do it?
- When will you do it?
- What obstacles exist?
- Who needs to know?
- On a scale from 1-10, how motivated are you to do it?
- What will it take for you to commit to that?
- What are the specific tasks, times, costs, etc.?
- How/when would you like me to check in with you?

The Coaching Habit

Seven Questions:

- Kickstart
  What's on your mind?
- AWE
  And what else?
- Focus
  What's the real challenge?
- Foundation
  What do you want?
- Lazy
  How can I help?
- Strategic
  What will you say No to?
- Learning
  What was most useful for you?


The Coaching Habit Say Less, Ask More & Change the Way You Lead Forever
Michael Bungay Stanier
BESTSELLING AUTHOR OF
DO MORE GREAT WORK

RECOMMENDED READING

Coaching with Powerful Questions

Open-ended

Asked with genuine curiosity

- **Build Understanding**

What challenges are you facing?
What matters to you right now?
What's on your mind today?
What opportunities are you seeing?

- **Set Direction**

What is the best possible outcome?
What are you trying to achieve?
What does success look like?
How will you know if you've succeeded?

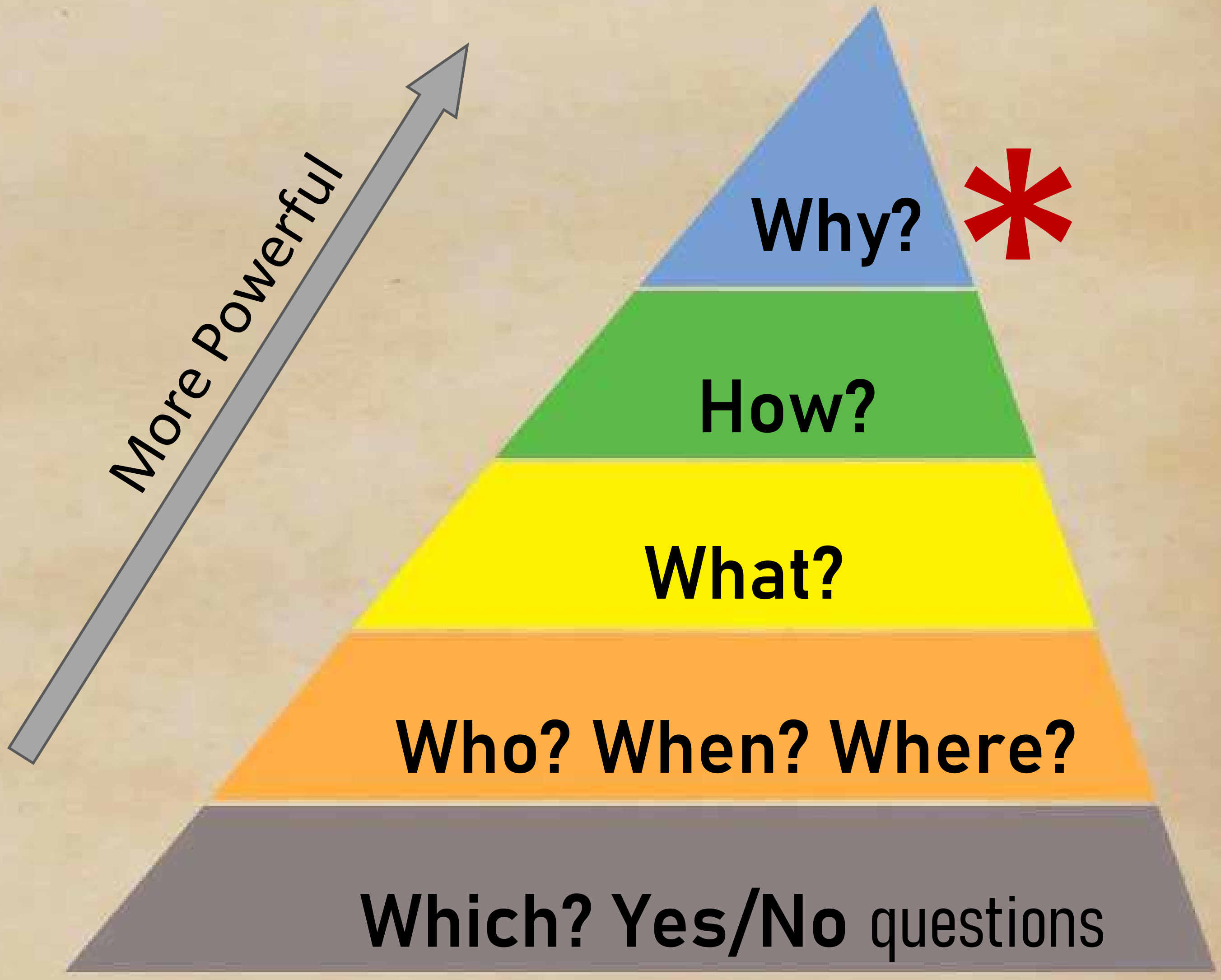
- **Shape Options**

What have you tried?
What options do you have?
How possible is each option?
What would you have to believe for this option to be right?

- **Define Next Actions**

What info / data do you need to make a decision?
What action can you take now?
What are you taking away from this conversation, as a next step or new way of thinking?
What support do you need? Where will you get it?
How can I help?

Powerful Coaching Questions



More Powerful

Why? *

How?

What?

Who? When? Where?

Which? Yes/No questions

The Art of Powerful Questions, Eric E. Vogt, Juanita Brown & David Isaacs

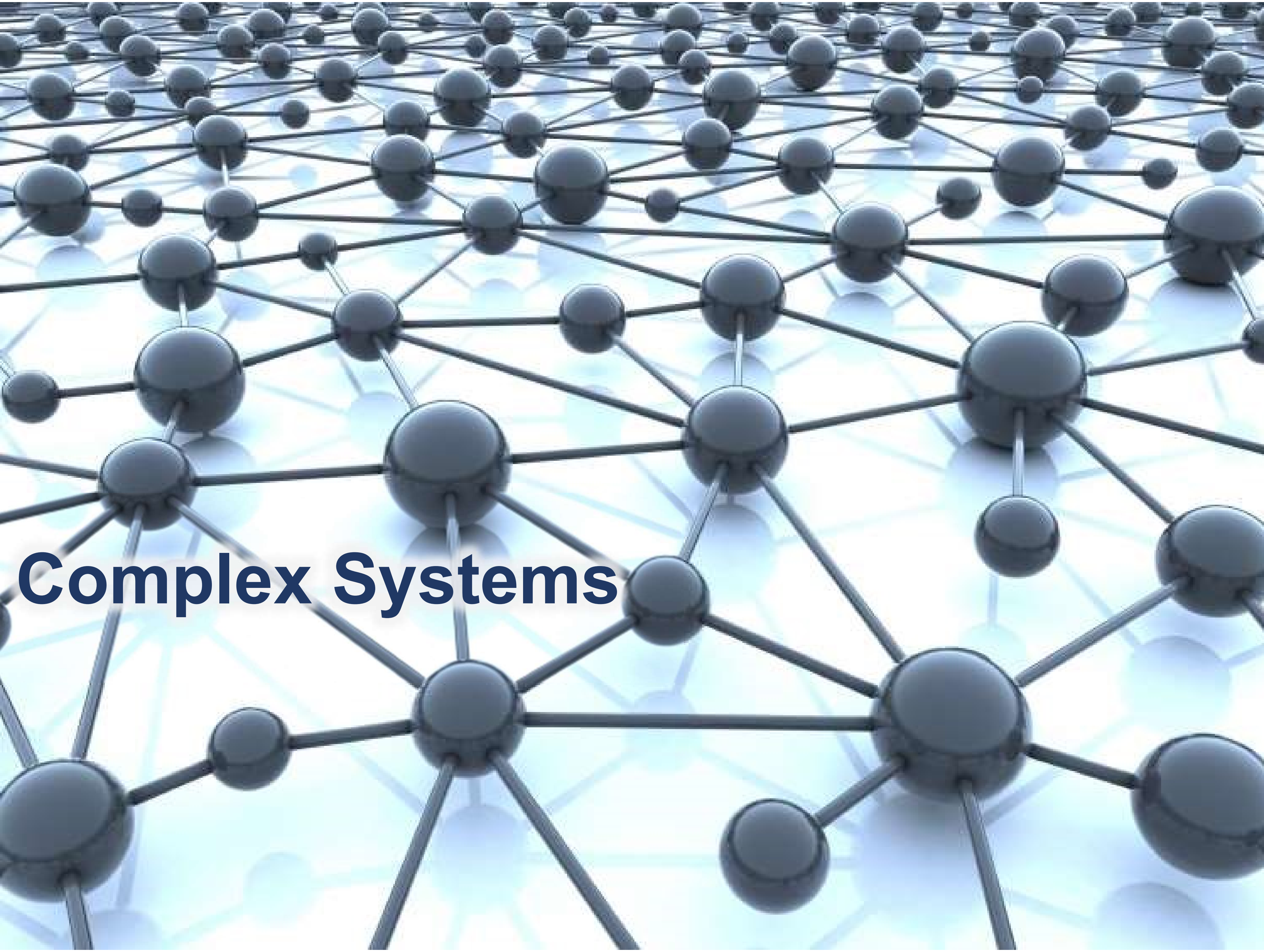
Assignment #3:

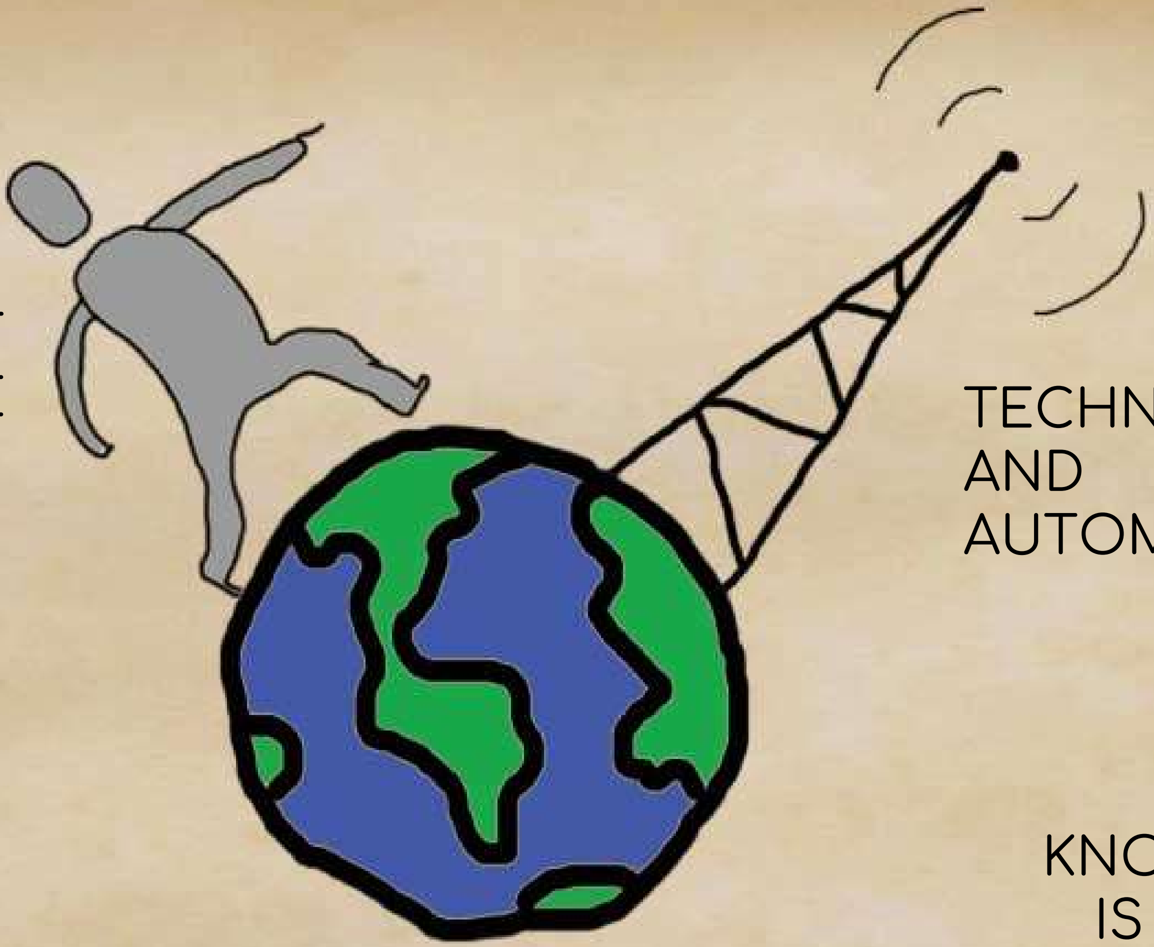
When you get back to a work environment, apply an appropriate coaching technique for two interventions.

Document these (with observations and learnings) in your personal journal.

TO DO	DOING	DONE
DAY 2	Organizational Change	Why & What of Agile and Scrum
Self-managing Teams		Scrum Leadership
XP & Kanban		Facilitation
Agile Product Development		Coaching
Teaching		
Scaling Scrum / Agile Organization		
Scrum Leadership		

Complex Systems





PACE OF CHANGE

TECHNOLOGY AND AUTOMATION

CUSTOMER EXPECTATIONS

KNOWLEDGE IS NOW A COMMODITY

GLOBALIZATION

PRODUCT COMPLEXITY

HYPER-COMPETITION

INCREASED REGULATIONS

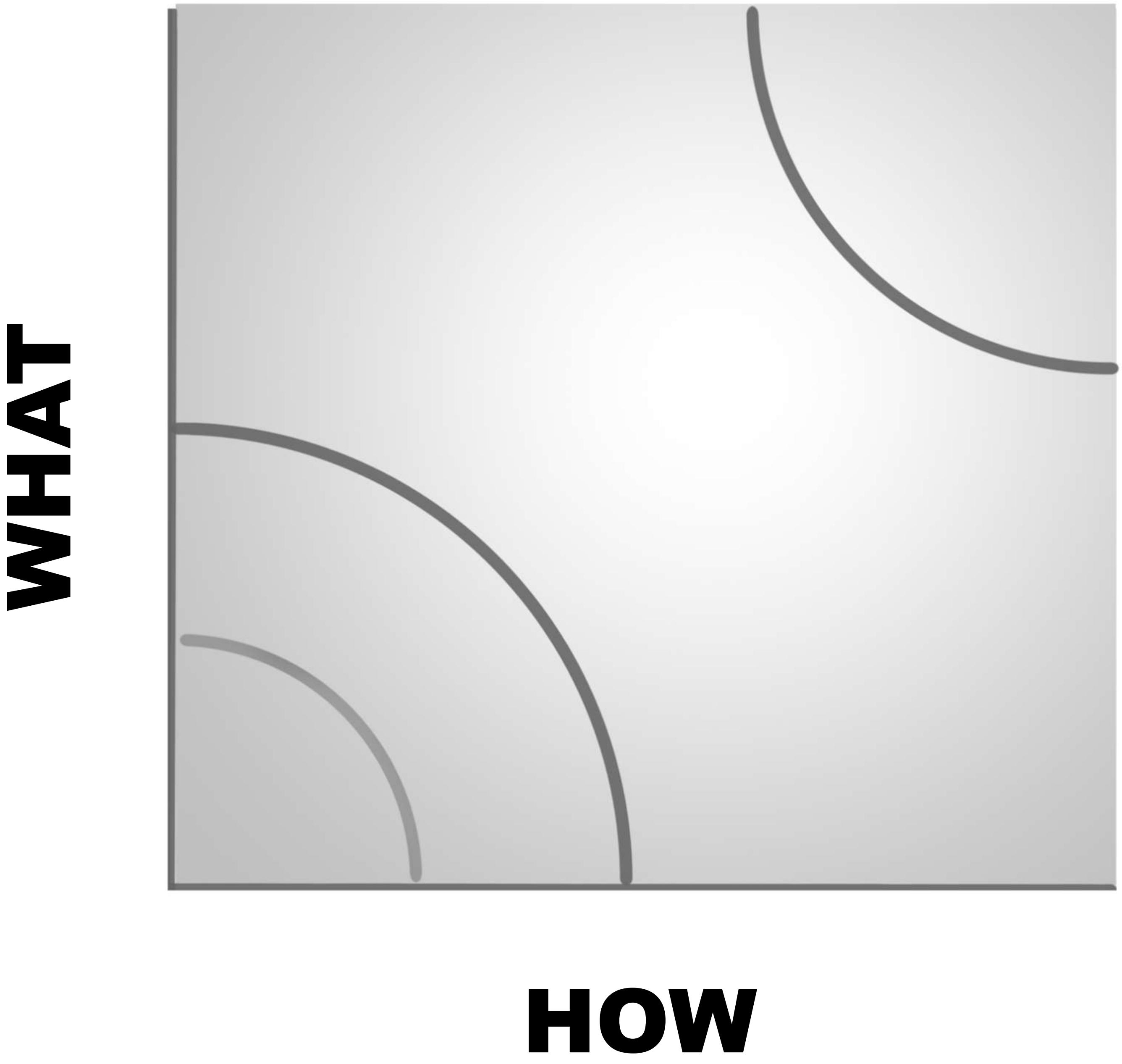
illustration inspired by Pete Behrens, Trail Ridge Consulting

Simple *vs.* Complex

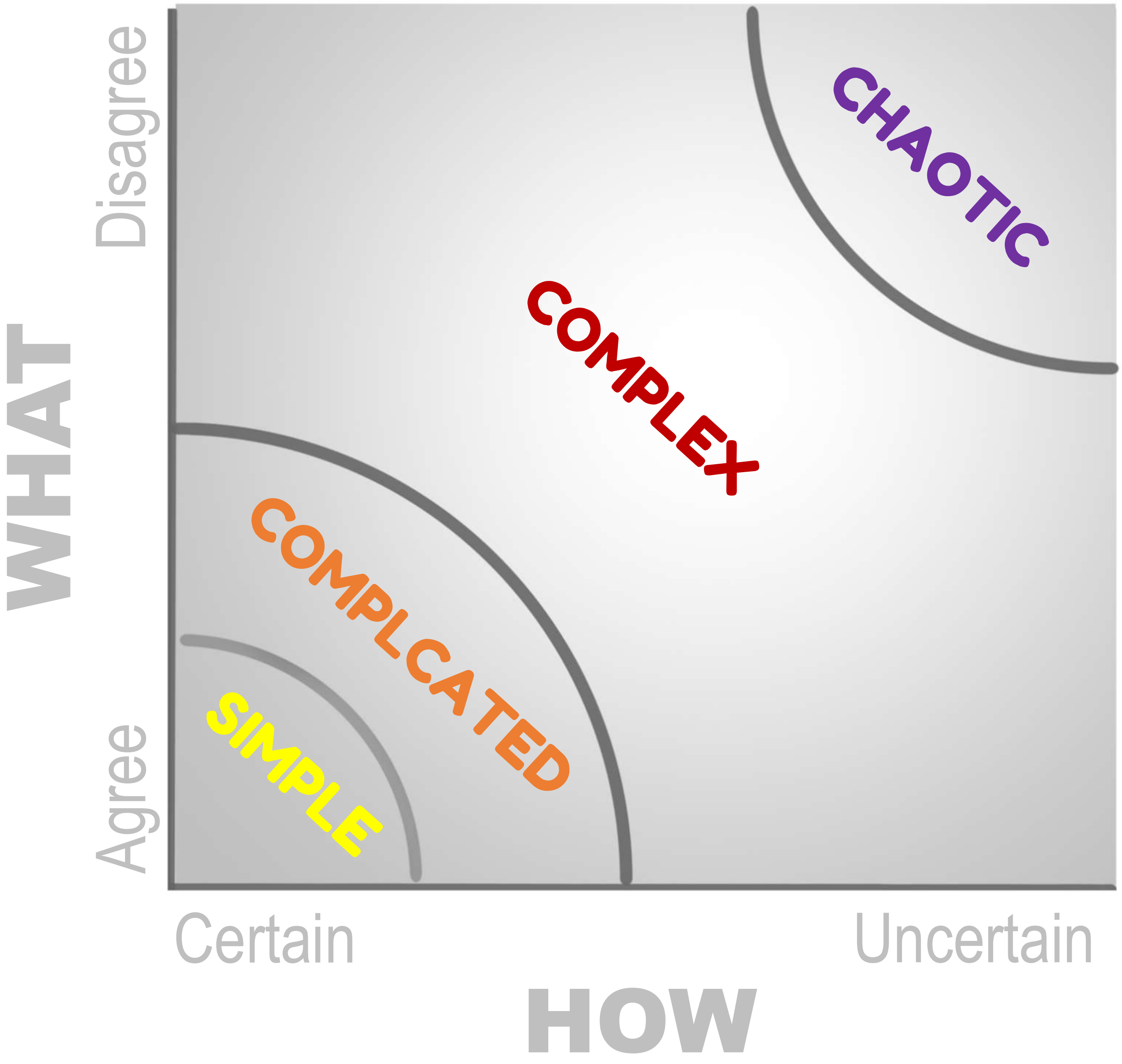
- Cause-and-Effect works
- Probability and forecasting work
- stable, repeatable and predictable
- study history > forecast and plan the future

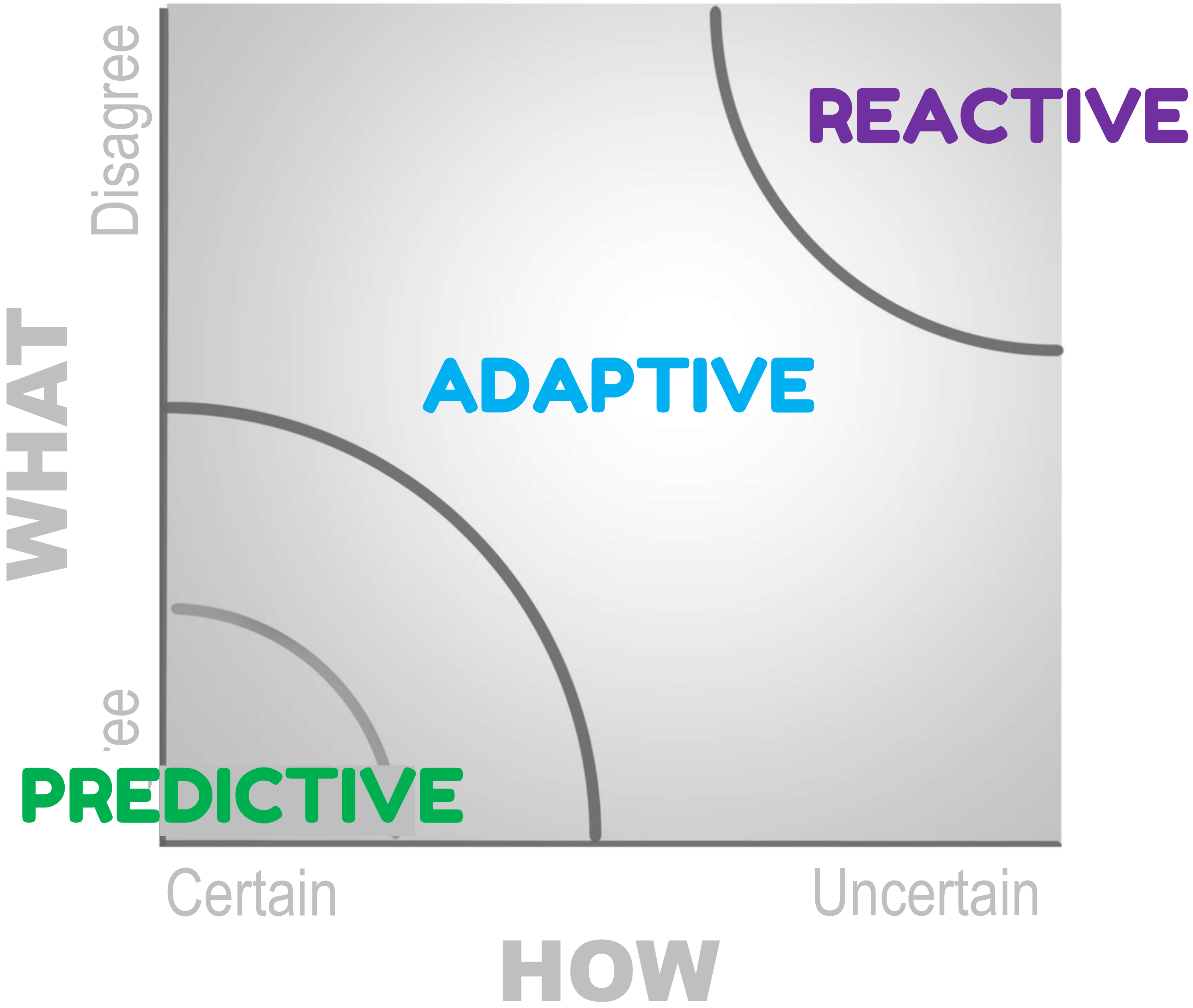
- Cause-and-Effect are only detectable in hindsight
- unpredictable due to massive numbers of variables and influences
- probability is limited
- forecasting creates a false sense of confidence

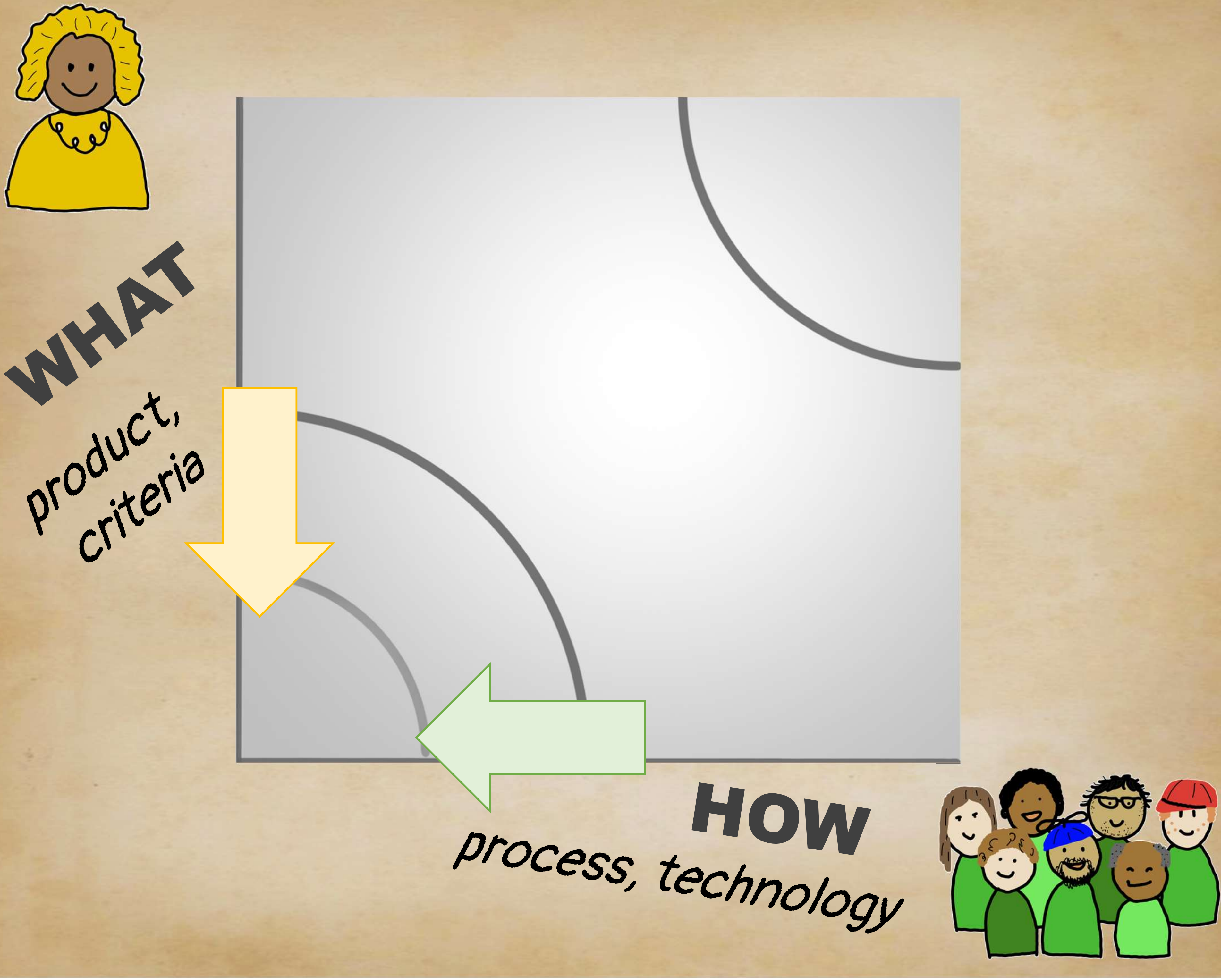
Agreement & Certainty Matrix (Ralph D. Stacey)



Stacey; Strategic management and organisational dynamics: the challenge of complexity. Harlow: Prentice Hall, 2002.


WHAT
Disagree
Agree
CHAOTIC
COMPLEX
COMPLCATED
SIMPLE
Certain
Uncertain
HOW


Disagree
WHAT
ee
REACTIVE
ADAPTIVE
PREDICTIVE
Certain
Uncertain
HOW



WHAT

product, criteria

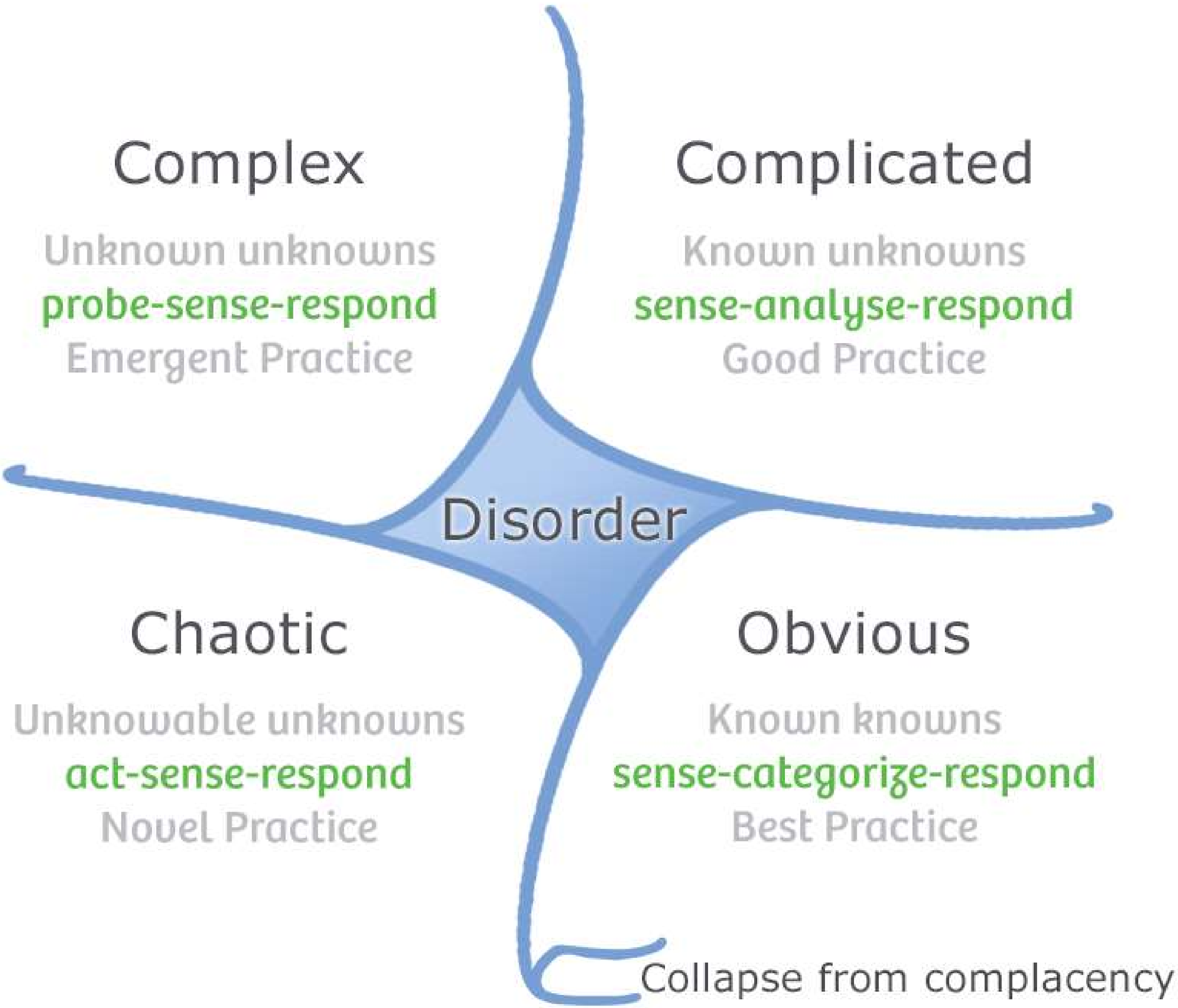
HOW

process, technology

Cynefin

- The Cynefin framework (*kuh-NEV-in*) is a conceptual framework used to aid in decision-making
- Created in 1999 by David Snowden when he worked for IBM Global Services
- Cynefin has been described as a sense-making device – it is not a categorization model
- It recognizes causal differences and gives people a quick and easy way to flip between them applying the appropriate method in the appropriate domain

Video for Reference: https://www.youtube.com/watch?v=N7oz366X0-8



Complex

Unknown unknowns

probe-sense-respond

Emergent Practice

Complicated

Known unknowns

sense-analyse-respond

Good Practice

Disorder

Chaotic

Unknowable unknowns

act-sense-respond

Novel Practice

Obvious

Known knowns

sense-categorize-respond

Best Practice

Collapse from complacency

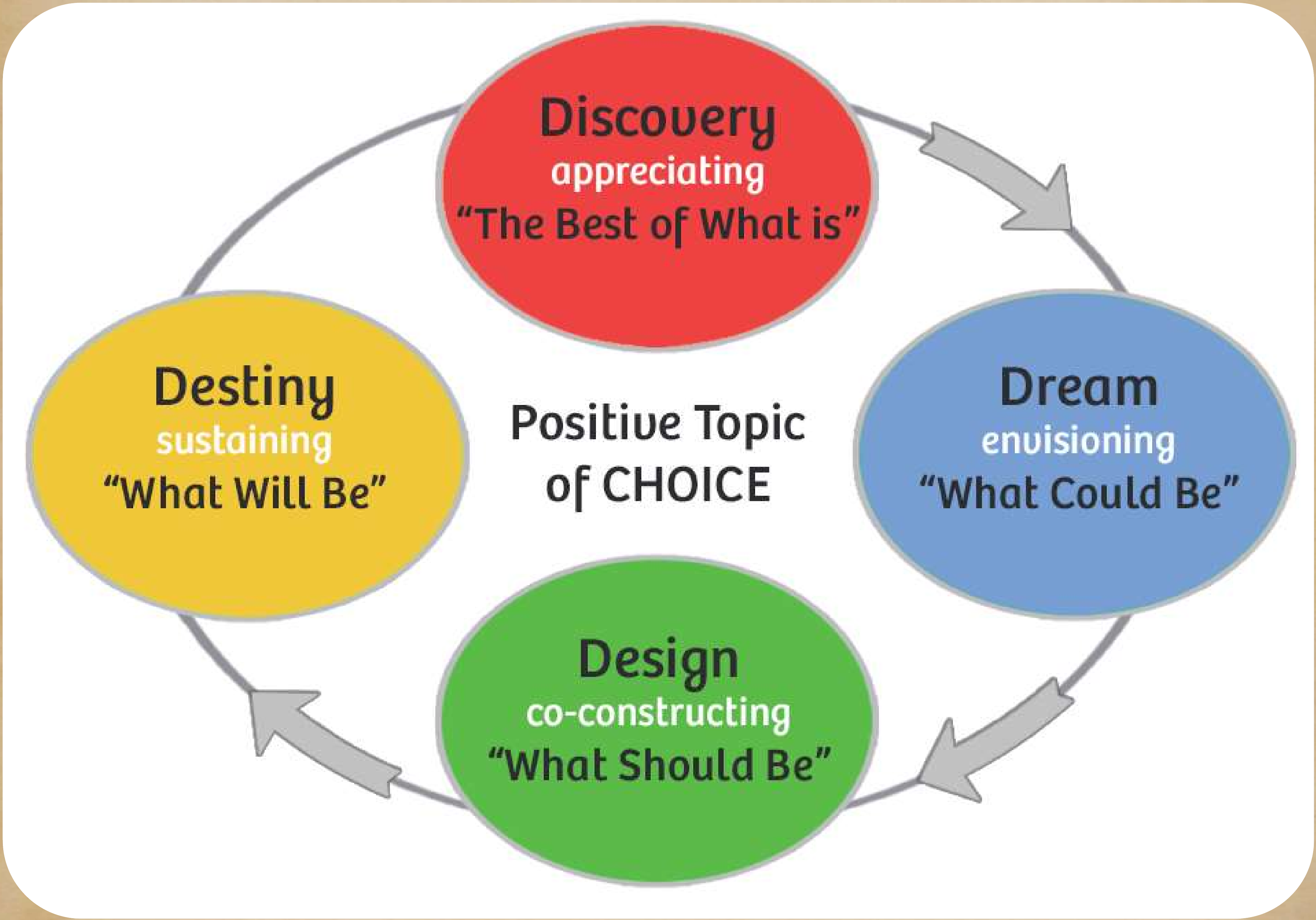
Dave Snowden et al, Cognitive Edge Pte Ltd.

Catalyzing Organizational Change

Some frameworks:

- Appreciative Inquiry
- ADKAR Model
- Kotter 8-Step Process for Leading Change:
  - Create → Build → Form → Enlist → Enable → Generate → Sustain → Institute
- McKinsey & Company’s 7-S Framework:
  - Style, Skills, Systems, Structure, Staff, and Strategies = Shared Values & Goals
- Satir Change Management Model:
  - Late Status Quo → Resistance → Chaos → Integration → New Status Quo

Appreciative Inquiry (AI)



The 4-D Model was developed by Suresh Srivastva, Ron Fry, and David Cooperrider in 1990

Traditional Problem Solving *vs.* Appreciative Inquiry

Traditional Problem Solving

- *Identify the problem*
- *Analyze the causes*
- *Plan the actions*
- *Basic assumption:*
  - *the organization is a problem to be solved*

Appreciative Inquiry

- *Value what is*
- *Envision what could be*
- *Discuss next steps*
- *Basic assumption:*
  - *the organization and the people in it know the possibility*

A.I. is Future-Oriented

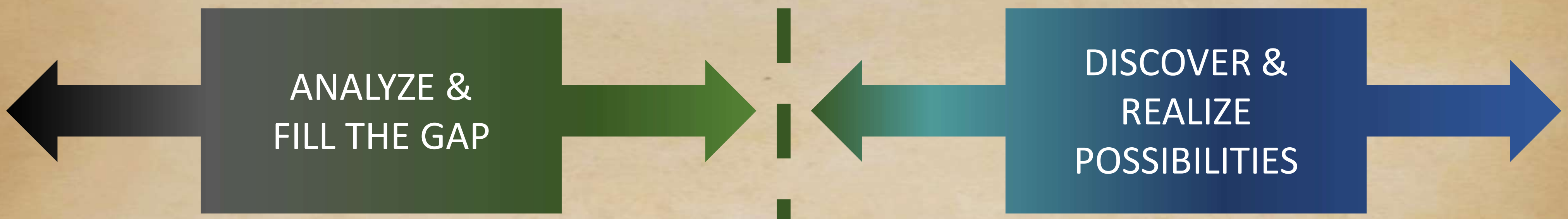
Traditional Problem Solving

Appreciative Inquiry

PAST

CURRENT STATE

FUTURE



ANALYZE & FILL THE GAP

DISCOVER & REALIZE POSSIBILITIES

What's wrong?

What happened?

Who's to blame?

How do we fix it?

What's working?

What's possible?

Where's the passion?

How do we achieve it?

"Appreciative inquiry in organizational life" Cooperrider, D. L. & Srivastva, S. (1987)

ADKAR Model

Awareness *of the need for change*

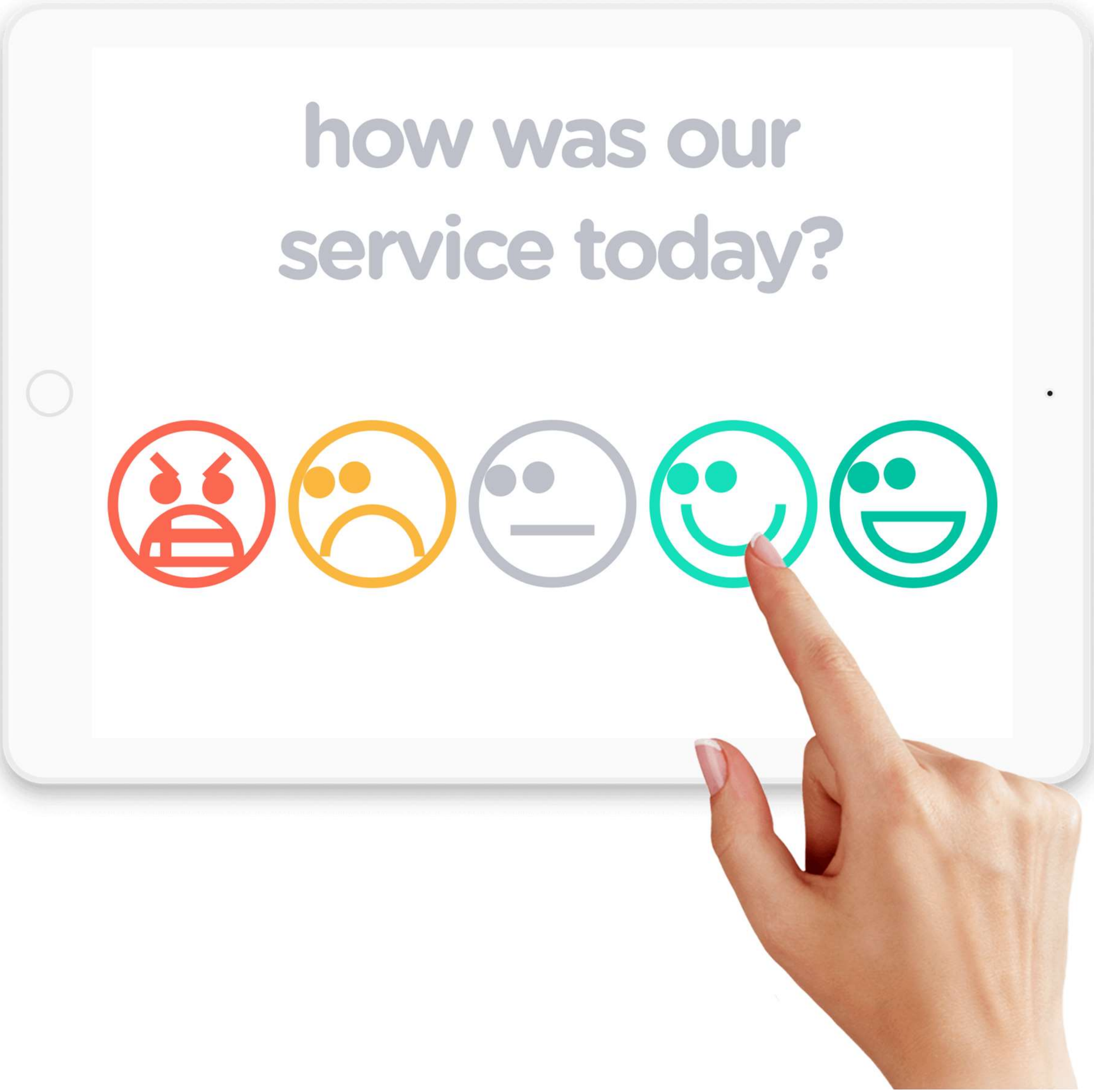
Desire *to support and participate in the change*

Knowledge *of how to change*

Ability *to implement required skills and behaviors*

Reinforcement *to sustain the change*

Prosci Research, Jeffrey Hiatt, 2006


how was our
service today?

END OF DAY 1

See ya tomorrow …

Advanced Certified Scrum Master (A-CSM) Training

Day 2

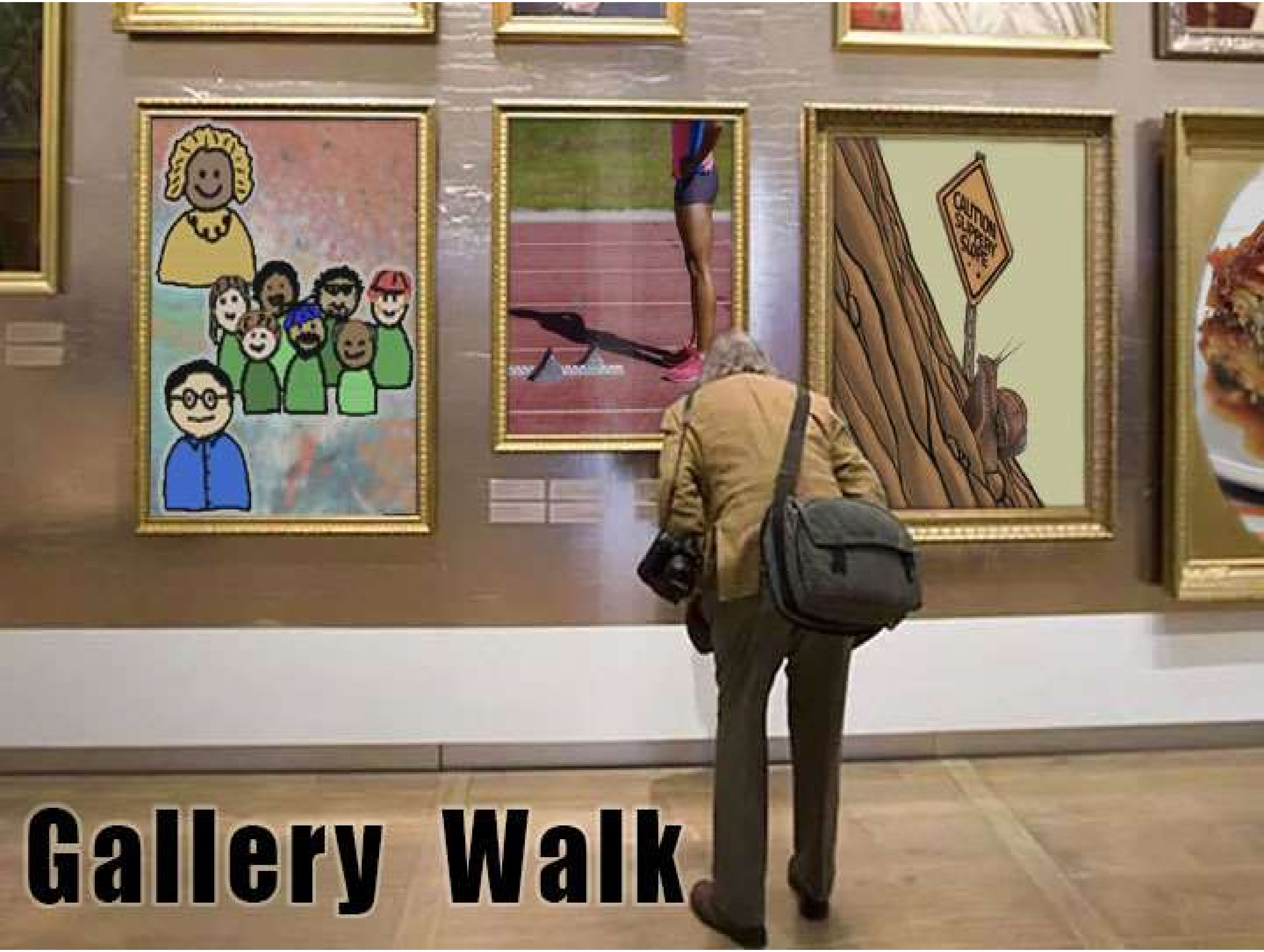
powered by Agile Coffee

ROCKET NINE SOLUTIONS

via zoom

Victor Bonacci
Certified Scrum Trainer
Host of the *Agile Coffee Podcast*
victor@rocketninesolutions.com

V4.1 – Oct 2022



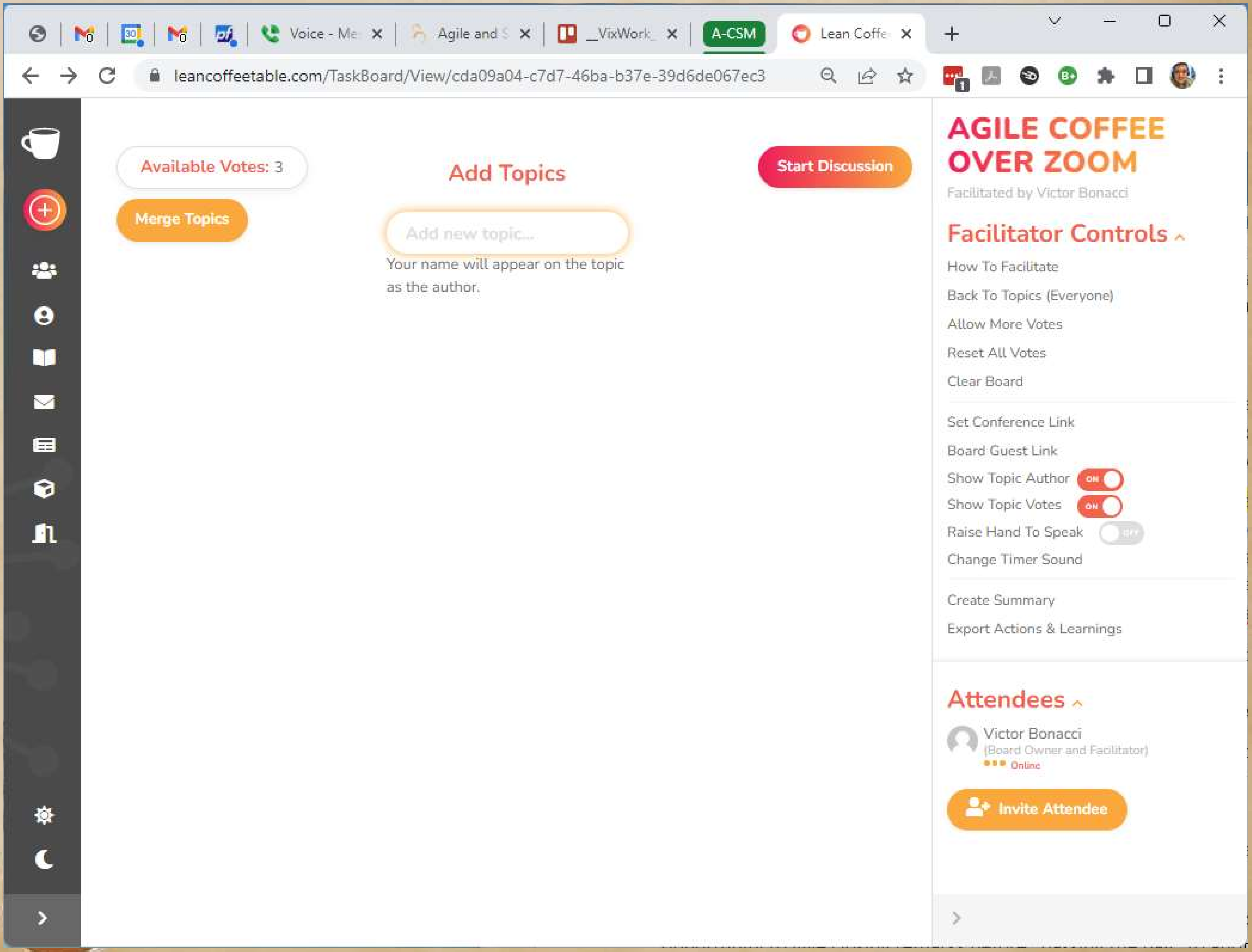
Gallery Walk

Parking Lot

Lean Coffee

create a simple kanban board

TO DISCUSS
DISCUSSING
DISCUSSED


Voice - Me
Agile and S
VixWork
A-CSM
Lean Coffe
leancoffeetable.com/TaskBoard/View/cda09a04-c7d7-46ba-b37e-39d6de067ec3

Available Votes: 3

Merge Topics

Add Topics

Add new topic...

Your name will appear on the topic as the author.

Start Discussion

AGILE COFFEE OVER ZOOM

Facilitated by Victor Bonacci

Facilitator Controls

How To Facilitate

Back To Topics (Everyone)

Allow More Votes

Reset All Votes

Clear Board

Set Conference Link

Board Guest Link

Show Topic Author ON

Show Topic Votes ON

Raise Hand To Speak OFF

Change Timer Sound

Create Summary

Export Actions & Learnings

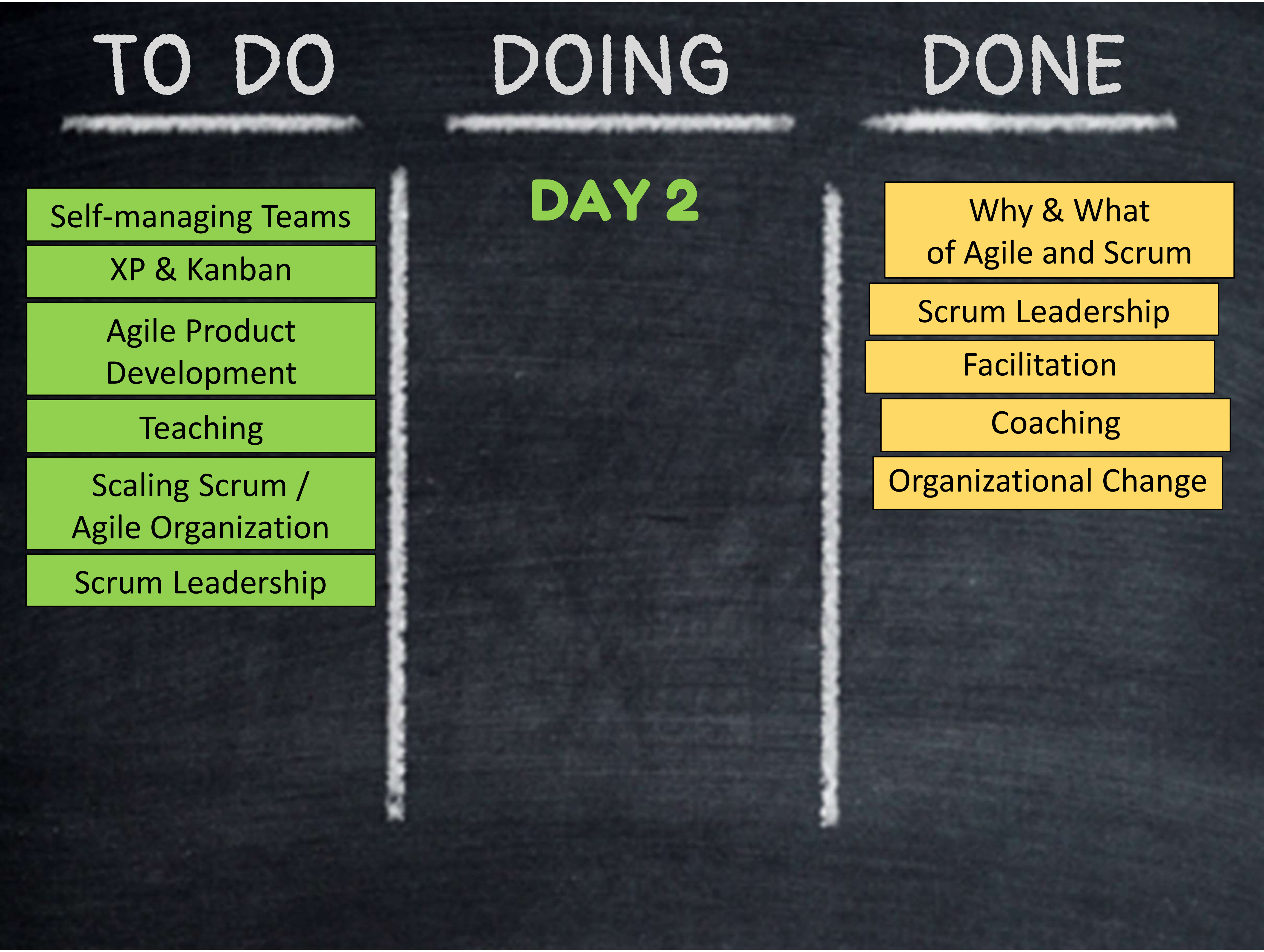
Attendees

Victor Bonacci
(Board Owner and Facilitator)
Online

Invite Attendee

LeanCoffee.org

2009 - Benson and Lightsmith - LeanCoffee.org



TO DO

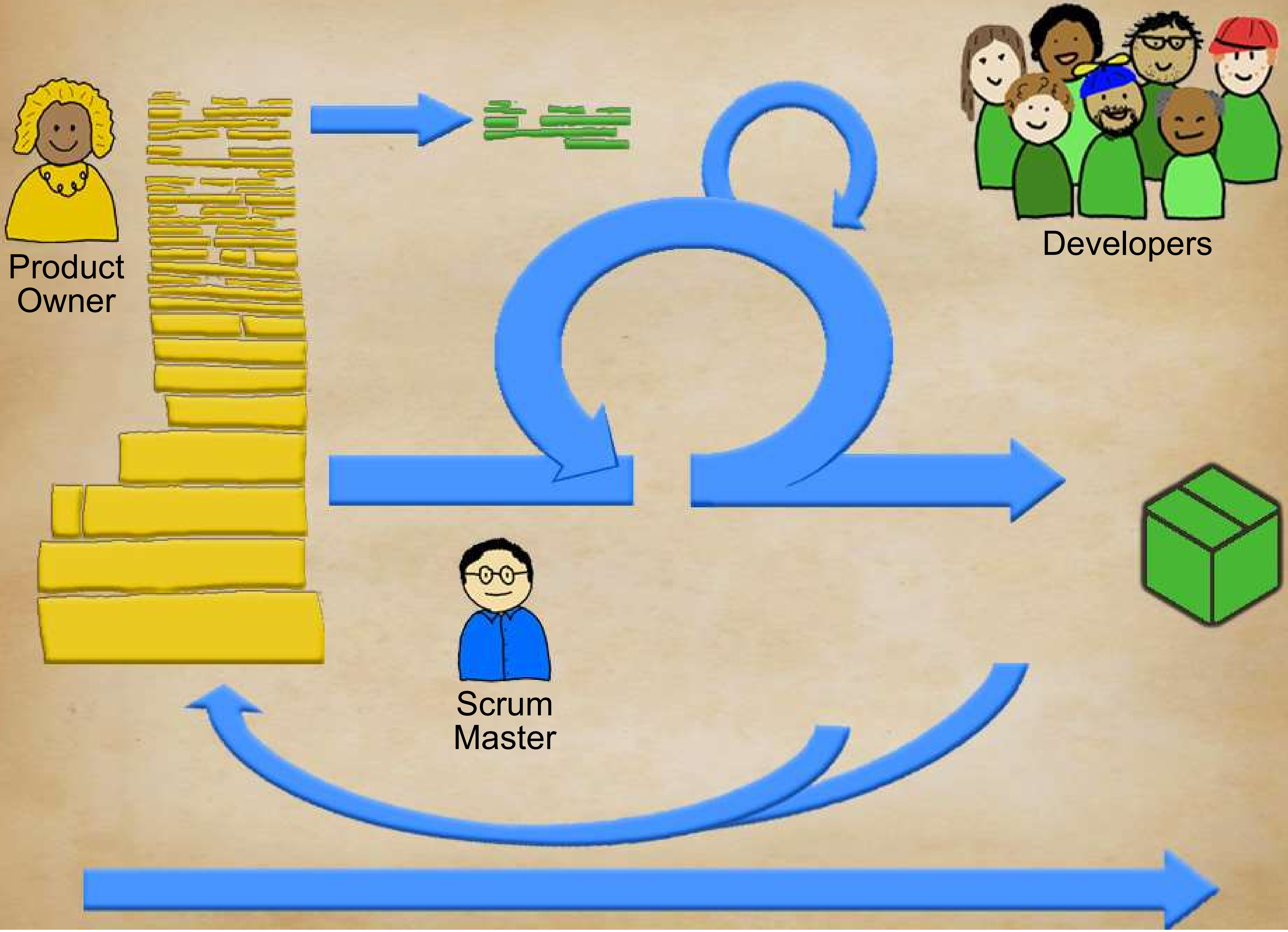
- Self-managing Teams
- XP & Kanban
- Agile Product Development
- Teaching
- Scaling Scrum / Agile Organization
- Scrum Leadership

DOING

DAY 2

DONE

- Why & What of Agile and Scrum
- Scrum Leadership
- Facilitation
- Coaching
- Organizational Change

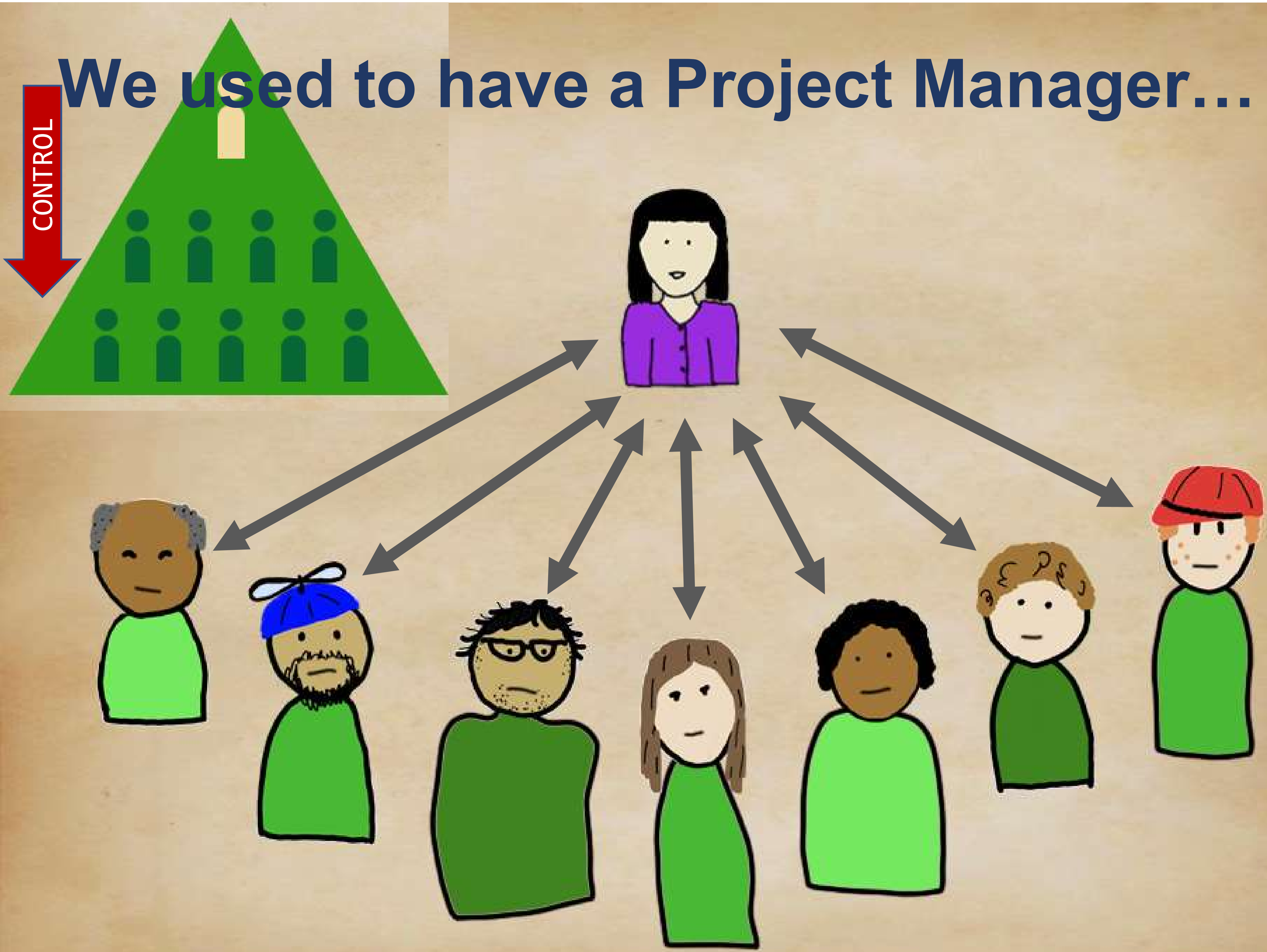

Product Owner
Developers
Scrum Master

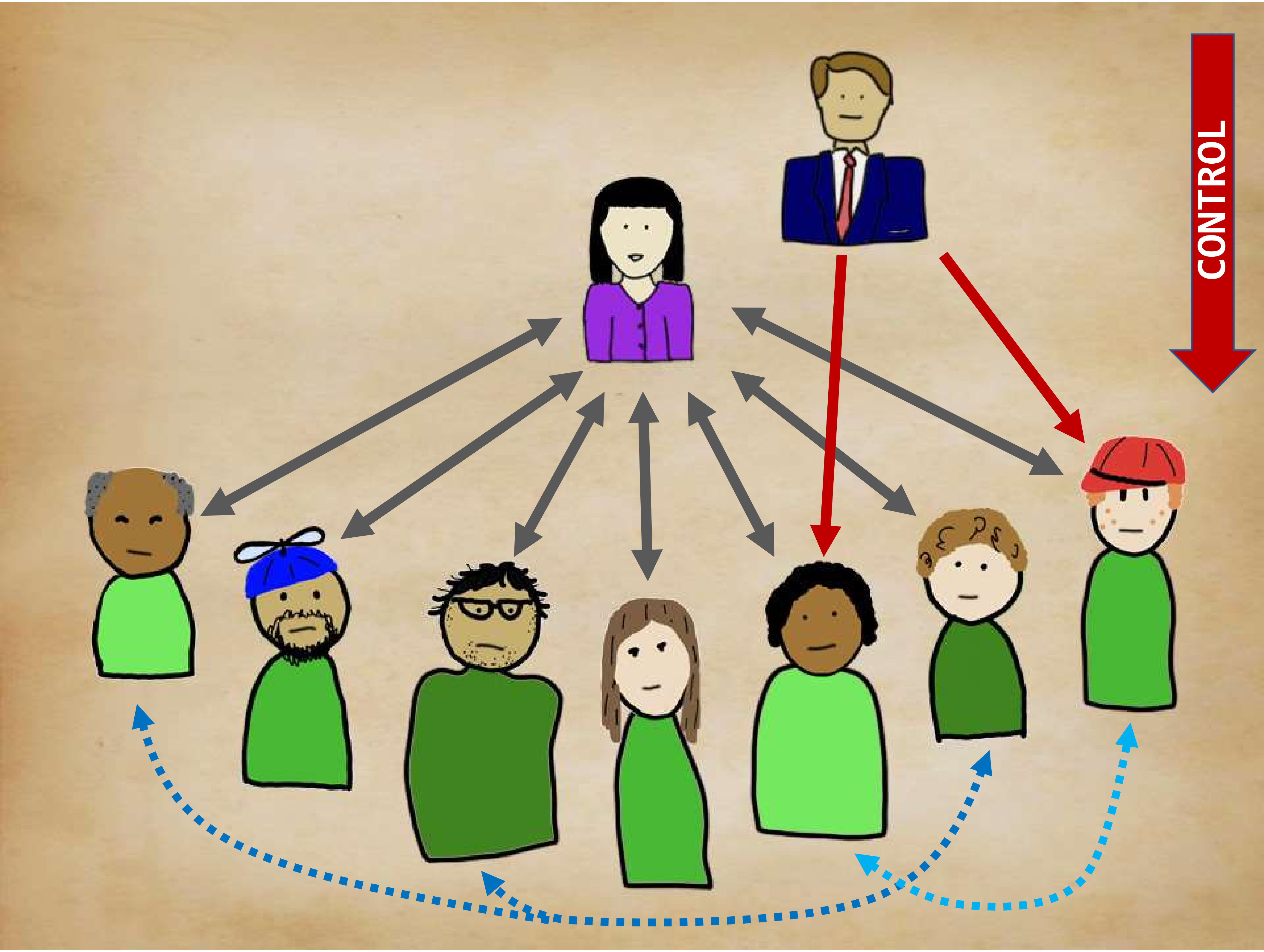
One Team, Focused on One Product

The goal was to eliminate the concept of a separate team within a team that has led to "*proxy*" or "*us and them*" behavior between the PO and Dev Team.

There is now just ***one Scrum Team*** focused on the same objective, with three different sets of *accountabilities*:
PO, SM, & Developers

We used to have a Project Manager…

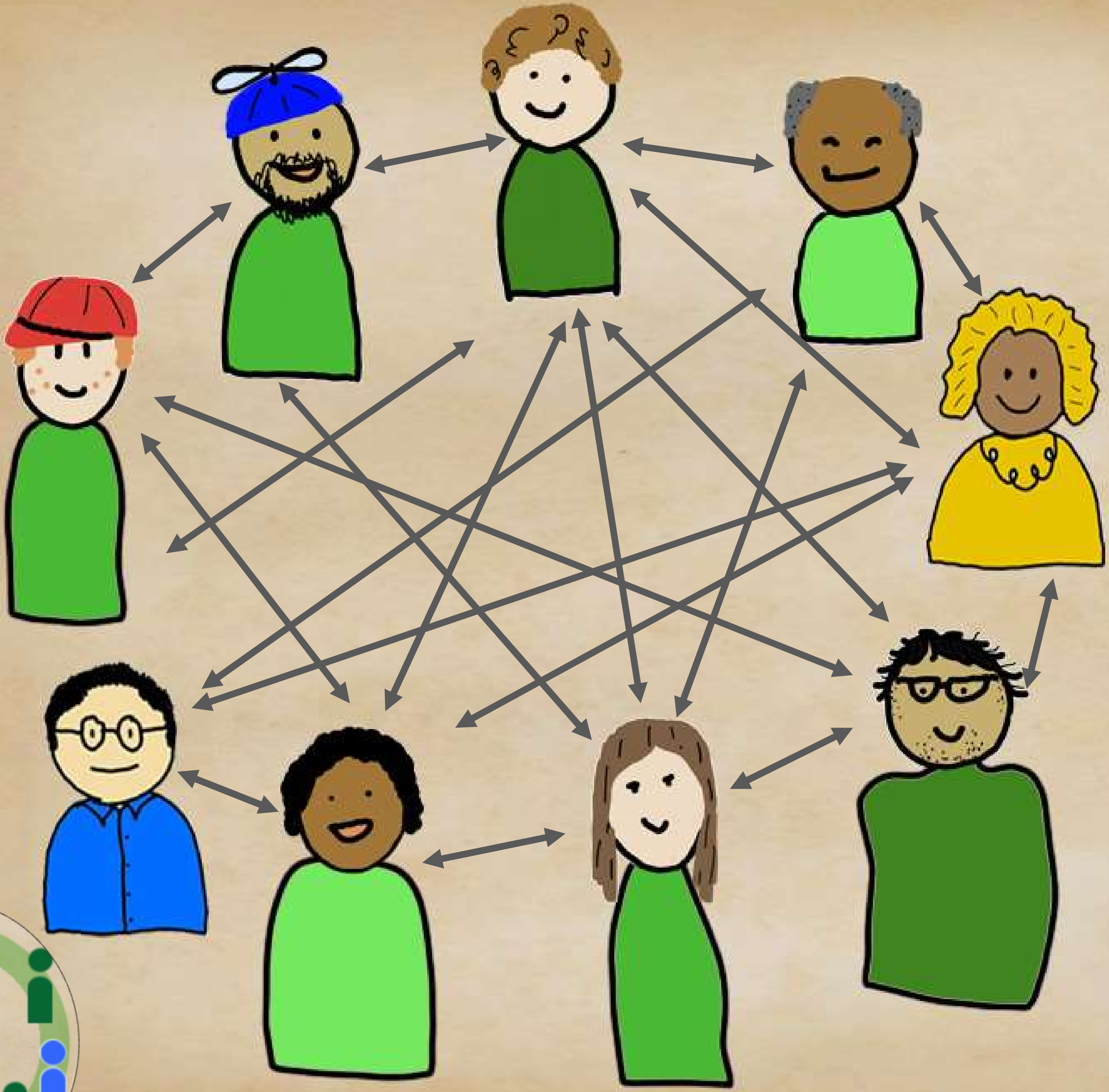



CONTROL

a Team, or a collection of individuals?

A team may be people doing something together, but the “*something*” part isn’t what makes a team - the “***together***” part is.

- Work is a bonding activity
- The obstacles are *grist for the mill* of change
- Absence of norms, working agreements or rules of engagement prevents teaming



Self-Managing over Self-Organizing

Previous Scrum Guides referred to
Development Teams as self-organizing,
choosing who and how to do work.

With more of a focus on the Scrum Team,
the 2020 version emphasizes
a self-managing Scrum Team,
choosing ***who***, ***how***, and ***what*** to work on.

Scrum Team

- self-managing
- "They internally decide who does what, when, and how."
- "Adaptation becomes more difficult when the people involved are not empowered or self-managing."

What are some attributes of a self-managing Scrum Team?

The Breakout Bunch

What are some attributes of a self-managing Scrum Team?

What are some techniques to improve the team’s ability to manage themselves?

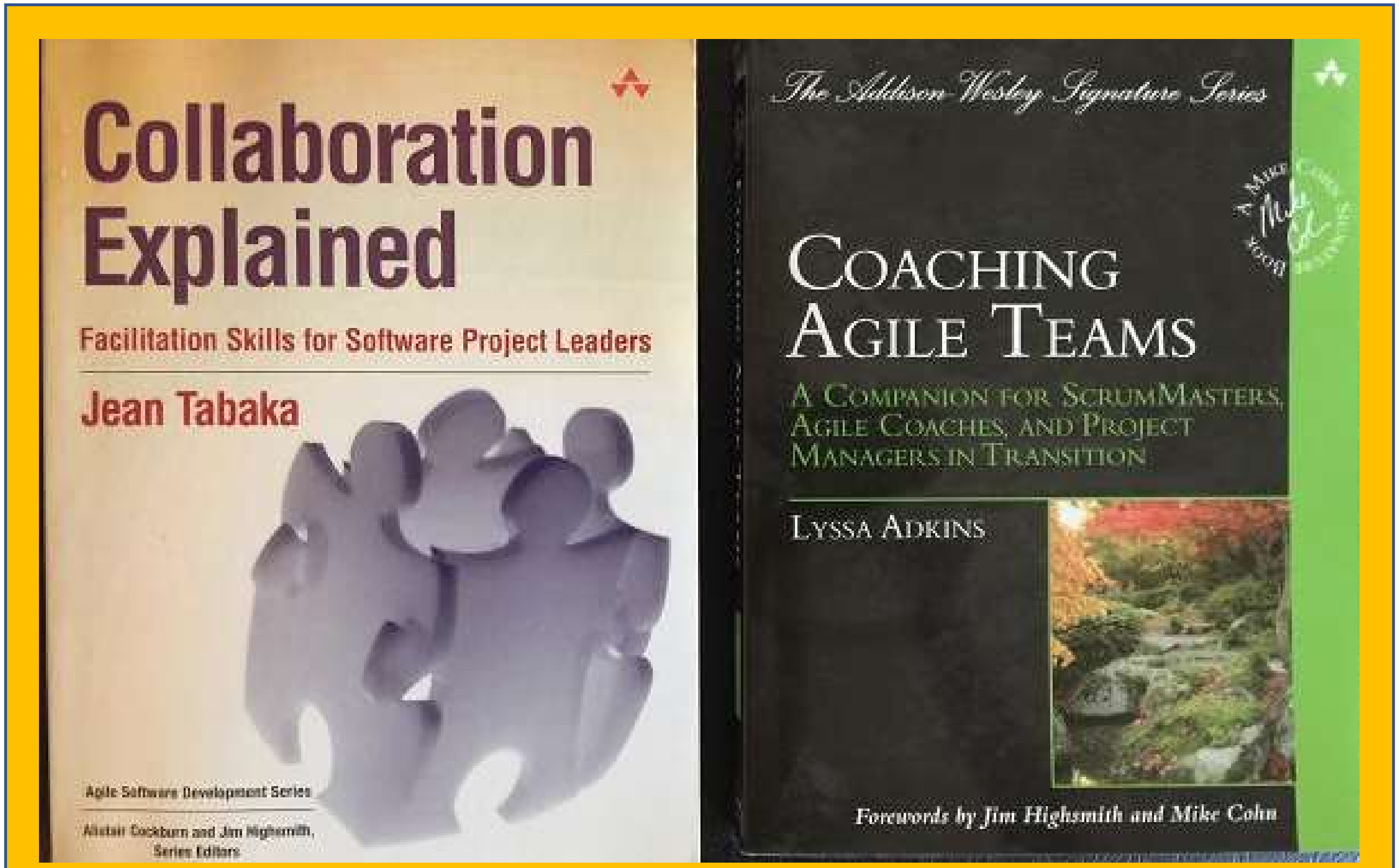
(Remember to Stretch)

Assignment #4:

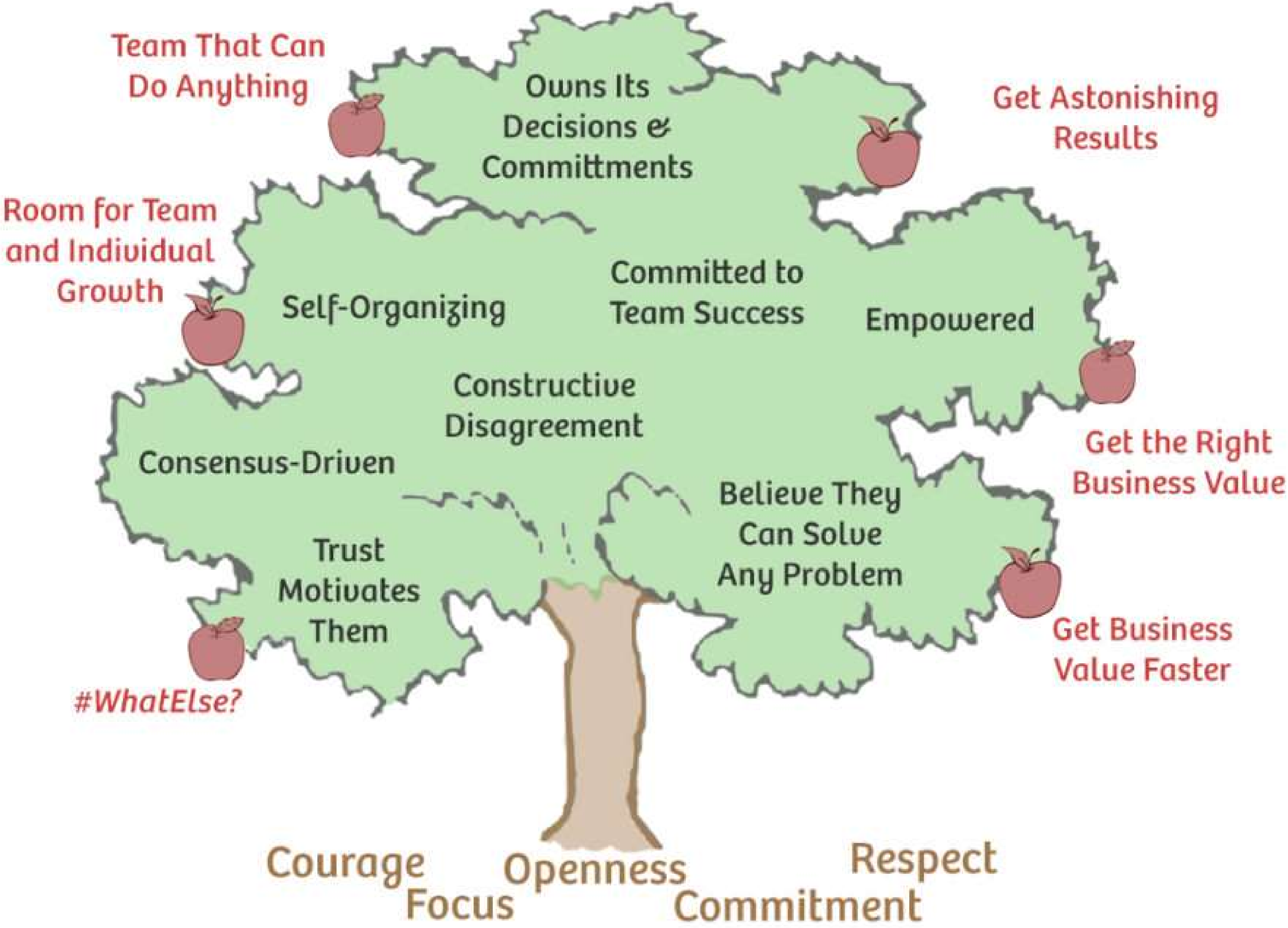
When you get back to a work environment, apply a technique to improve the team's ability to manage themselves.

Document this (and your reactions, learnings) in your personal journal.

Expect High Performance


Collaboration Explained
Facilitation Skills for Software Project Leaders
Jean Tabaka
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors
The Addison-Wesley Signature Series
A Mike Cohn Signature Book
Coaching Agile Teams
A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition
Lyssa Adkins
Forewords by Jim Highsmith and Mike Cohn

RECOMMENDED READING



Team That Can Do Anything

Owns Its Decisions & Committments

Get Astonishing Results

Room for Team and Individual Growth

Self-Organizing

Committed to Team Success

Empowered

Constructive Disagreement

Consensus-Driven

Get the Right Business Value

Believe They Can Solve Any Problem

Trust Motivates Them

Get Business Value Faster

#WhatElse?

Courage

Focus

Openness

Commitment

Respect

Coaching Agile Teams – Adkins 2010

OPTIONAL

Watch the two recommended videos (or find other similar videos or podcasts) on your own time.

Reflect on these and document your reactions in your personal journal.

Trusting Teams | Simon Sinek

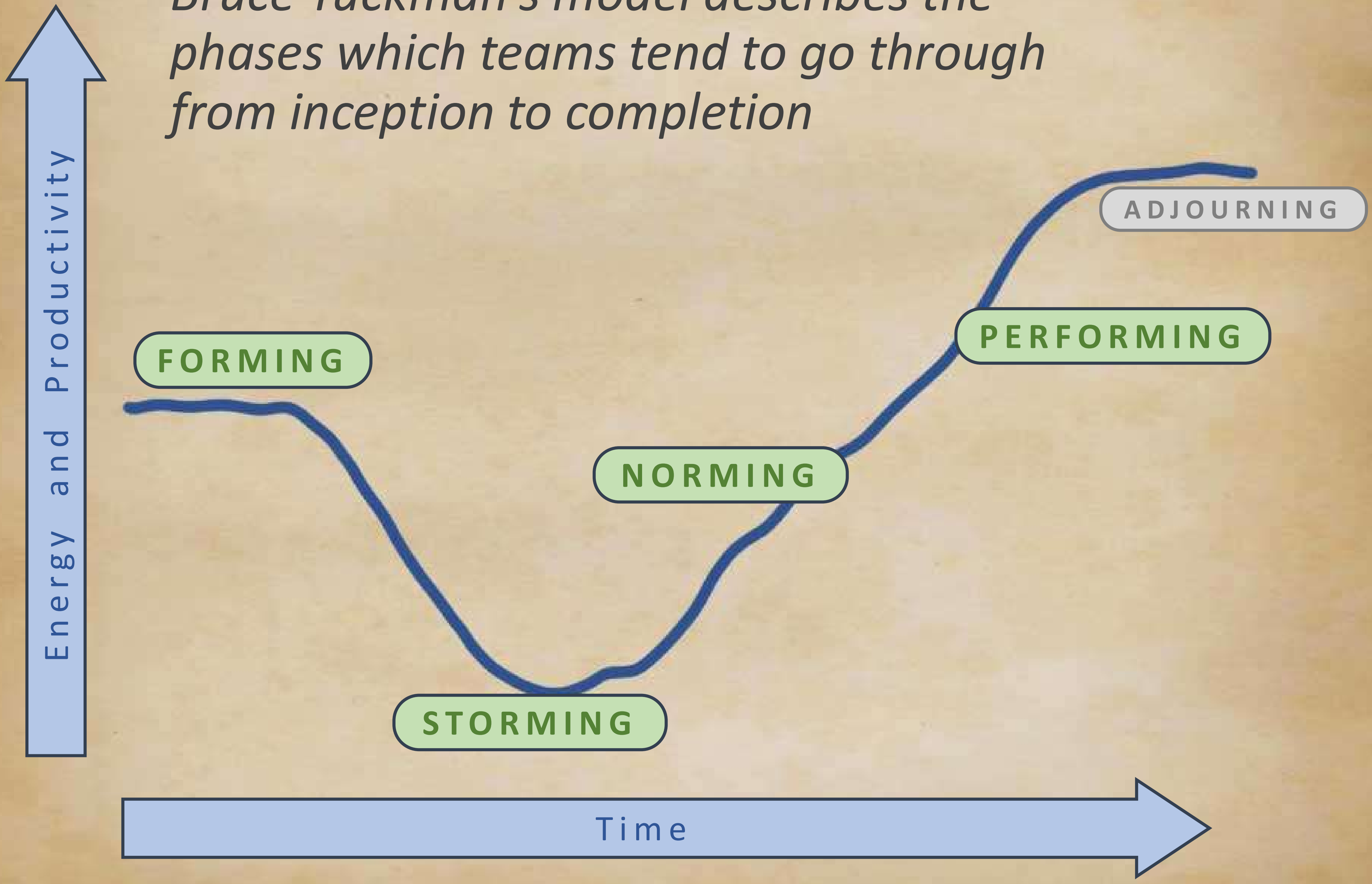
09:16

How to create a high performance culture | Andrew Stillitoe

06:33

Tuckman's **Team Development Model**

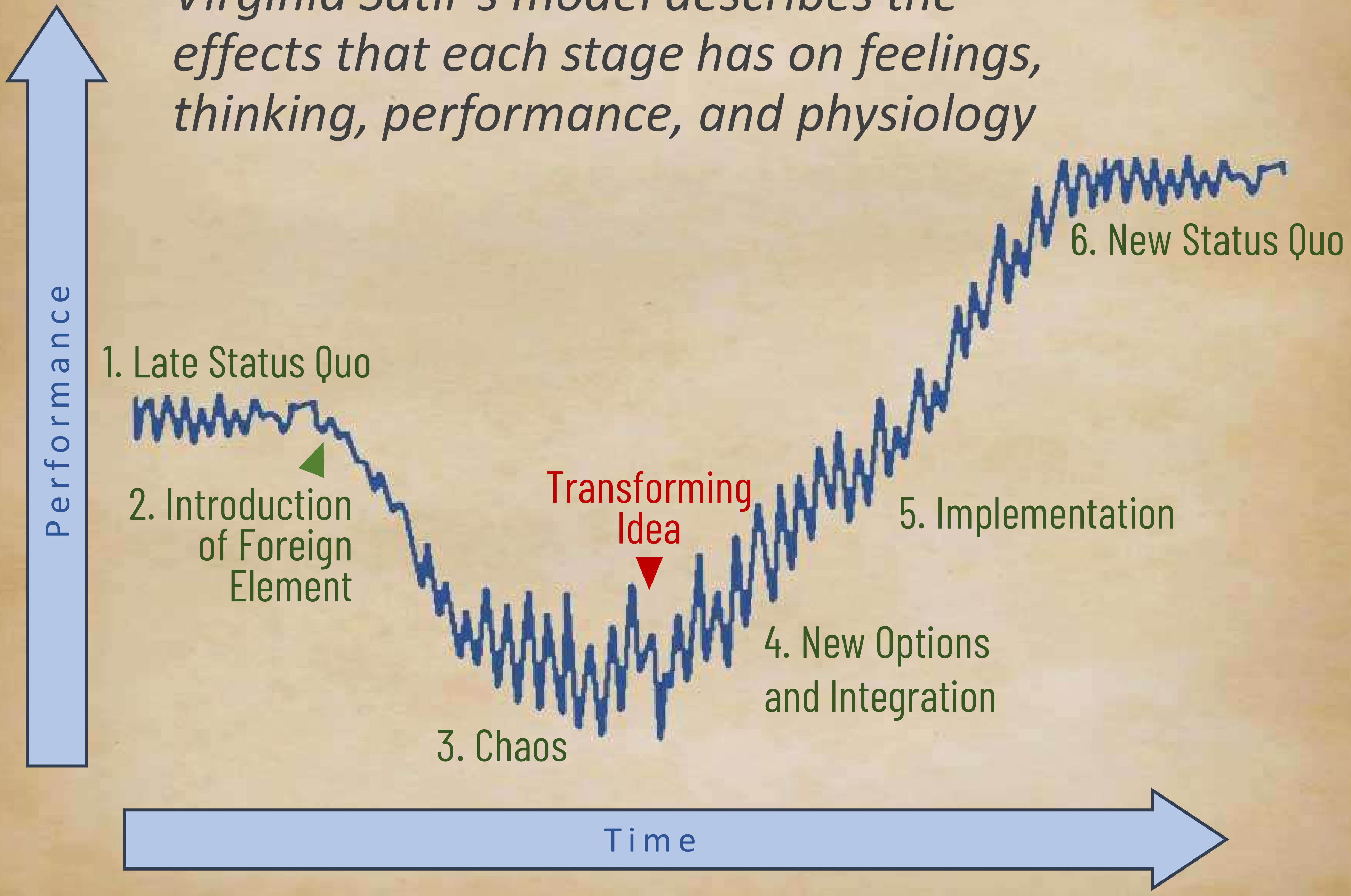
Bruce Tuckman's model describes the phases which teams tend to go through from inception to completion



Team Development Model - Bruce Tuckman (1965)

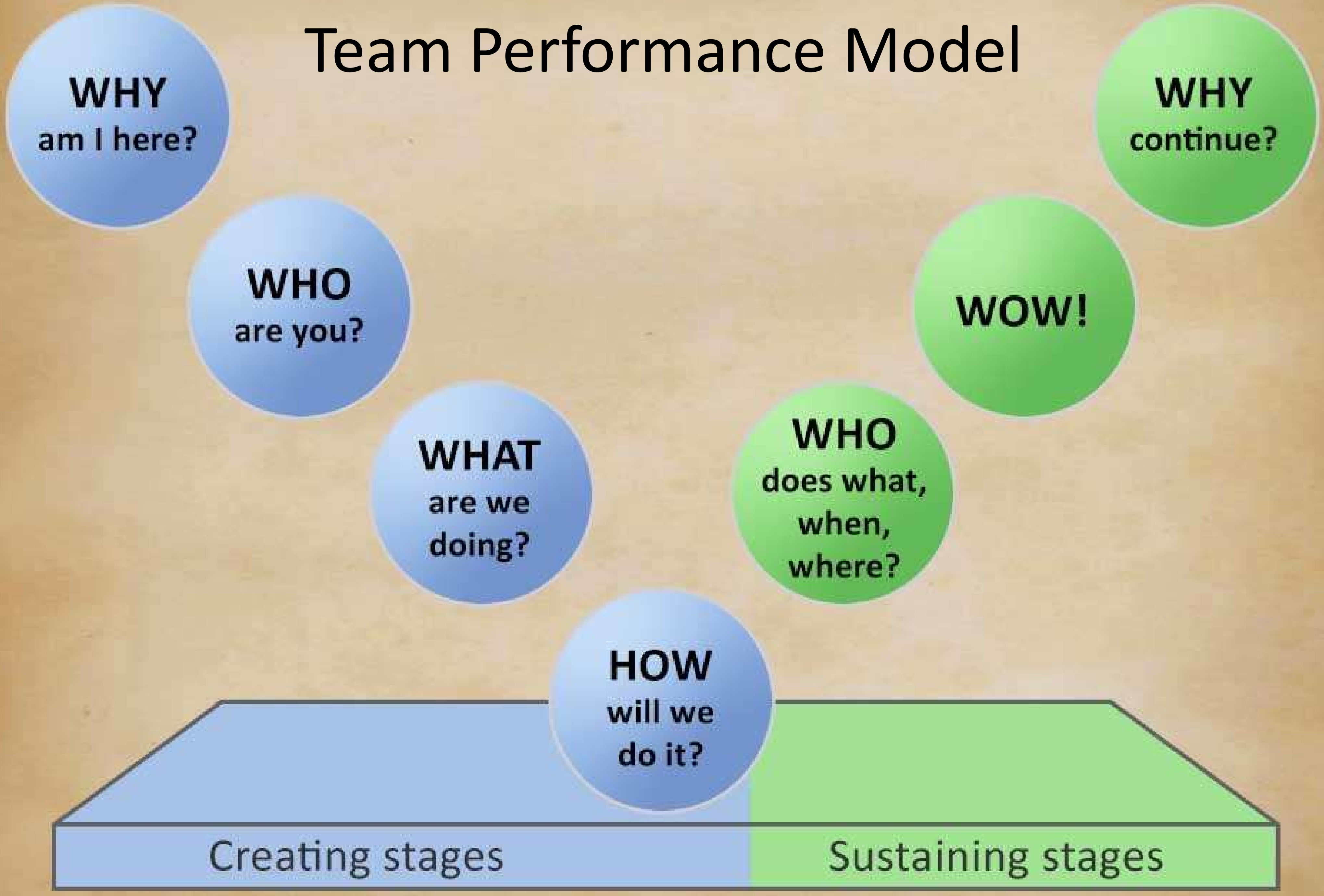
Satir Change Model

Virginia Satir's model describes the effects that each stage has on feelings, thinking, performance, and physiology

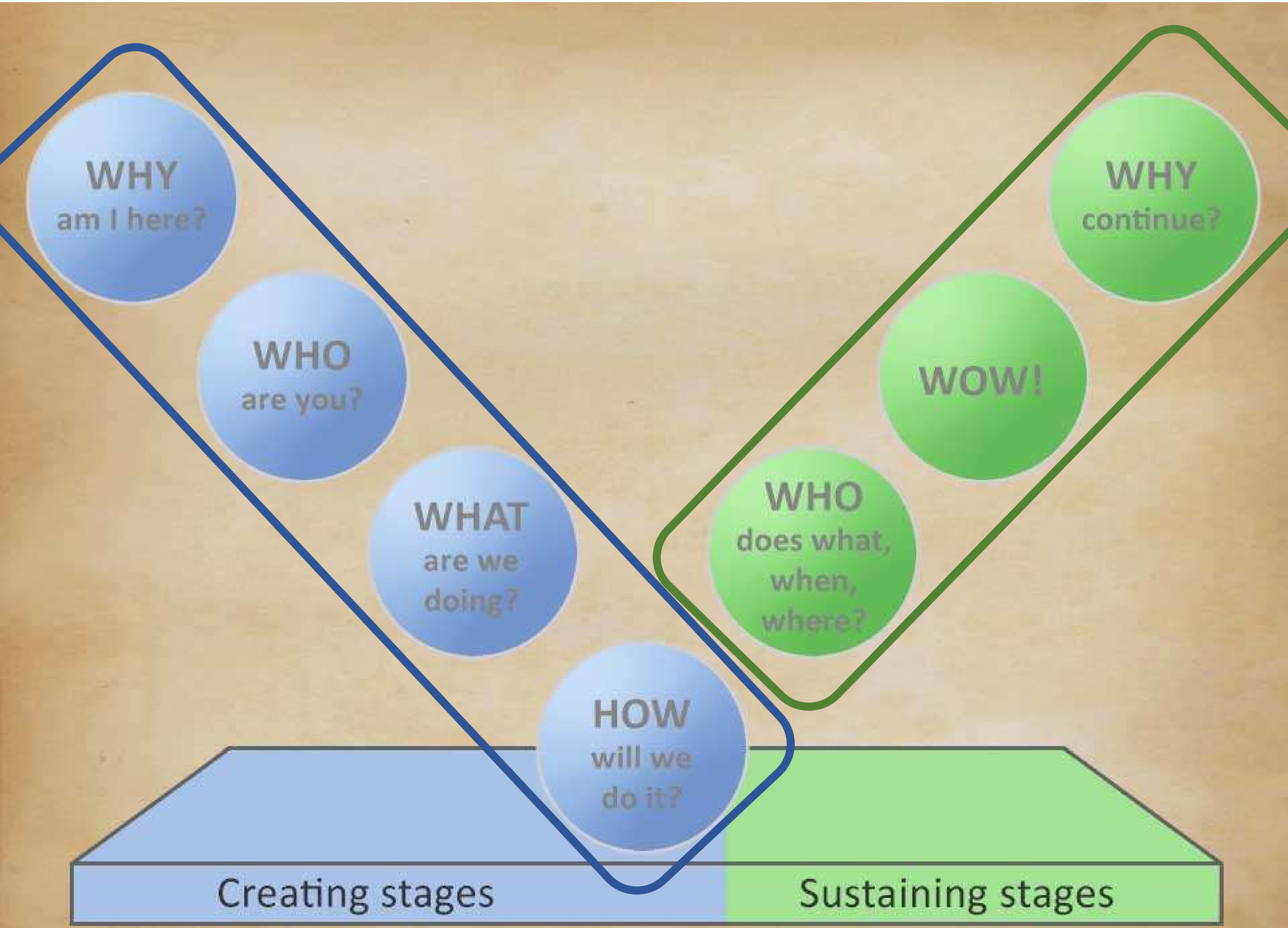


Satir's Stages of Change – from The Satir Model; Virginia Satir, et al. (1991)

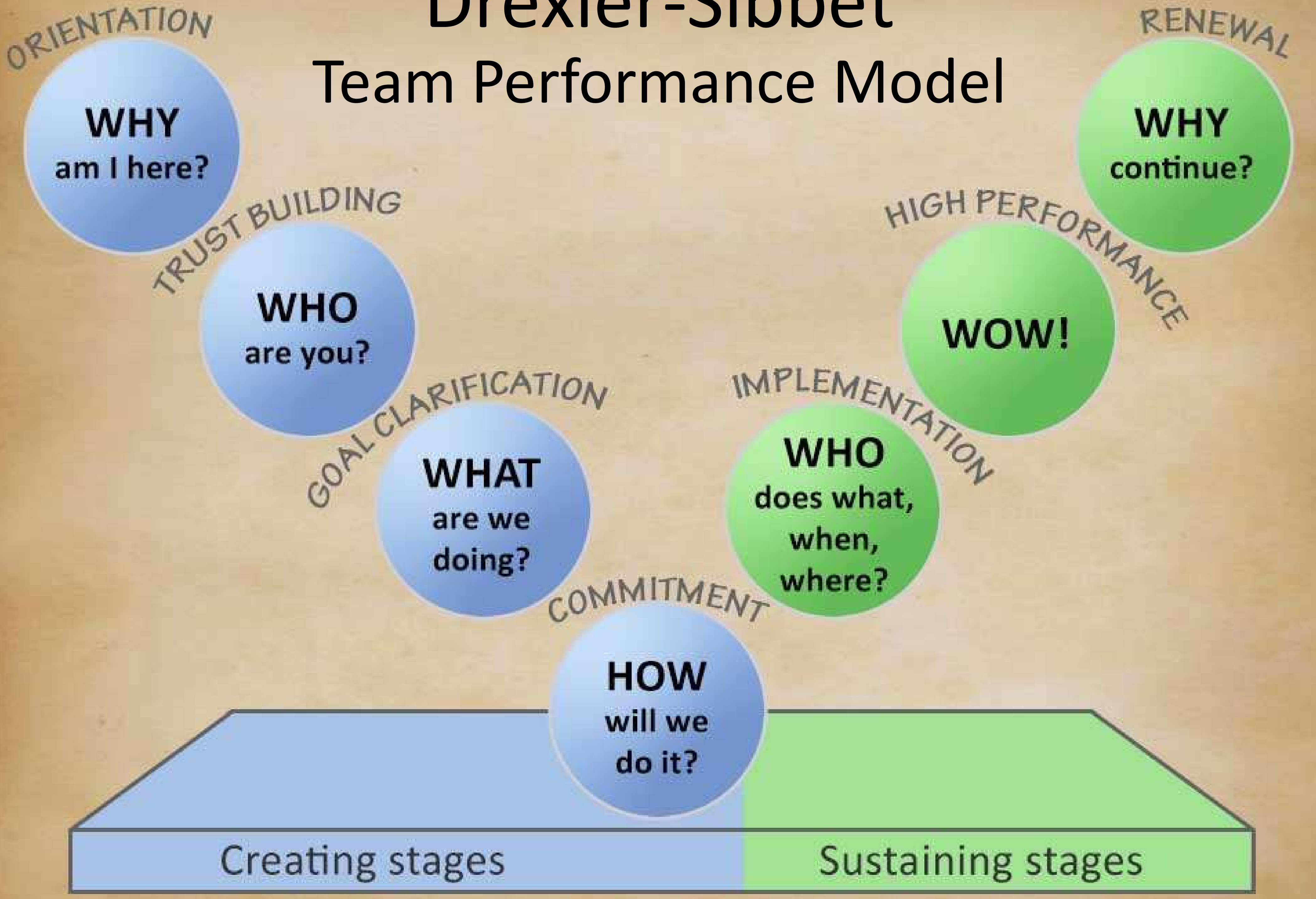
Drexler-Sibbet
Team Performance Model



Drexler/Sibbet Team Performance Model from The Grove Consultants International


WHY
am I here?
WHO
are you?
WHAT
are we
doing?
HOW
will we
do it?
WHO
does what,
when,
where?
WOW!
WHY
continue?
Creating stages
Sustaining stages

Drexler-Sibbet Team Performance Model



Grove website: https://grovetools-inc.com/collections/team-improvement

TO DO	DOING	DONE
Agile Product Development	XP & Kanban	Why & What of Agile and Scrum
Teaching		Scrum Leadership
Scaling Scrum / Agile Organization		Facilitation
Scrum Leadership		Coaching
		Organizational Change
		Self-managing Teams

The New New Product Development Game

Design Patterns

Manufacturing Industry in Japan

Toyota Production System

Scrum

XP

Lean

Lean Software Development

Agile

Kanban

Lean Startup

Design Sprints

Four Steps to the Epiphany

Startup

Snowbird, Utah
February 2001

Purpose: uncovering

We are ~~figuring out~~ better ways of developing sofware by doing it and helping others to do it.

interactions

1) individuals and ~~social networks~~ over process + tools

4 ~~2)~~ responding to change over following ~~a~~ plan

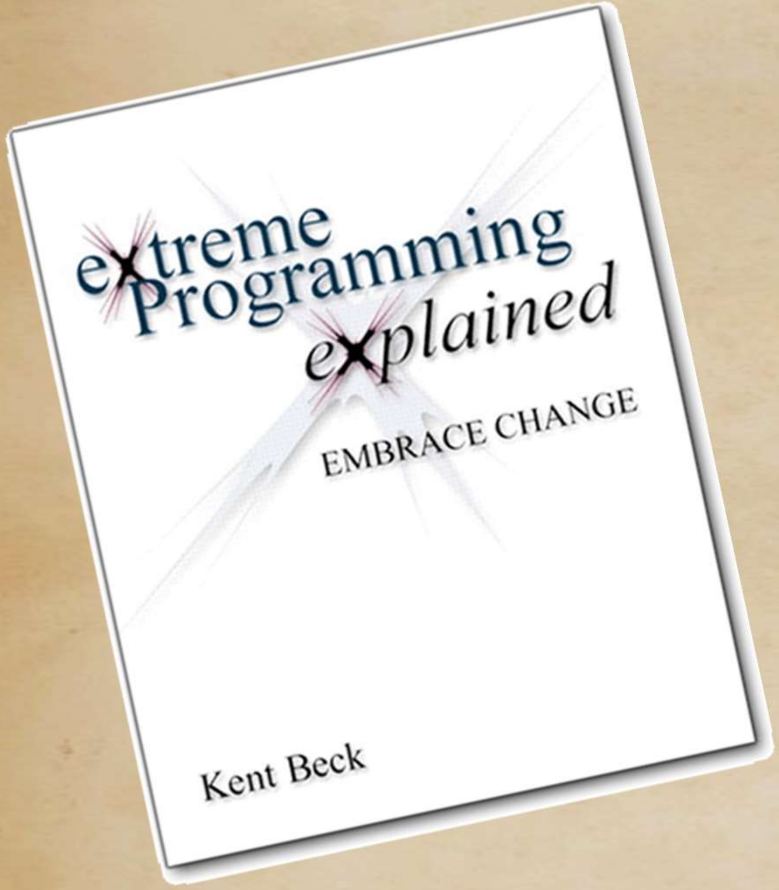
3 ~~3)~~ customer colloboration over contract negotiation

Writing Effective Use Cases
Alistair Cockburn
Robert C. Martin Series
PRENTICE HALL
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
DSDM
Business Focused Development
Second Edition
DSDM Consortium
Jennifer Stapleton
aptive
tware
OBJECT-ORIENTED METHODS
A FOUNDATION
James Martin
and
James J. Odell
20th Anniversary Edition
The Pragmatic Programmer
your journey to mastery
David Thomas
Andrew Hunt
Foreword by Saron Yitbarek
Framewo
for Bus
Centred
DSDM consortium
viving
ect-Oriented
jects
Foreword by
Rebecca Wirfs-Brock
Alistair Cockburn

RECO DING

The 1990's
World Cup USA 94
HARRY POTTER
J.K. ROWLING
HOGWARTS EXPRESS
intel pentium
eBay
Seinfeld
George
Yahoo!
2370
62
ROSS PEROT
The 1

eXtreme Programming (XP)



- Pair-programming
- Test-Driven Design (TDD)
- Unit Test first
- Continuous Integration
- Whole team
- Simple designs
- Small releases
- No overtime
- Sustainable Pace

XP: Paired Programming

- Paired programming is a technique in which two team members work together at one computer - one person is typically writing code while the other is performing a real-time code review
- The people working in the pair can switch off as needed
- This does not mean 2 people doing the work of 1…pairs typically complete work faster than one person assigned to do the same task alone
- Errors are usually caught earlier in this process reducing the overall defect rate and improving the quality of the application

extremeprogramming.org

XP: Test Driven Development (TDD)

- Test Driven Development, or TDD, is a technique in which the test case is written first - usually one that fails that helps define what the function or requirement should do
- Code is written to pass the test and improved to what the acceptance standards are
- This technique also helps to “move quality up in the process” and prevent defects as opposed to hunting for them later in the process

extremeprogramming.org

XP: Refactoring

"Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure."

Martin Fowler; Refactoring: Improving the Design of Existing Code

- Refactoring is a disciplined technique for improving or restructuring existing code
- These are typically small changes that do not modify requirements
- Refactoring is typically done to improve code readability, reduce complexity and to improve maintainability of source code

extremeprogramming.org

XP: Continuous Integration (CI)

- Traditional software development practices involve testing for quality after code is developed and integrated in later stages in the development lifecycle
- Continuous integration involves frequently integrating new code with existing code to check for errors
- This moves quality up in the process and helps to prevent defects as opposed to searching for them later
- Checking in code once a day is not CI

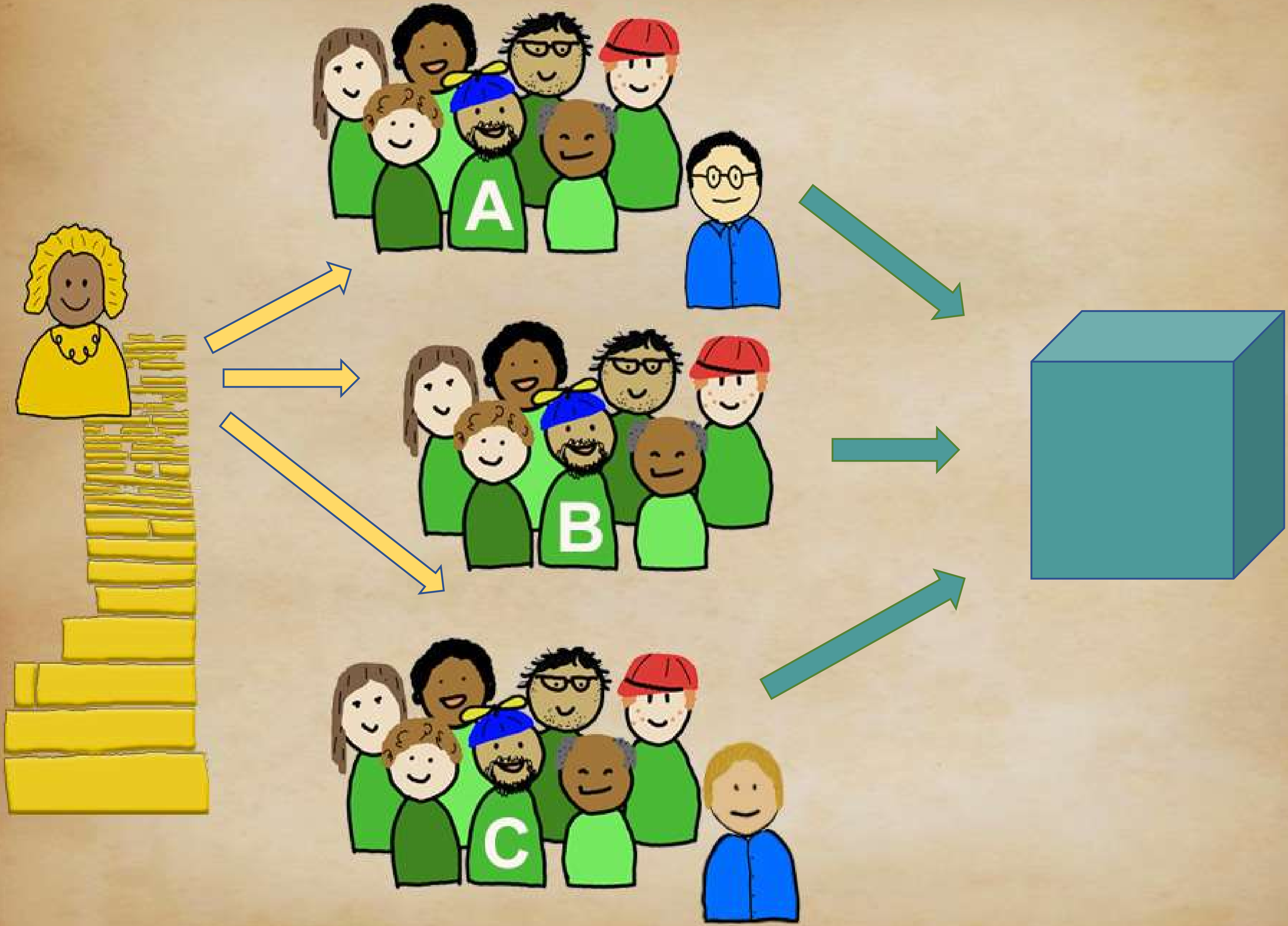
extremeprogramming.org

XP: Collective Ownership

- No one programmer “owns” the code – it’s a shared responsibility to change functionality, fix any defects, improve design or to refactor
- The “test first” practice works well and hand-in-hand with this practice because instead of relying on a person to test the code, the automated testing is what prevents defects
- This is more reliable than one person since we can’t control when someone may leave the organization, be out sick, etc.

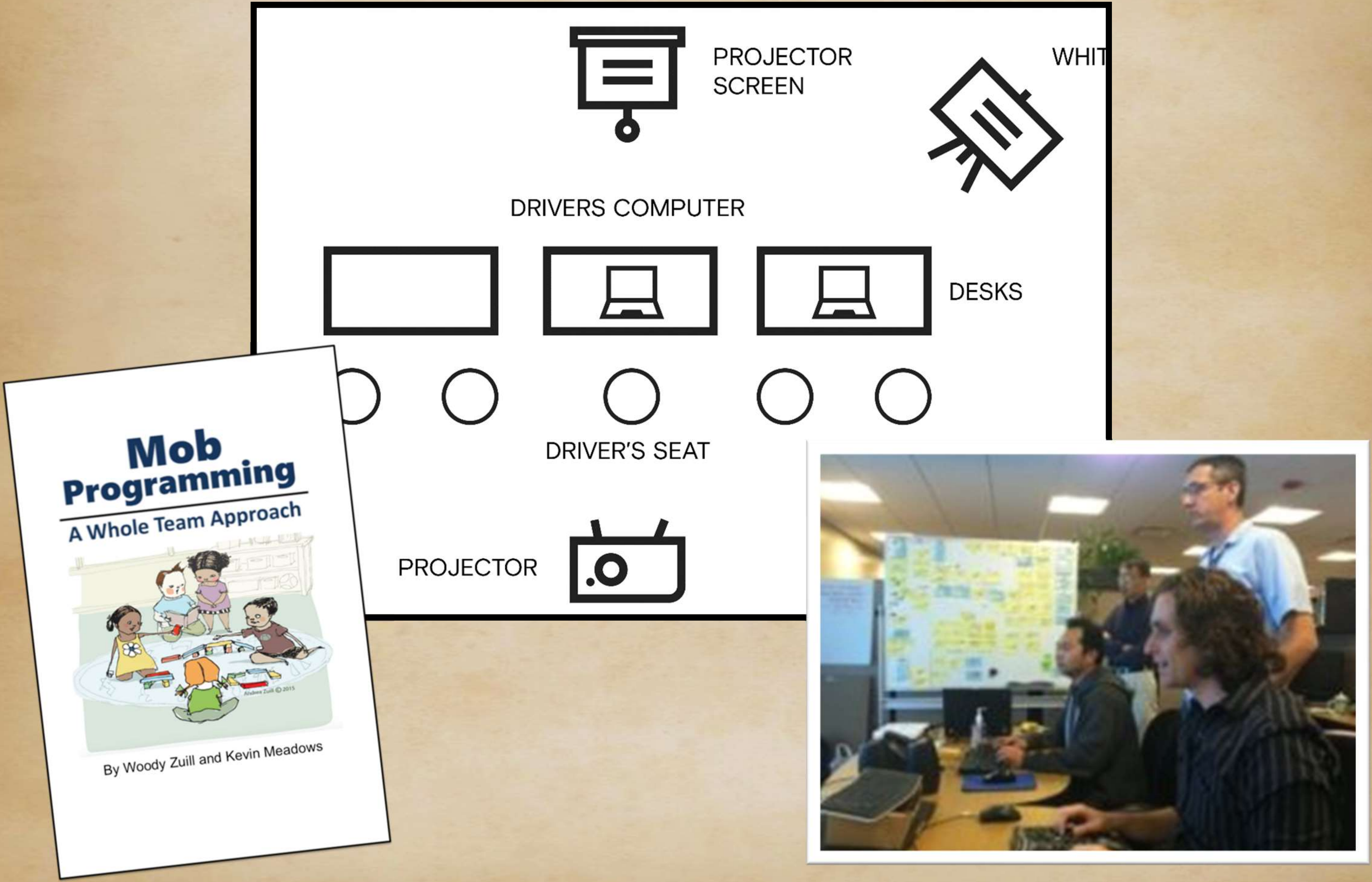
extremeprogramming.org

Explain at least two ways development practices may impact the Scrum Team’s ability to deliver valuable, usable Increments each Sprint.


A
B
C

How could development practices be beneficial in a multiple team environment?

Mob Programming



Mob
Programming
A Whole Team Approach

By Woody Zuill and Kevin Meadows

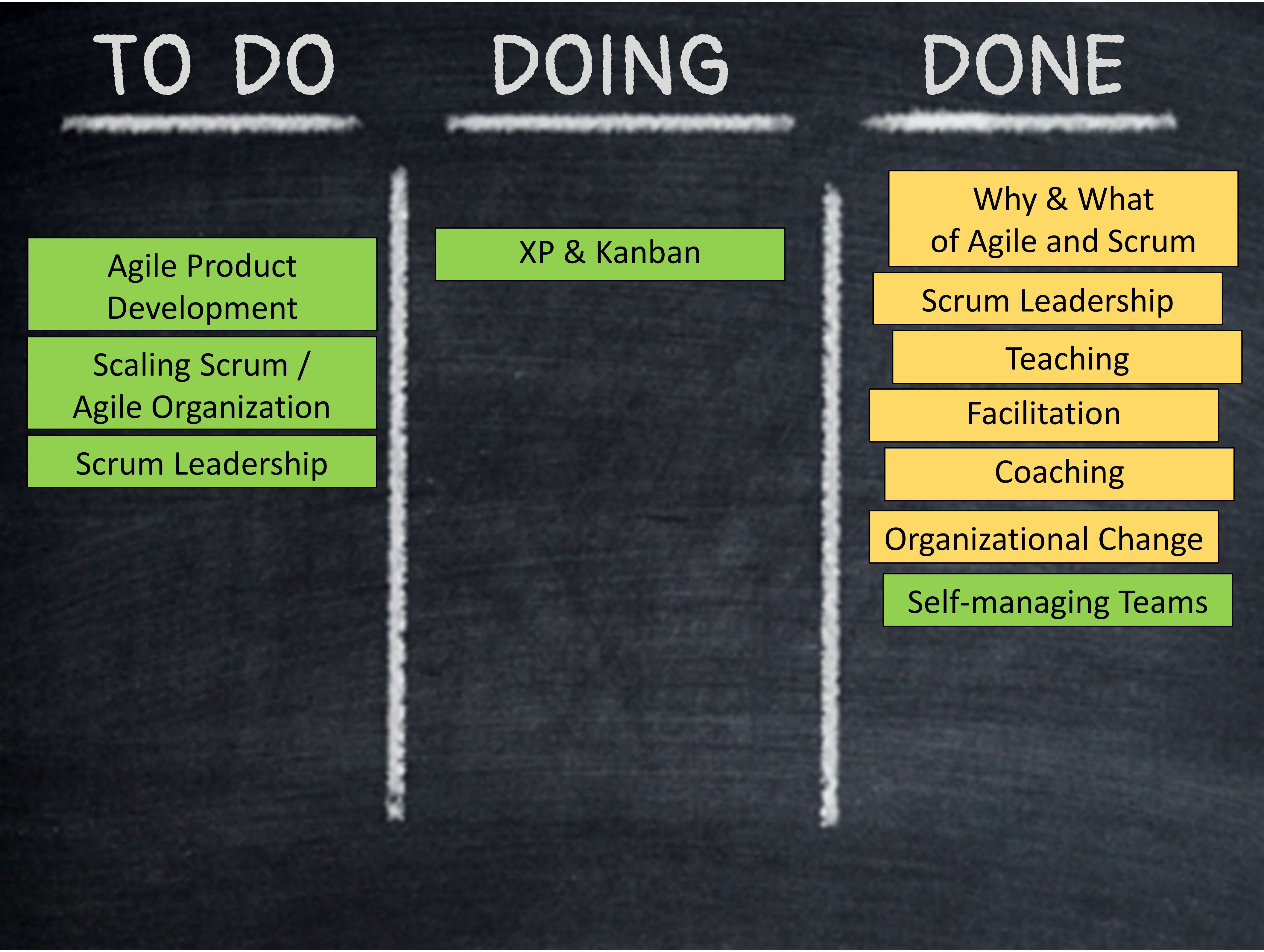
mobprogramming.org

A day of Mob Programming 2016 | Mob Programming

03:06

看板

Kanban



TO DO

- Agile Product Development
- Scaling Scrum / Agile Organization
- Scrum Leadership

DOING

- XP & Kanban

DONE

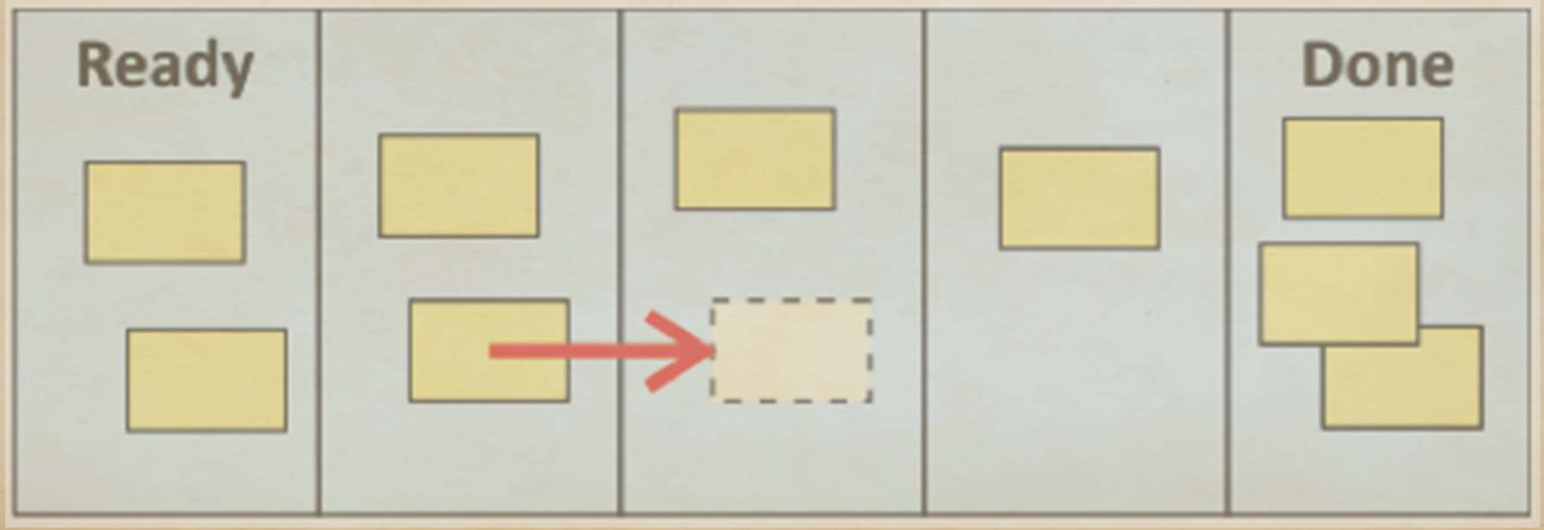
- Why & What of Agile and Scrum
- Scrum Leadership
- Teaching
- Facilitation
- Coaching
- Organizational Change
- Self-managing Teams

Kanban

- Based on Lean Manufacturing at Toyota
- Taiichi Ohno produced kanbans to control production between processes and implement Just In Time (JIT) manufacturing
- Kanban = “Signboard” in Japanese
- Kanban = “Signal Card” at Toyota

Kanban: What are the Rules?

- There are no prescribed roles in Kanban
- Start with the existing process, the existing roles or “what you do now”
- Agree to pursue evolutionary change
- Initially, respect current processes, roles and responsibilities
- Encourage acts of leadership at every level



Kanban Practices

Visualize the Work

Make Policies Explicit

Limit WIP

WiP = work in progress

Implement Feedback Loops

Manage Flow

Improve Collaboratively, Evolve Experimentally

Visualize the Work

Unlike the construction of physical products, in the knowledge work processes the inventory is not visible, it does not take up space and it is not accounted for. If we do not visualize the work in progress we will not be able to see the queues or the points of improvement of the system.

In addition to visualizing the work process and work items, there are many other things to visualize. We must visualize everything that helps us make decisions and visual signals and controls that indicate when we have to act or when there is a problem.

Limit Work in Progress (WIP)

- Limiting work in progress is key to implementing a pull system, since by limiting the capacity of a process we know immediately when there is free capacity to take more work from the previous process.

- By putting the focus on the work in progress we begin the change of mentality from resource efficiency, typical of the traditional company, to flow efficiency, where what matters to us is that the work flows as quickly as possible and not who is doing what.

- Implementing and respecting WiP limits changes a “push” system to a “pull” system, where new items cannot start until the job is completed. Too much partially complete work is a common disease that lengthens the time-to-market and is an impediment to the organization responding to changing circumstances.

Manage Flow

- The workflow in a Kanban System must maximize value delivery, minimize time-to-market and be as predictable as possible. Therefore, empirical control requires transparency, inspection and adaptation.

- Bottlenecks and blockages are particularly important and must be properly managed to improve predictability and system performance.

- Kanban practitioners are obsessed with flow. We want the work to flow as quickly as possible to the client with the highest quality and safety. This is a lean heritage. We do not manage people, we manage the system to allow a fast flow.

Make Policies Explicit

- Human beings tend to have many assumptions and we cannot let assumptions and opinions lead companies. It is also very common that there are discrepancies between company strategy and what is really happening. Therefore, we must ensure that all decisions, processes, criteria and data are explicit and visible to all.

- Explicit policies restrict action and result in emerging behaviors that can be improved through experiments. Policies must be simple, well defined, visible, always applied and easily modified by those who provide the service.

Implement Feedback Loops

- Feedback loops represent the pulse that keeps an organization alive and connected. They allow to periodically connect different levels of decision-making in the organization by exchanging information for continuous improvement.

- The Kanban method has seven Kanban Cadences (or Kanban Ceremonies). Cadences are the cyclical reviews that drive continuous improvement and the effective provision of services:

Improve Collaboratively, Evolve Experimentally

- Of the six Kanban practices, this is the one that receives the input of the other five, because after all Kanban is a method of continuous improvement.

- Kanban starts with the organization as it is now to seek continuous and incremental improvement and takes advantage of an evolutionary process to allow beneficial changes to occur within an organization, protecting it from extinction.

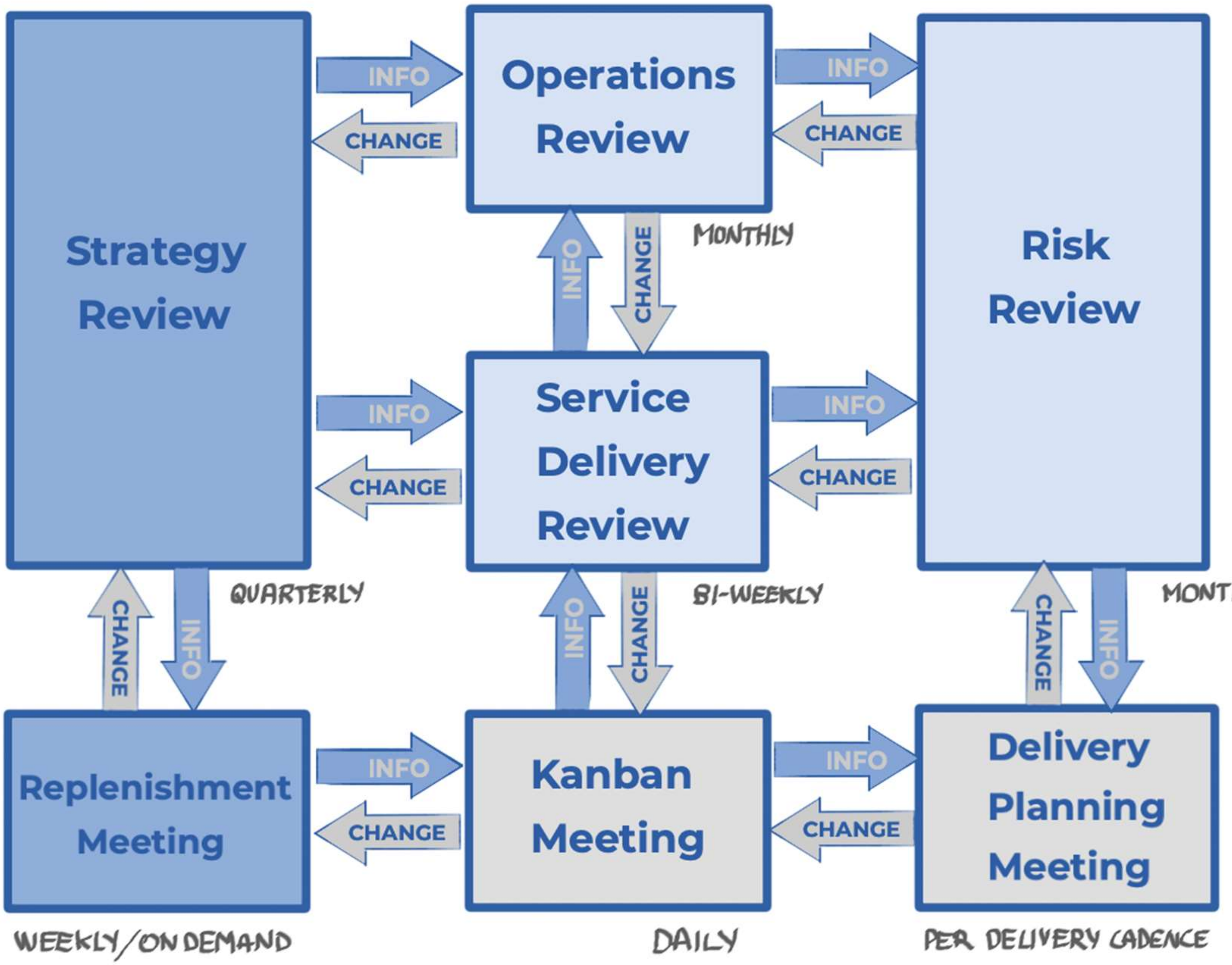
- The evolutionary process involves: dealing with differences; select to increase the value contributed to customers; maintain and amplify the useful change while rejecting or reversing the ineffective change.

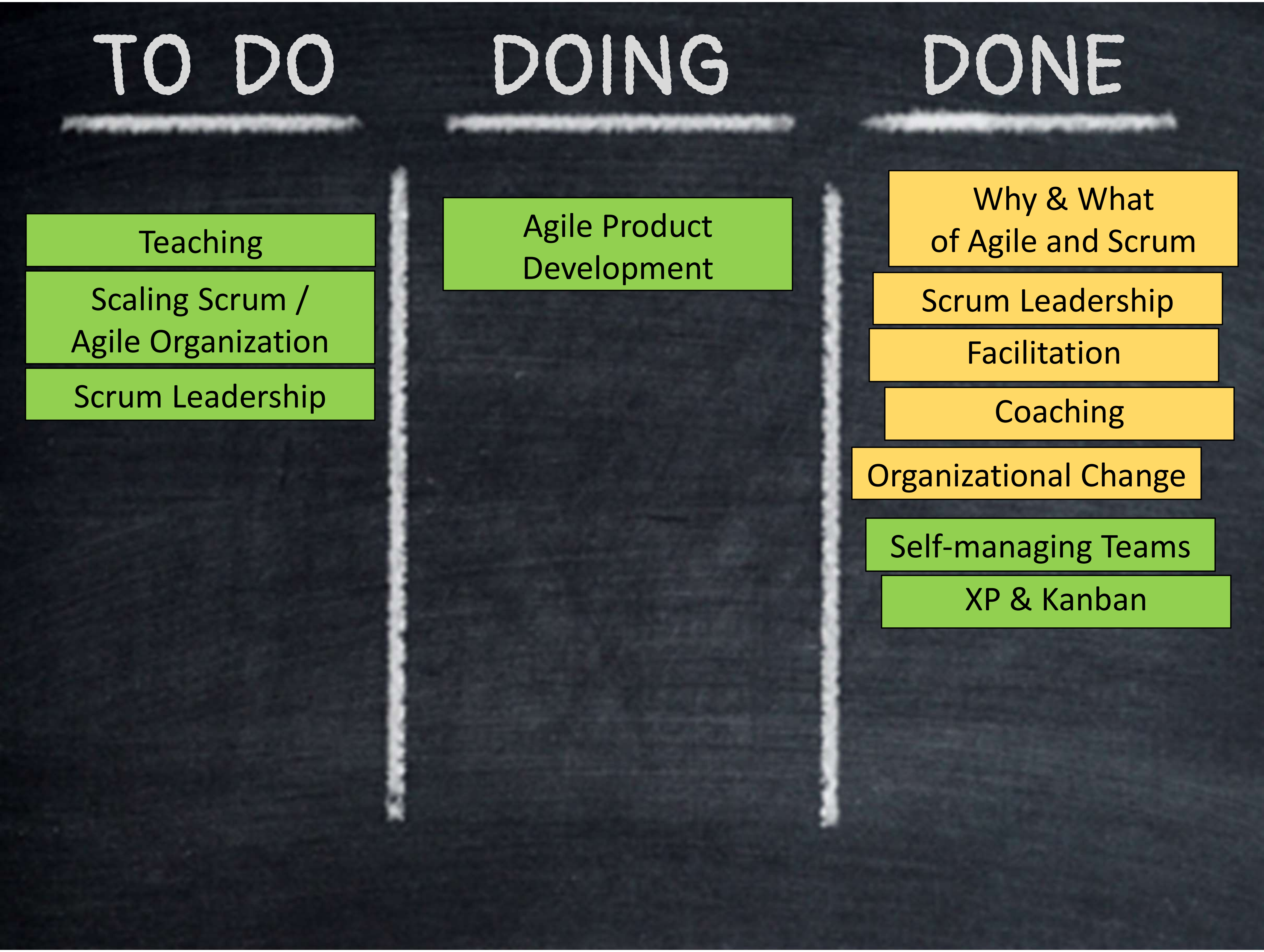
Kanban: Properties

- Visualize the flow: a common way to visualize the workflow is to use a card wall with cards and columns
- Limit Work In Progress (WIP): Limiting WIP implies that a pull system is implemented in parts or all of the workflow
- Improve Collaboratively, Evolve Experimentally: When teams have a shared understanding of theories about work, workflow, process, and risk, they are more likely to be able to build a shared comprehension of a problem and suggest improvement actions which can be agreed by consensus

Kanban: Properties

- Manage Flow: The flow of work through each state in the workflow should be monitored, measured and reported
- Make Process Policies Explicit: With an explicit understanding it is possible to move to a more rational, empirical, objective discussion of issues
- Implement Feedback Loops: The purpose of feedback loops is to be able to compare expected outcomes with actual outcomes and make adjustments.


Strategy Review
Operations Review
Risk Review
Service Delivery Review
Replenishment Meeting
Kanban Meeting
Delivery Planning Meeting
INFO
CHANGE
MONTHLY
QUARTERLY
BI-WEEKLY
MONTHLY
WEEKLY/ON DEMAND
DAILY
PER DELIVERY CADENCE

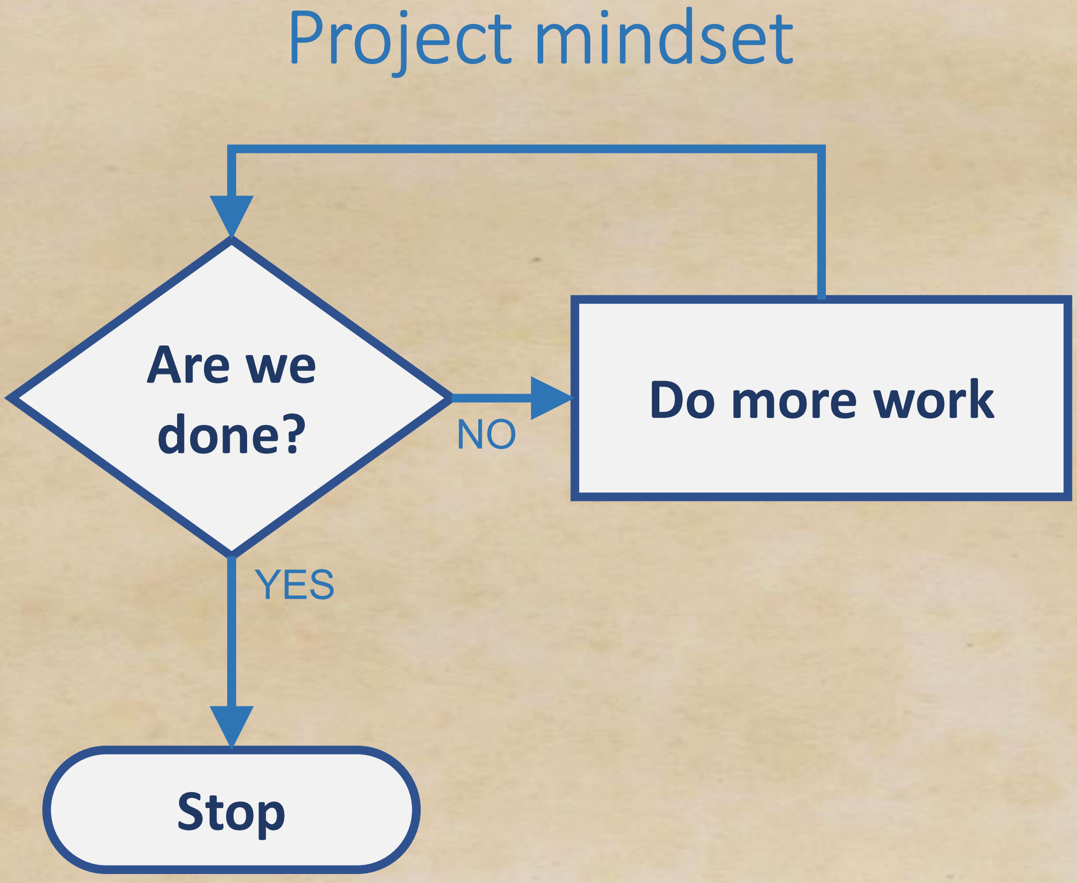


TO DO	DOING	DONE
Teaching	Agile Product Development	Why & What of Agile and Scrum
Scaling Scrum / Agile Organization		Scrum Leadership
Scrum Leadership		Facilitation
		Coaching
		Organizational Change
		Self-managing Teams
		XP & Kanban

Project vs Product

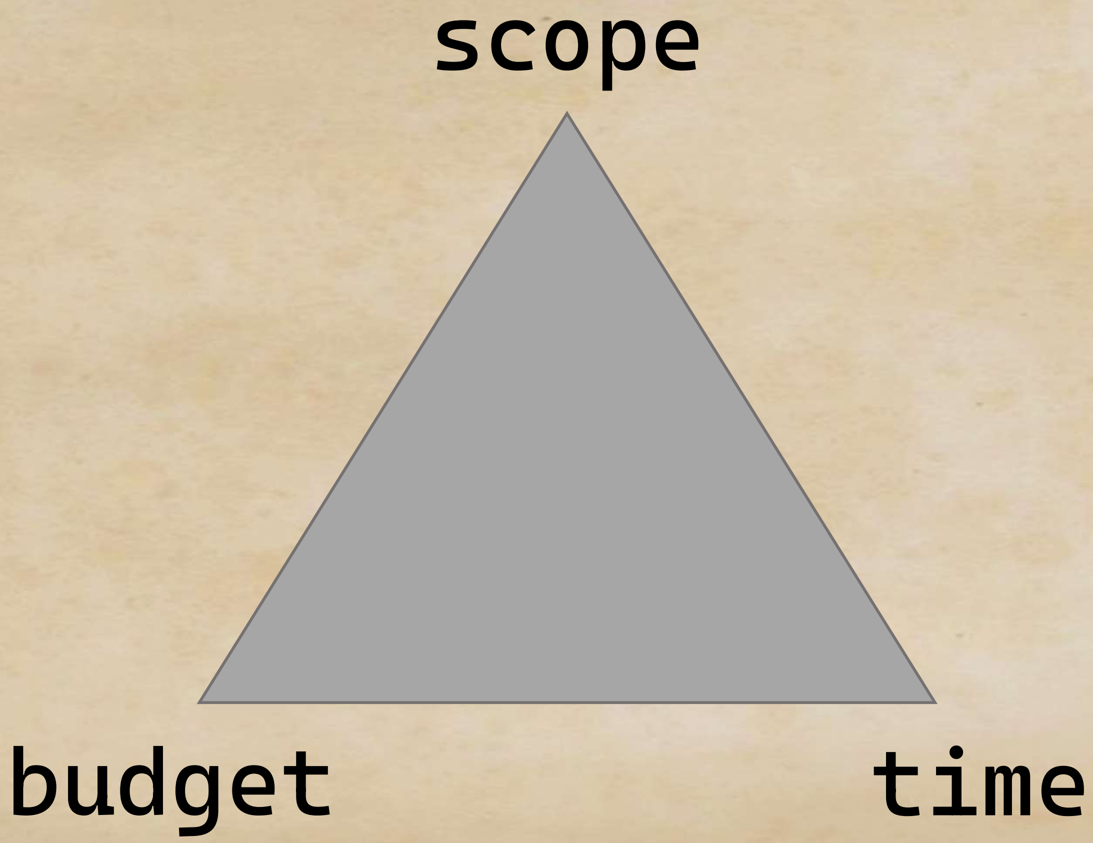
What's the difference?

Mindset: Project



Mindset: Project

Project mindset



scope

budget

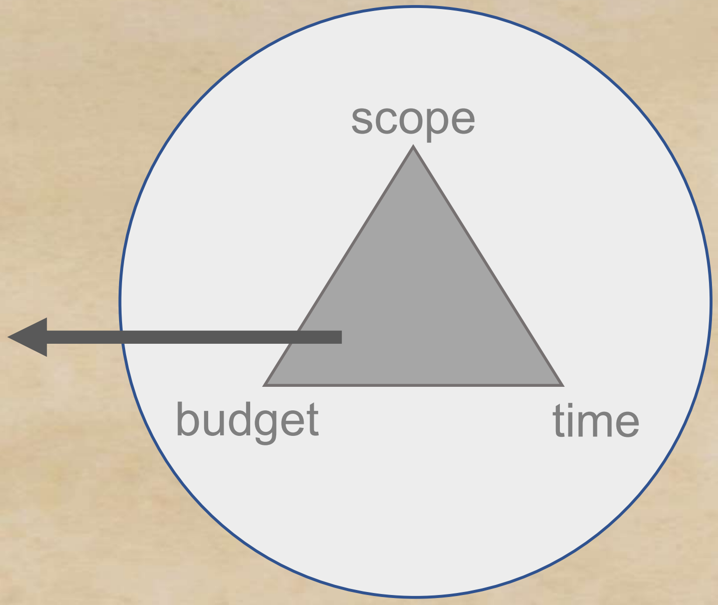
time

Mindset: Project

Project mindset

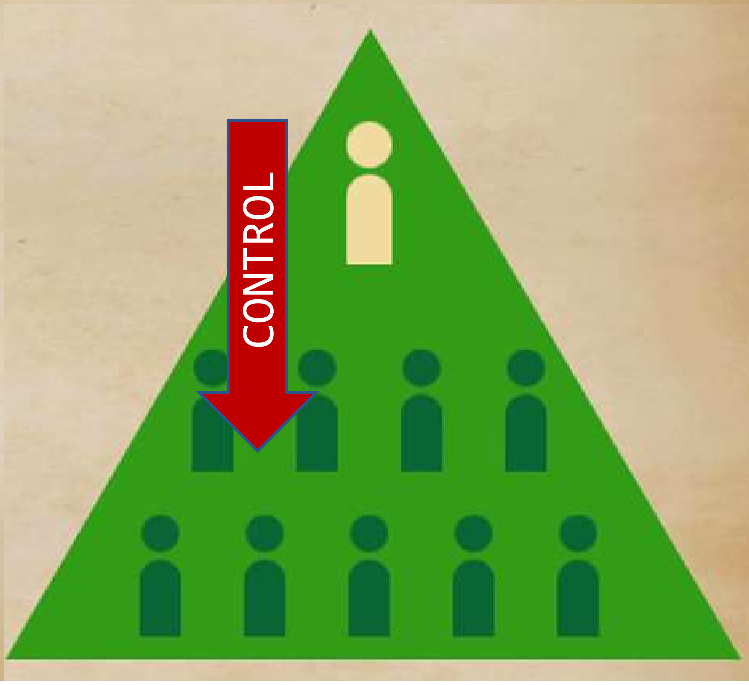
Success ***upfront*** defined inside out:

- Scope
- Budget
- Time

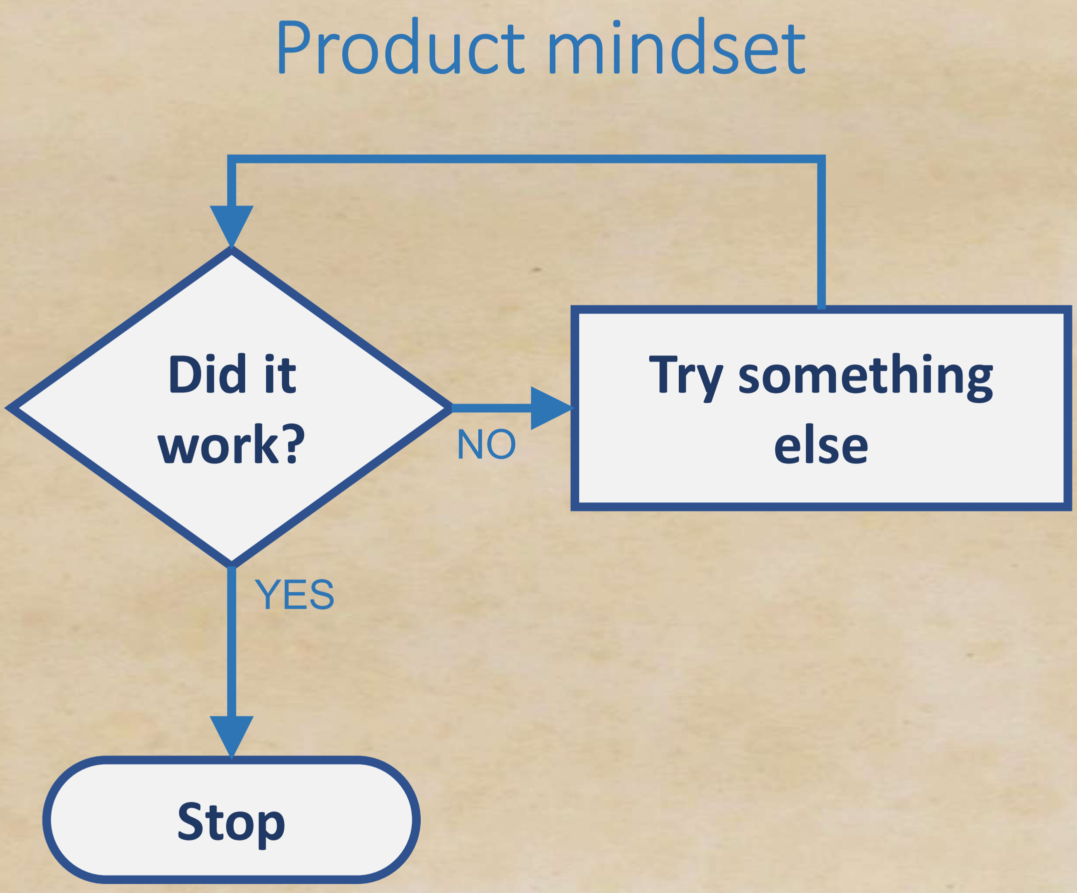


Leads to:

- Less business involvement
- More task management



Mindset: Product

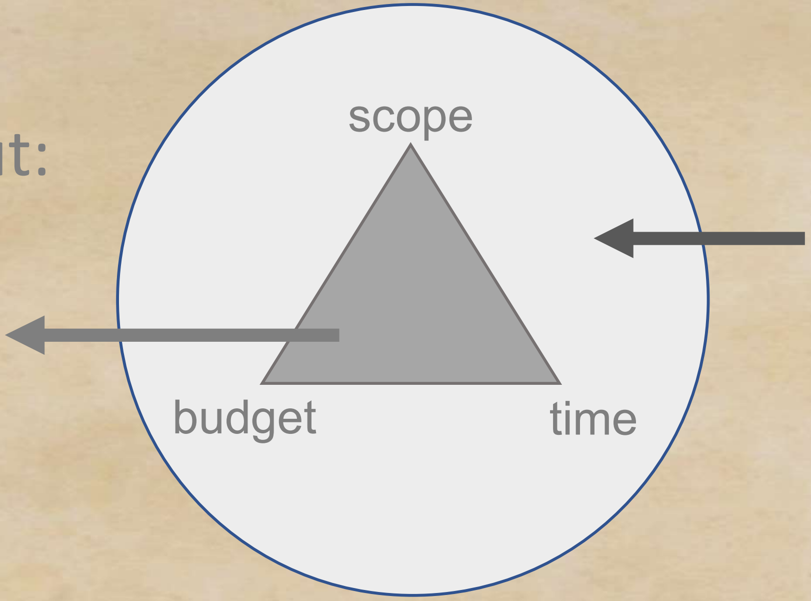


Mindsets compared

Project mindset

Success ***upfront*** defined inside out:

- Scope
- Budget
- Time



Product mindset

Continuously driven by business metrics outside in:

- User adoption
- Revenue
- Cost savings

Leads to:

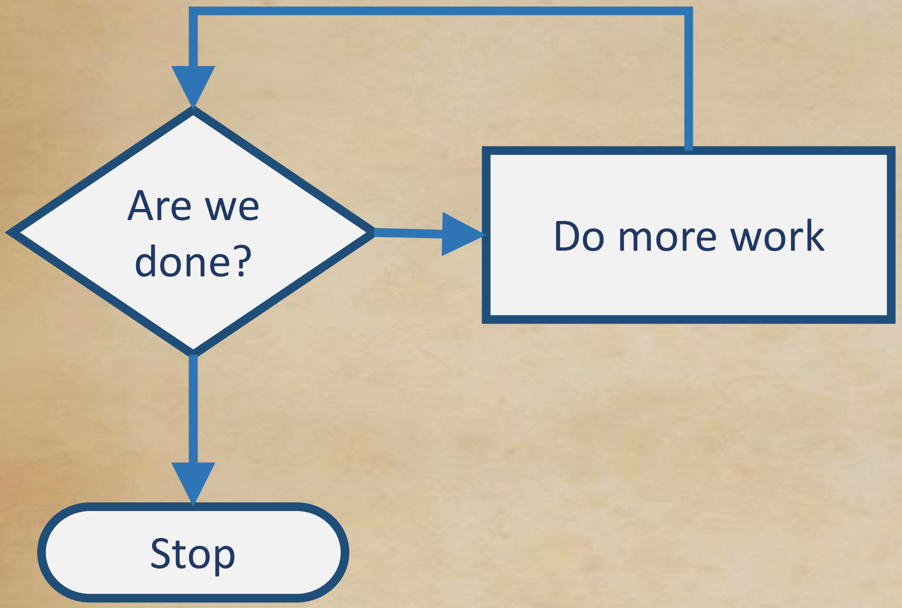
- Less business involvement
- More task management

Leads to:

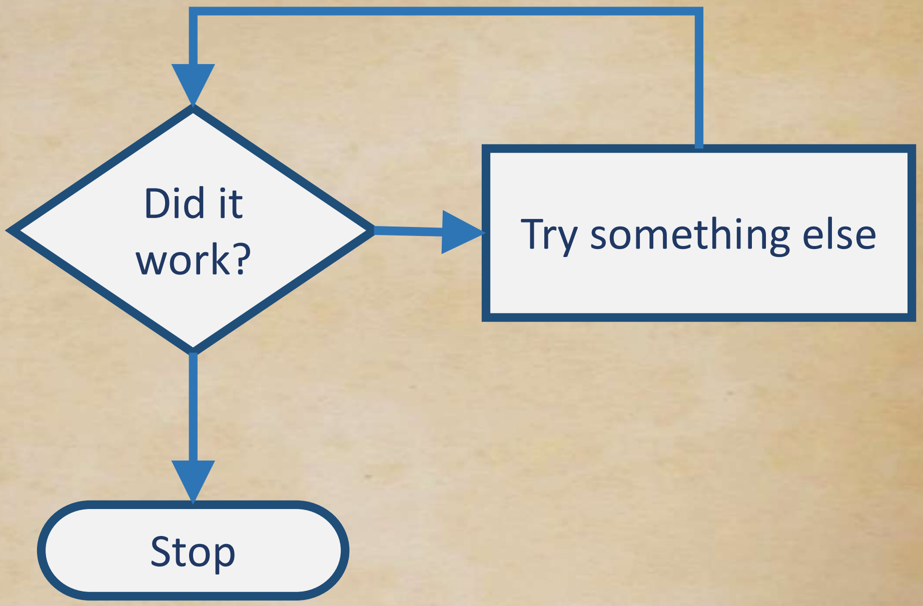
- Less waste
- More creativity
- More releases

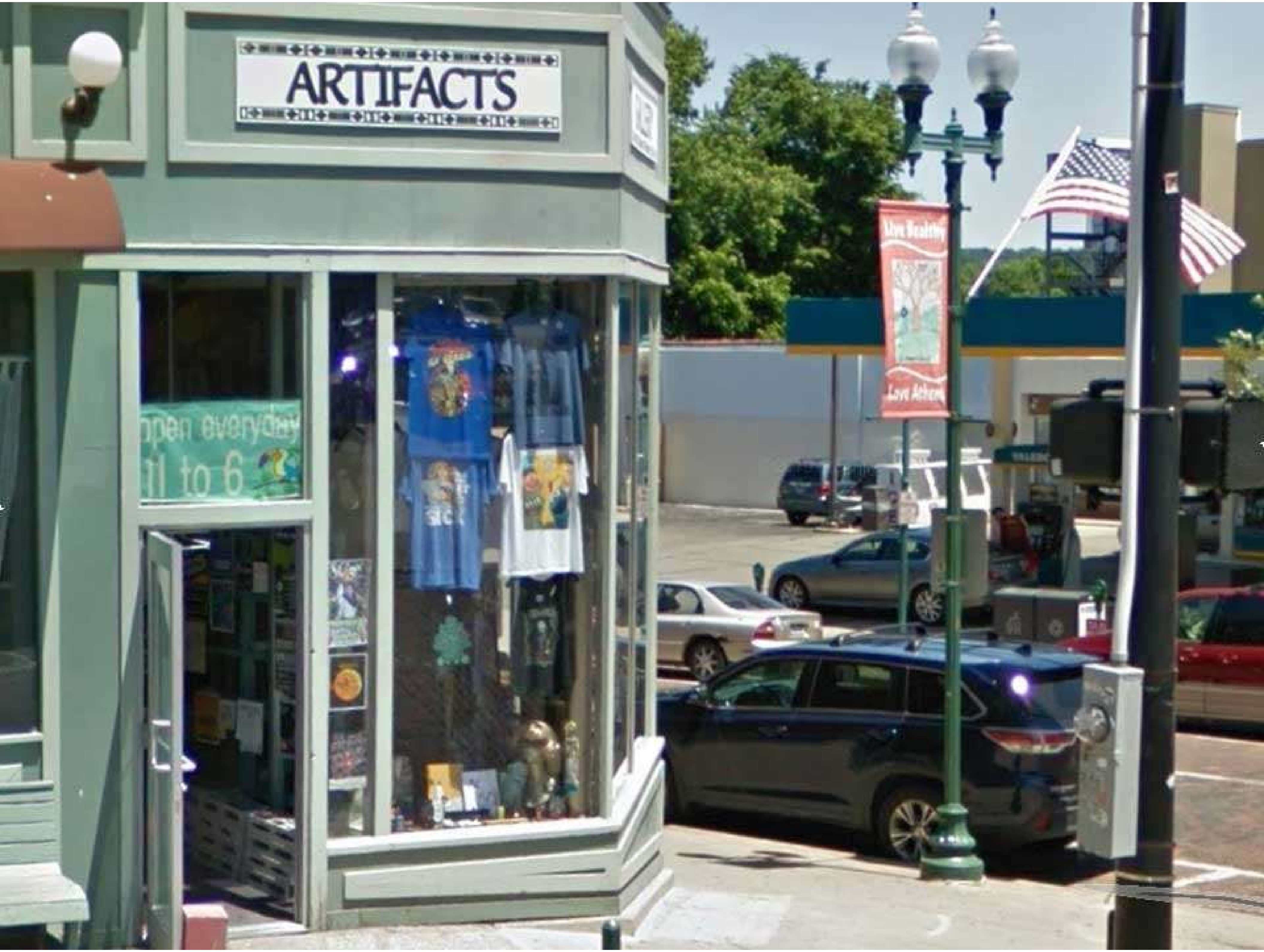
Mindsets compared

Project

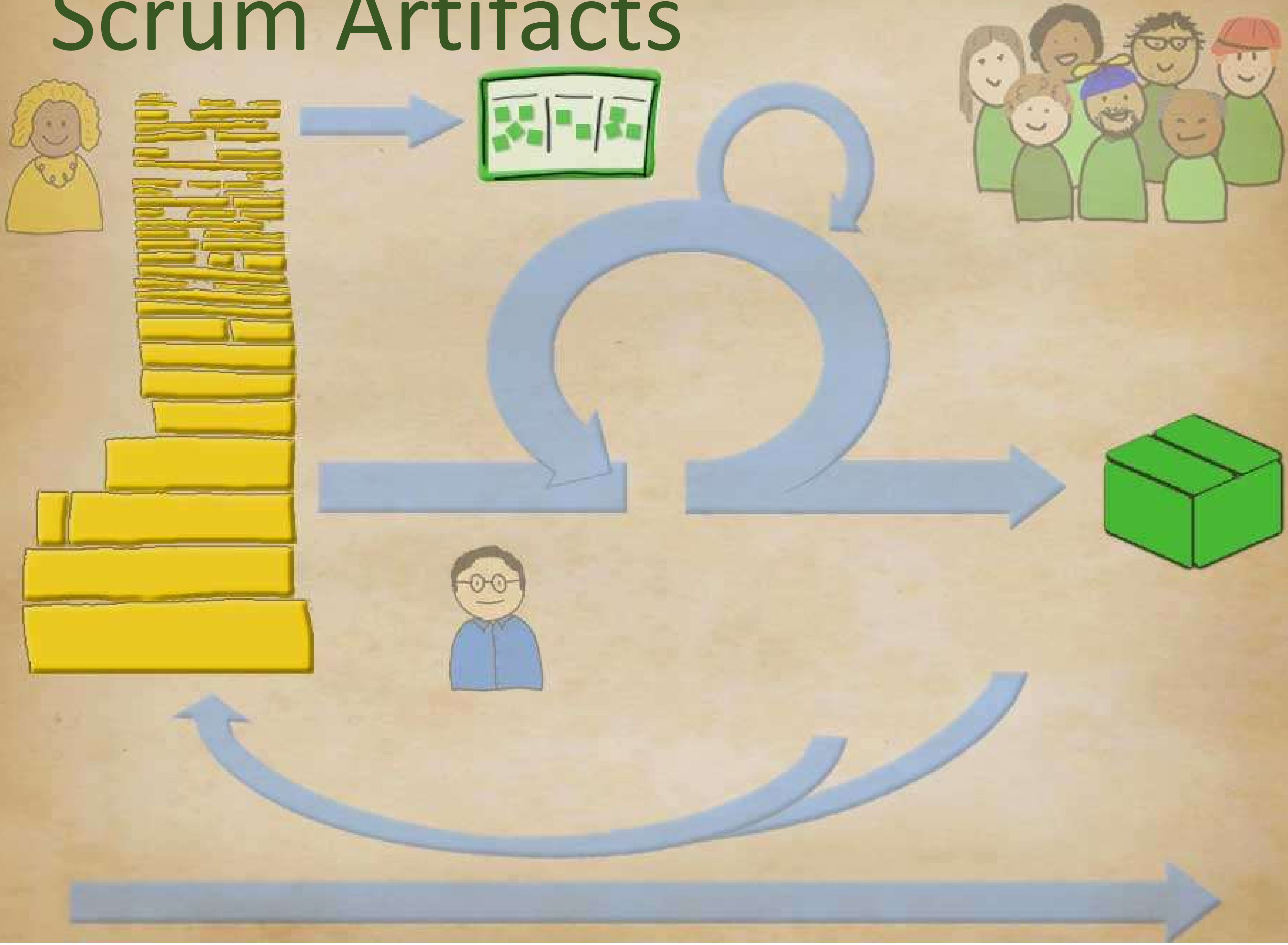


Product




ARTIFACTS
pen everyday
11 to 6
Live Healthy
Love Athens

Scrum Artifacts



Scrum Artifacts

Scrum's artifacts represent ***work*** or ***value***.

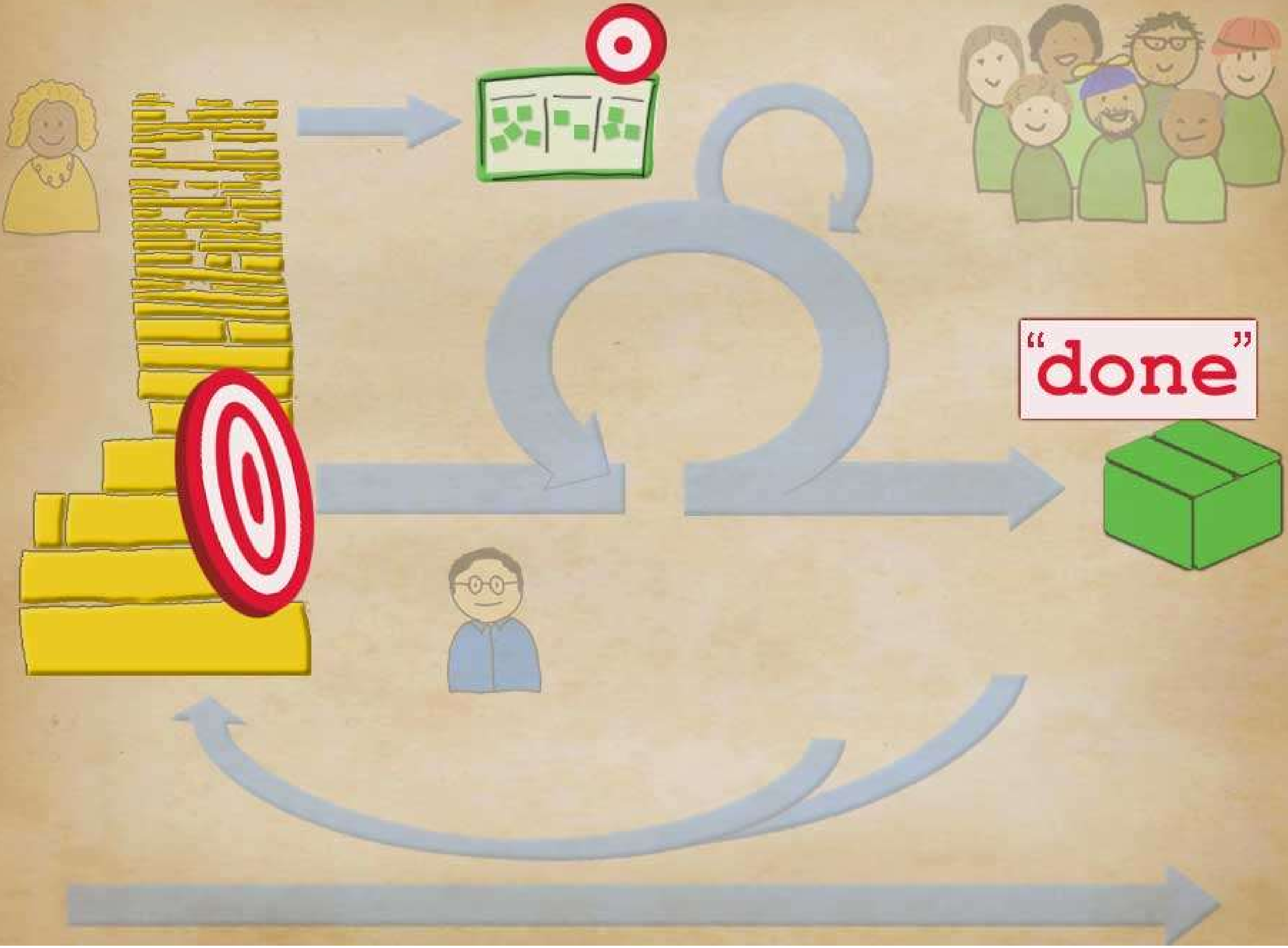
They are designed to maximize transparency of key information.

Thus, everyone inspecting them has the same basis for adaptation.

Scrum Artifacts

Each artifact contains a ***commitment*** to ensure it provides information that enhances transparency and focus against which progress can be measured:

- for the *Product Backlog* it is the ________________
- for the *Sprint Backlog* it is the ________________
- for the *Increment* it is the __________________



"done"

Scrum Artifacts

Each artifact contains a ***commitment*** to ensure it provides information that enhances transparency and focus against which progress can be measured:

- for the *Product Backlog* it is the **Product Goal**
- for the *Sprint Backlog* it is the **Sprint Goal**
- for the *Increment* it is the **Definition of Done**

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

The Scrum Guide - November 2020

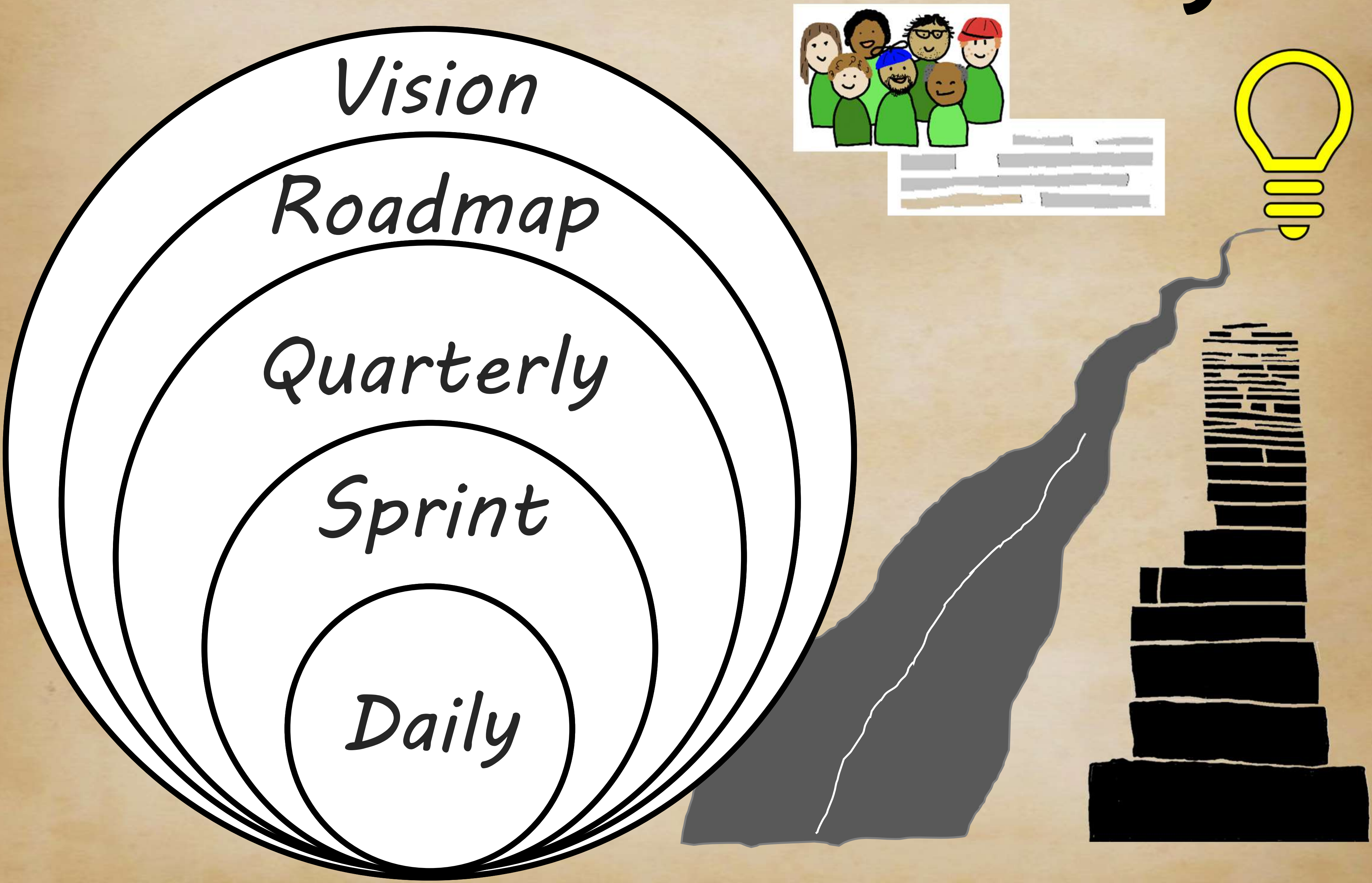
The Breakout Bunch

What is the difference between a Vision and a Goal?

Why might they have removed “vision” from the Scrum Guide?

Why did they add “Product Goal”?

The Five Levels of Planning



Adapted from Hubert Smits Five Levels of Planning

The Five Levels of Planning

Vision

BIG IDEA

>1 year

Elevator Pitch

Elevator Pitch

Vision

For ____________

who _____________,

the _________

is ________________.

Unlike _________,

our product _________.

Geoffrey Moore - Crossing the Chasm

The Pixar Pitch

Once upon a time ____________________.

Every day, ____________________________.

One day ____________________________.

Because of that, ________________________.

Because of that, _____________________.

Until finally, ___________________________.

Finding Nemo

Once upon a time … there was a widowed fish, named Marlin, who was extremely protective of his only son, Nemo.

Every day … Marlin warned Nemo of the ocean’s dangers and implored him not to swim far away.

One day … in an act of defiance, Nemo ignores his father’s warnings and swims into the open water.

Because of that … he is captured by a diver and ends up in the fish tank of a dentist in Sydney.

Because of that … Marlin sets off on a journey to recover Nemo, enlisting the help of other sea creatures along the way.

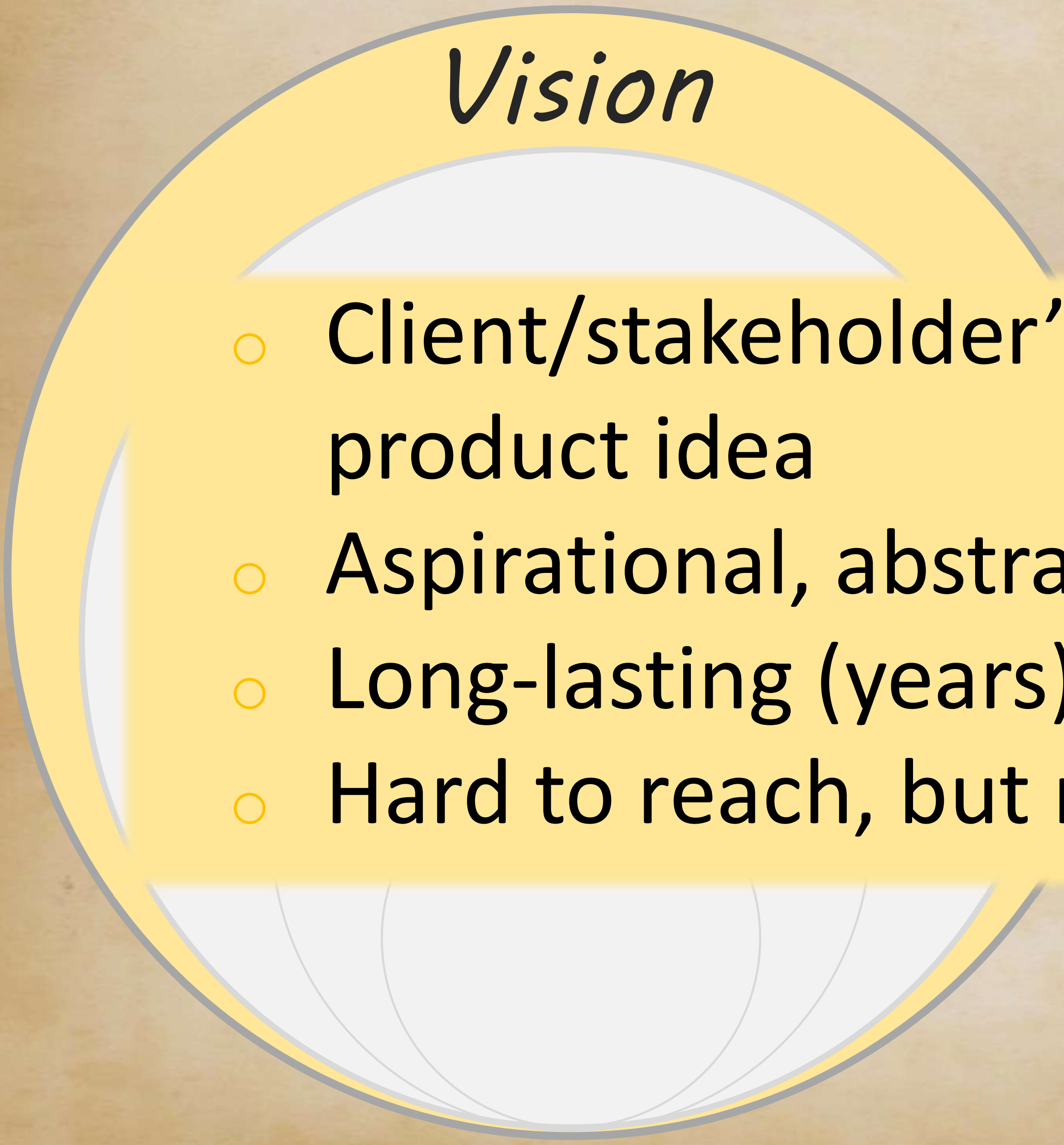
Until finally … Marlin and Nemo find each other, reunite and learn that love depends on trust.

Goals *vs* Visions

- Visions are limitless, goals are not
- Goals require action, visions don't
- Goals are short term, visions are not
- Goals are realistic
- Goals produce results

5 Differences between setting a goal and a vision - Bhattacharyya (2016)

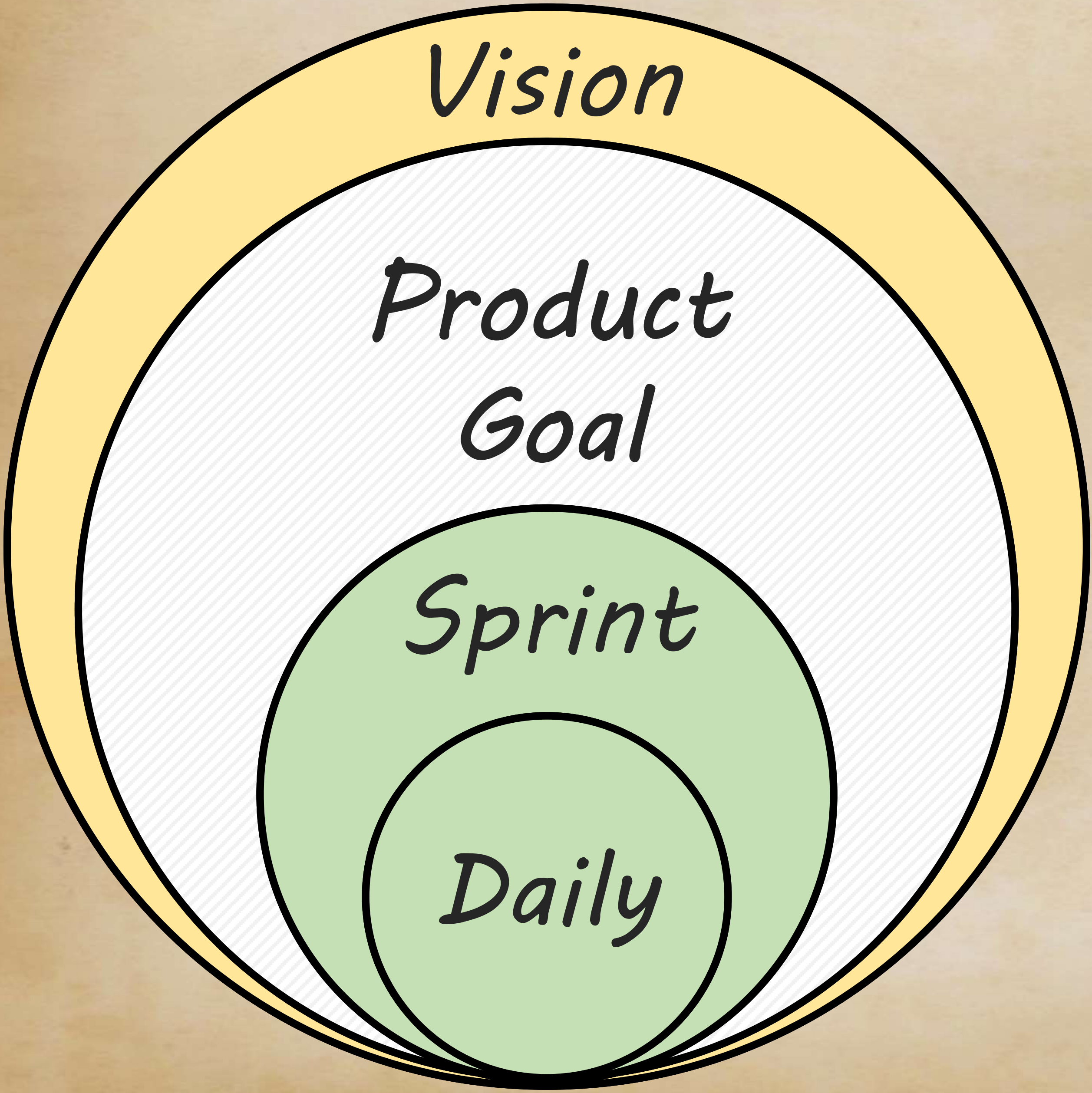
Product Vision



○ Client/stakeholder’s version of the product idea
○ Aspirational, abstract, high-level
○ Long-lasting (years)
○ Hard to reach, but not impossible

Adapted from Hubert Smits Five Levels of Planning

Product Goal *bridges the gap*



Product Goal

- Derived from Vision
- Gives Focus to Vision
- Measurable
- Attainable (perhaps within 3-6 months)
- Product Goals emerge and change over time
- Each Product Goal may have multiple features

Adapted from Hubert Smits Five Levels of Planning

Decompose the Vision into Product Goals

VISION

Future Product Goals

Next Product Goal

Product Goal #1

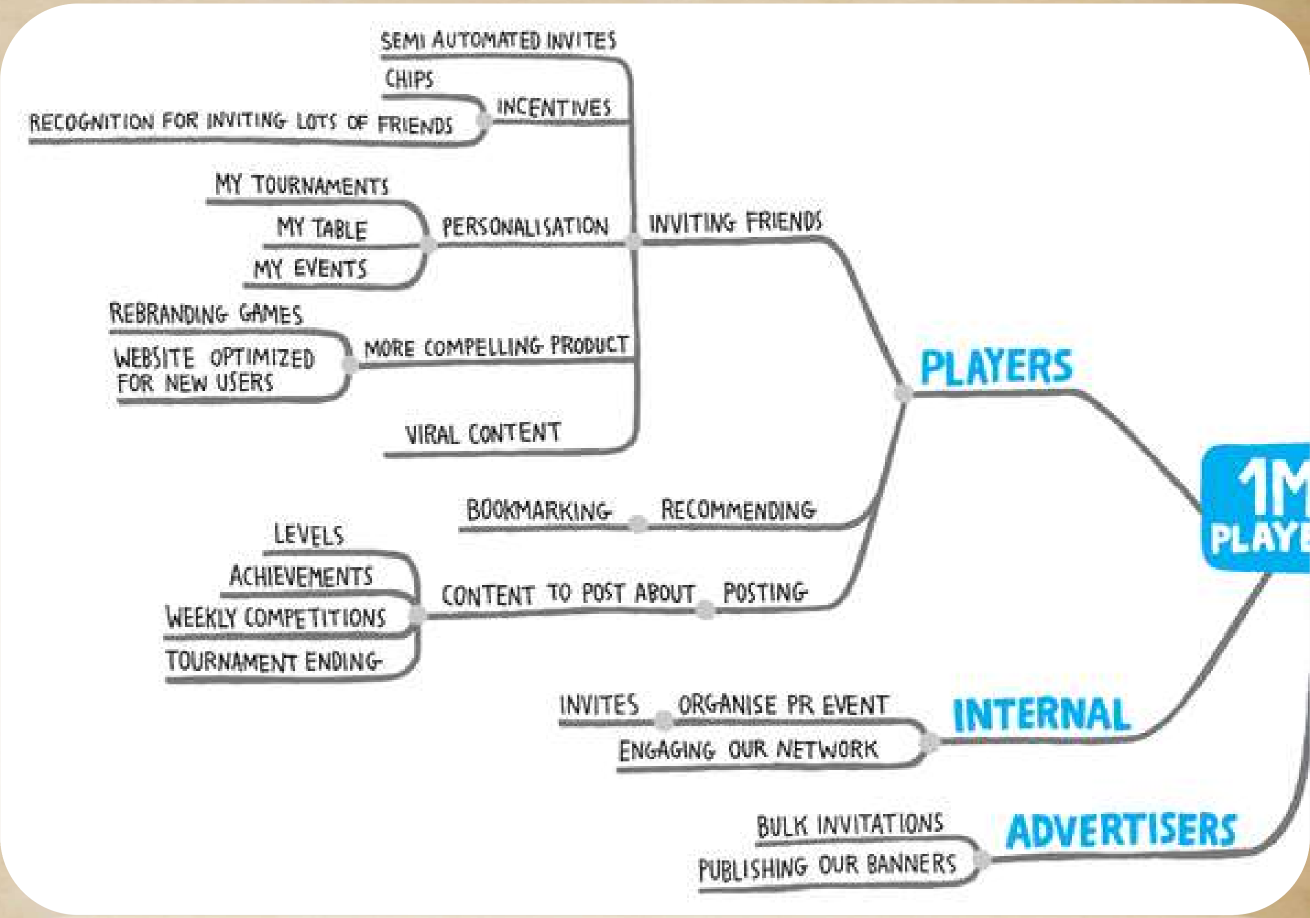
Impact Mapping

- Impact mapping is a lightweight, collaborative planning technique for Product Owners and teams to begin visualizing what’s needed to turn a Product Vision into a reality.
- It is based on user interaction design, outcome driven planning and mind mapping.
- Impact maps help delivery teams and stakeholders visualize assumptions, explain how deliverables connect to user needs, and communicate how user outcomes relate to higher level organizational goals.

Source: impactmapping.org

What do impact maps look like?

Structuring information as a mind map allows you to describe many different dimensions of a product milestone plan in a single visualization.



Source: impactmapping.org

Impact Mapping

Goal - The objective that is trying to be achieved or answering:

- Why are we pursuing this idea?
- Why is it needed or wanted?

Actor - The people who can influence the outcome, including:

- Who can produce the desired results or effect?
- Who can obstruct it?
- Who are the consumers or users of the product?

Impact Mapping

Impacts - The next branch puts the actors in perspective of the business goal:

- How should actors' behavior change?
- How can they help achieve the goal?
- How can they obstruct the goal?

Deliverables - The third branch addresses scope:

- What can we do as a company or a delivery team to support the required impacts?
- What are the features desired or needed?
- What do we need to deliver the features?

Scaling Impact Mapping

- In a scaled situation, Impact Mapping can also be used to identify and reduce dependencies between teams
- It is important in those situations to color-code so that each team's work can be more easily identified
- Joint sessions will need to be held accommodating all respective team members or sending a few trusted ambassadors from each involved team to an initial, joint session

Decompose the Product Goal into Sprint Goals

Product Goal
Sprints
Sprint
Sprint

The Product Backlog

- Single source of work undertaken by the Scrum Team
- Ordered list of what is needed to improve the product
  - *features, functions, requirements, enhancements, and fixes*
- Anything on the Product Backlog is a P.B.I. or Product Backlog Item
- Persists as long as there is a Product Goal
- Dynamic
  - *Constantly changes to identify product needs*
  - *Frequently reprioritized*
  - *Is emergent or progressively elaborating*

Assignments #5 & 6:

When you get back to a work environment:

- create or refine a Product Goal with the Scrum Team and stakeholders;
- create a Product Backlog that supports achievement of a Product Goal;

Document this (and your reactions, learnings) in your personal journal.

What’s the difference between Meetings and Activities ?

Why is Product Backlog Refinement an activity, not a meeting?

What does “refinement” look like, who participates, and is it one-and-done?

Product Backlog Refinement

… the act of breaking down
and further defining
Product Backlog items
into smaller
more precise
items

This is an ongoing activity to add details,
such as a description, order, and size

The Scrum Guide - November 2020

Product Backlog Refinement

Why split?

Too big:

- Cannot give accurate estimate
- Cannot fit into iteration

Dependencies exist:

- Compound PBIs need to be split – look for:
  - Conjunctions (and, but, or)
  - Limiting Phrases (unless, until, without, except)

Cannot satisfy INVEST:

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable

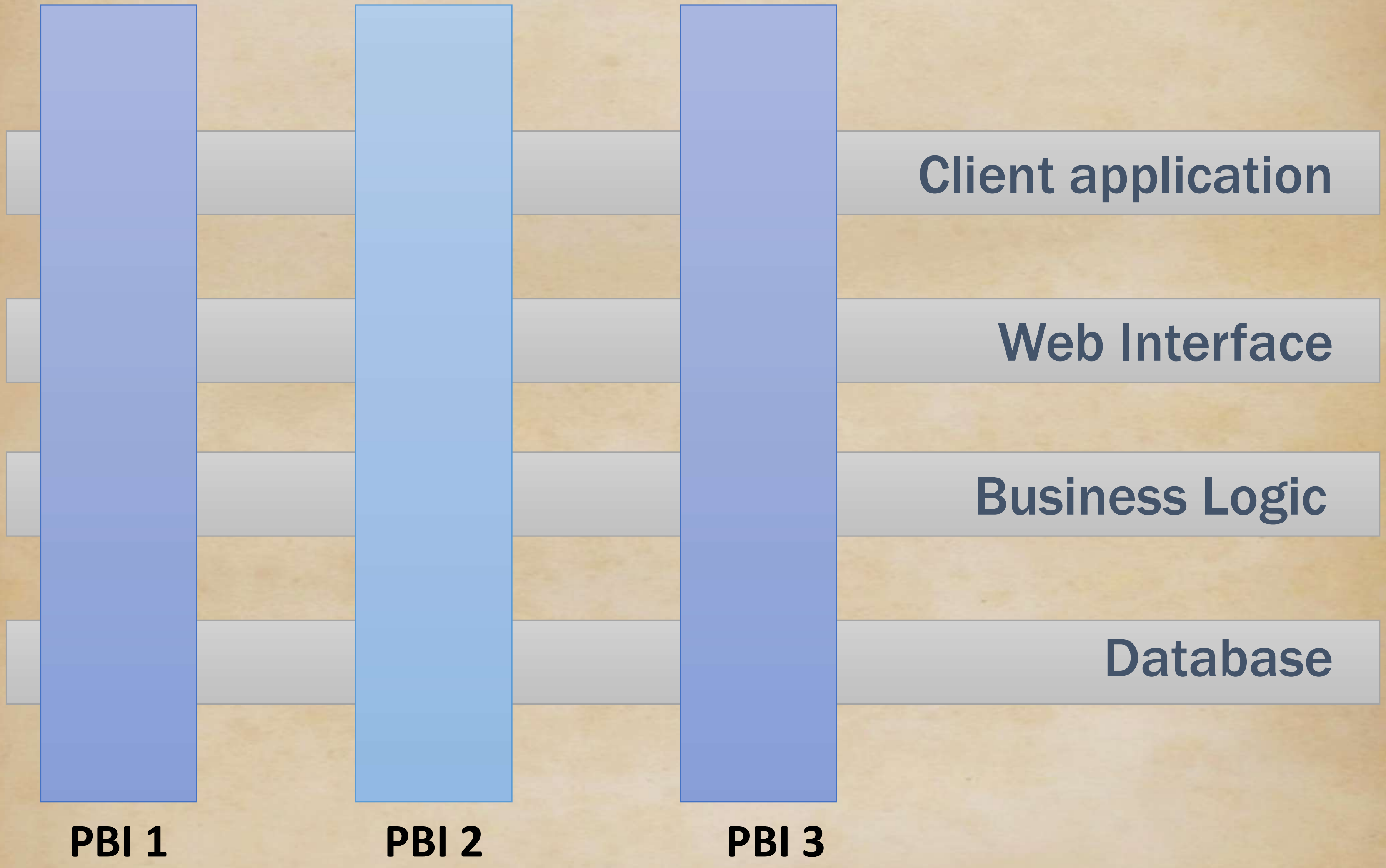
INVEST

Independent	The story should be self-contained, in a way that there is no inherent dependency on another story.
Negotiable	User stories, up until they are 'in-implementation', can be negotiated (between PO and team), changed or rewritten.
Valuable	A user story must deliver value to the end user.
Estimable	You must be able to estimate the size of a user story.
Small	User stories should not be so big as to become impossible to plan/task/prioritize with a high level of certainty.
Testable	The user story or its related description must provide the necessary information to make test development possible.

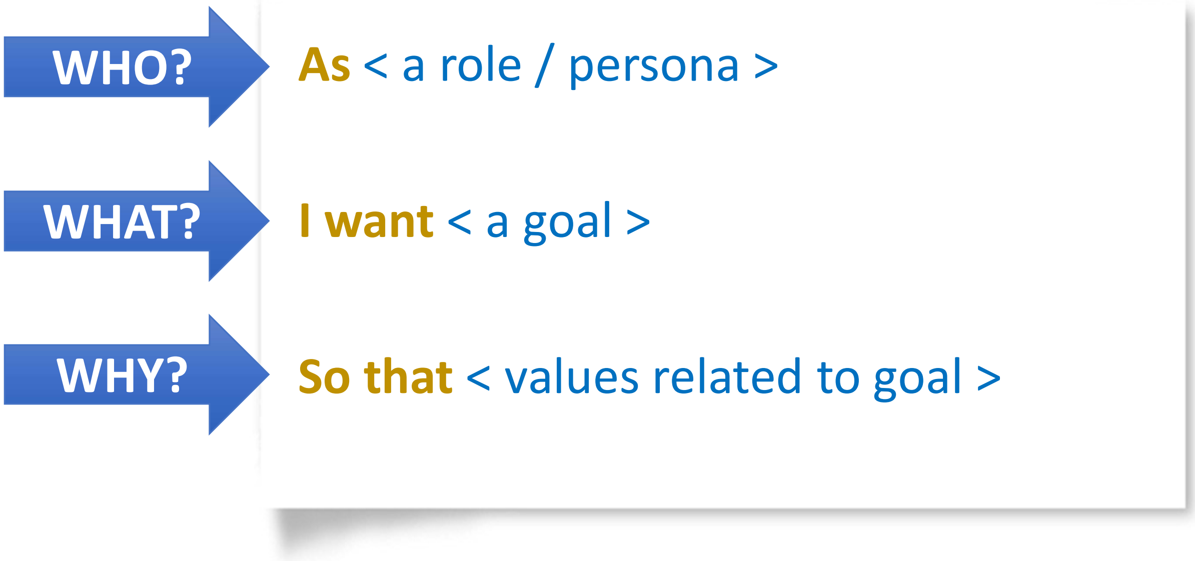
Product Backlog Refinement

- PBIs should represent some level of end-to-end functionality
- Do not split PBIs into components such as design, code front end, code middle tier, code back end
- Deliver cohesive subset of all layers
- Do simplest thing that could possibly work

Product Backlog Refinement



USER STORY – TEMPLATE



WHO?

As < a role / persona >

WHAT?

I want < a goal >

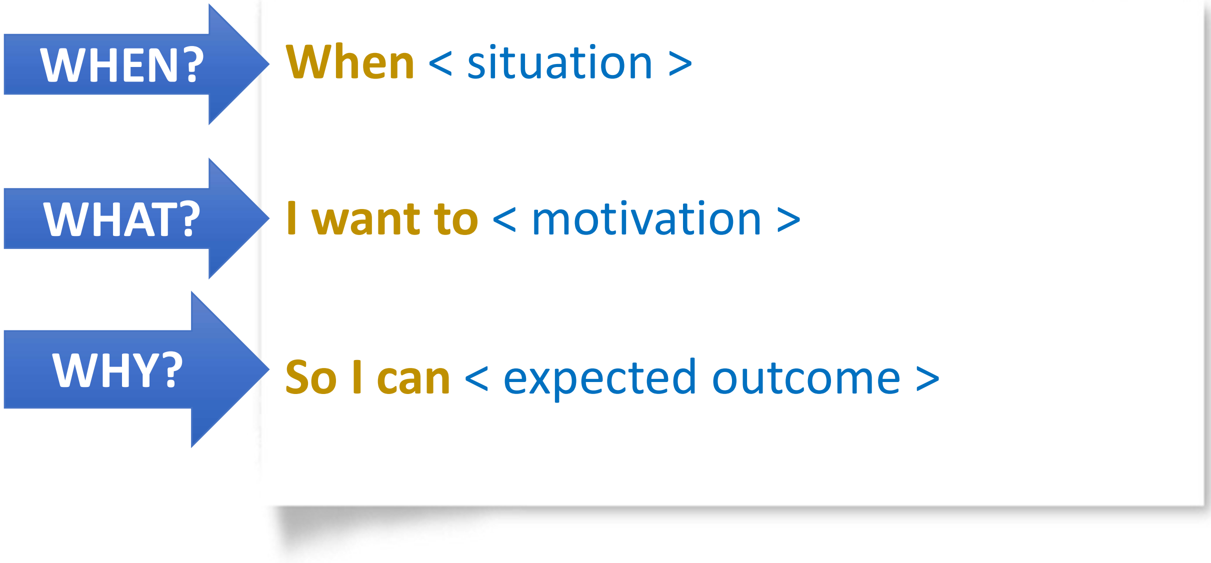
WHY?

So that < values related to goal >

Tip: Write user stories in a “first person” perspective

Any story template is a suggestion, not a rule.

JOB STORY – TEMPLATE

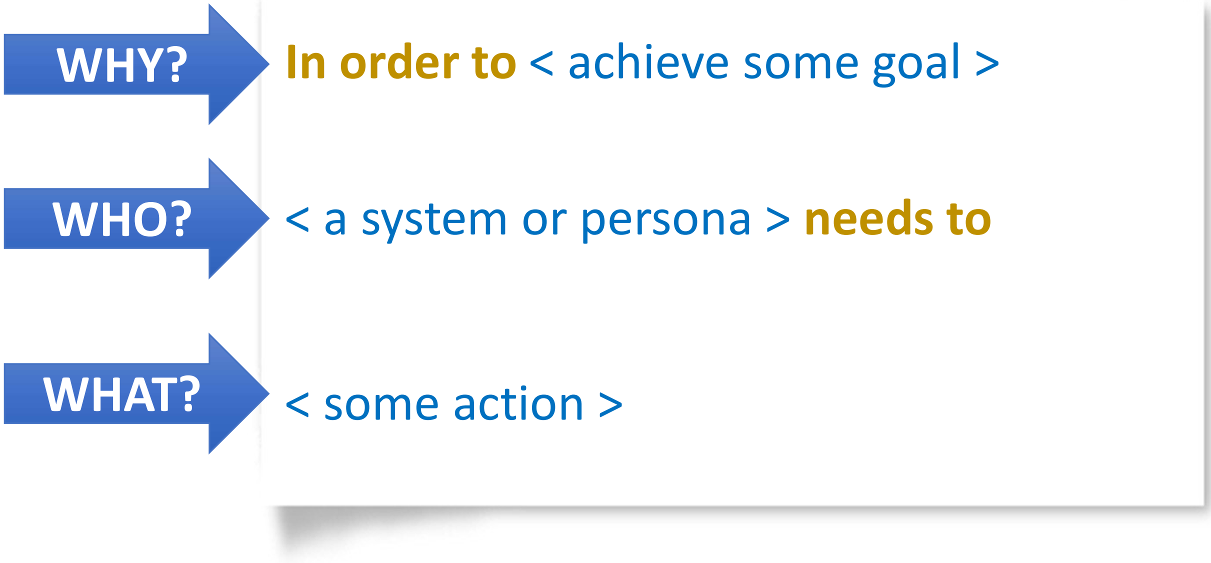


WHEN? When < situation >

WHAT? I want to < motivation >

WHY? So I can < expected outcome >

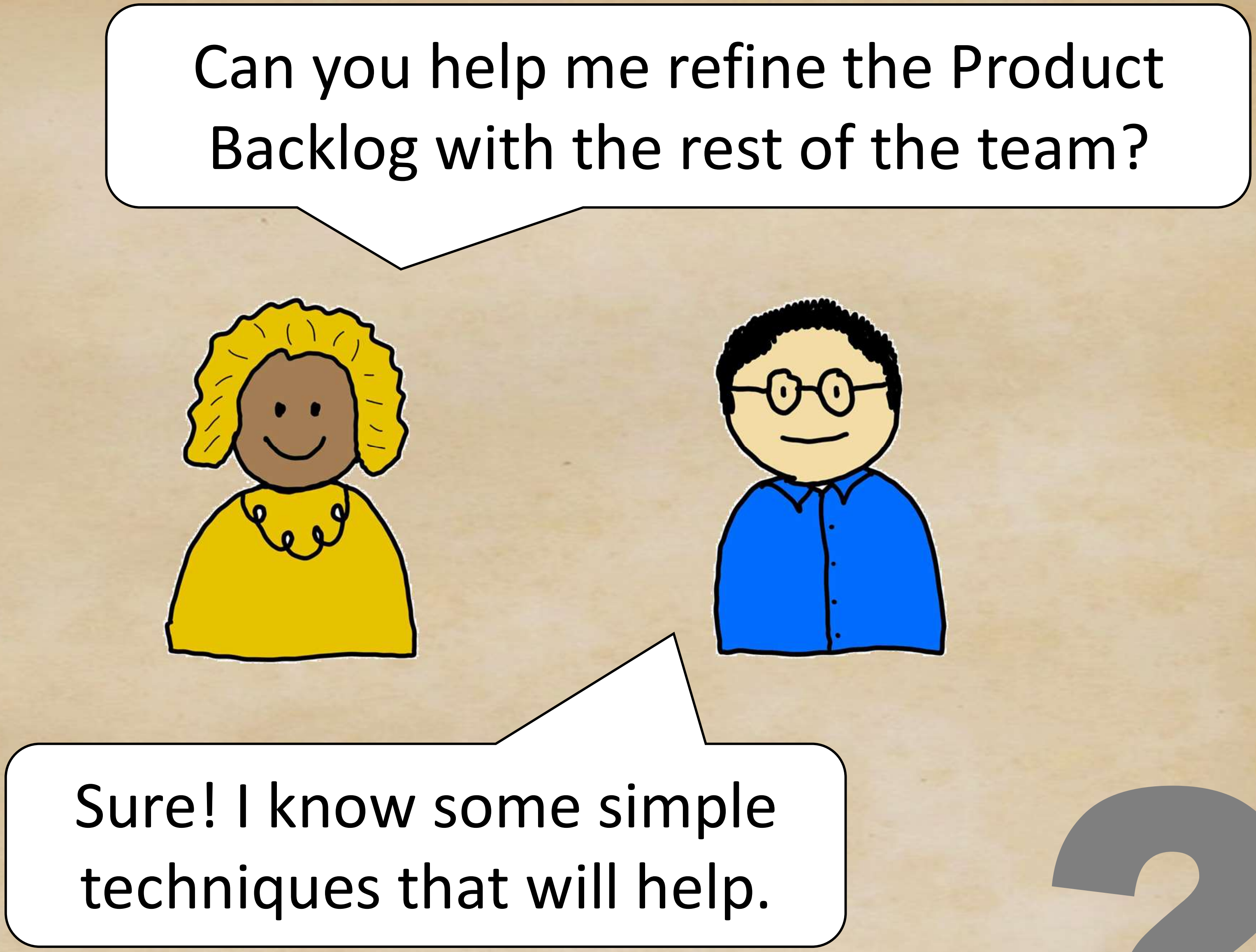
“SYSTEM” STORY – TEMPLATE



WHY? In order to < achieve some goal >

WHO? < a system or persona > needs to

WHAT? < some action >


Can you help me refine the Product Backlog with the rest of the team?
Sure! I know some simple techniques that will help.

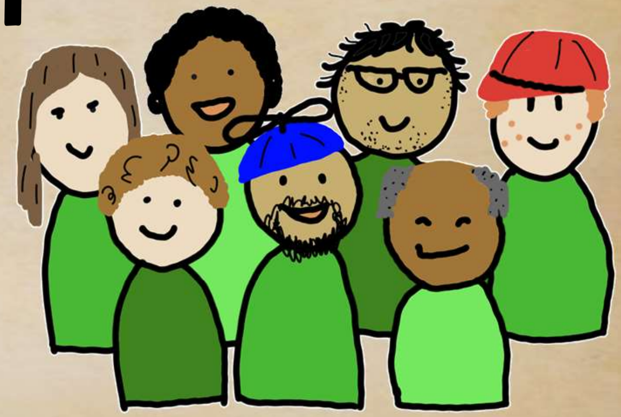
STORY MAPPING

STORY WRITING WORKSHOP

PERSONA CREATION

CUSTOMER JOURNEY MAP

IMPACT MAPPING



Definition of Done

- Th the opportunity to realize value at the end of each Sprint
- The minimum Scrum DoD is “Potentially Shippable”
- If the Team extends this Definition they should explicitly capture their additions
- The Team should strive to have “production worthy” code/product, even if the Product Owner doesn’t intend to release it

Definition of Done – SW example

- Unit tests created, checked in, all passing
- Code builds successfully on integration environment
- Automated unit test coverage > 80%
- All acceptance criteria pass
- Peer code review complete
- No open defects
- Needed documentation is complete
- ***Product owner has reviewed and accepted story***

What if your product isn't software?

Definition of Done - examples

Talent Recruiting:

- Interviews completed; background check; offer extended or pass on applicant

Sales Organization:

- Proposals moved to signature or closed

Home Renovation:

- Completed to acceptance; inspections pass; regulations met; mess hauled away

Do the Dishes:

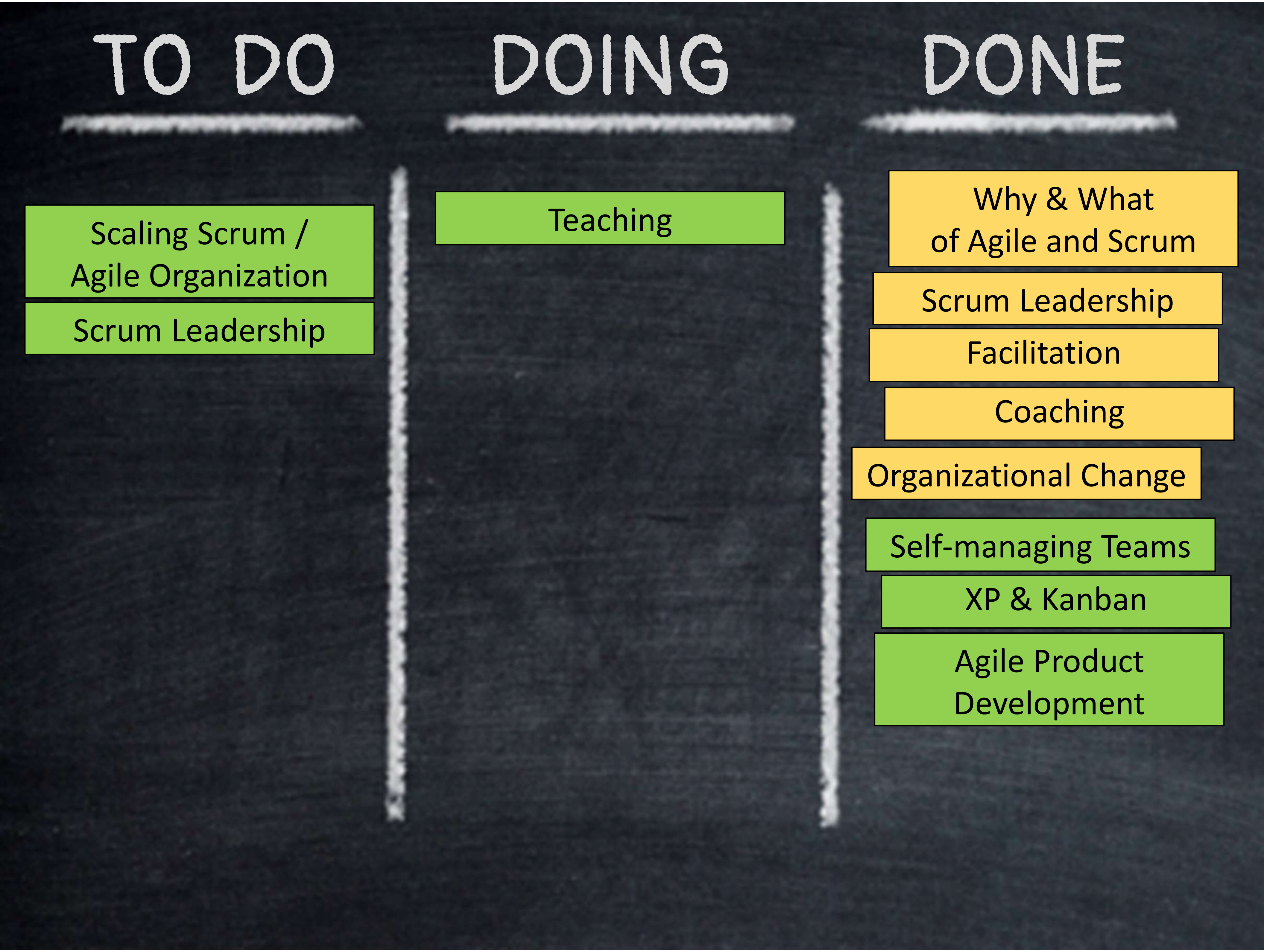
- Washed, Dried and put back into cupboard

Discuss how to create a strong Definition of Done.

Who participates?

How will you facilitate? What techniques could you use?

Are there any people or artifacts outside the team that you can call upon?



TO DO	DOING	DONE
Scaling Scrum / Agile Organization	Teaching	Why & What of Agile and Scrum
Scrum Leadership		Scrum Leadership
		Facilitation
		Coaching
		Organizational Change
		Self-managing Teams
		XP & Kanban
		Agile Product Development

How would you explain Scrum and its benefits to a business stakeholder?

Teach Back

Scrum is...

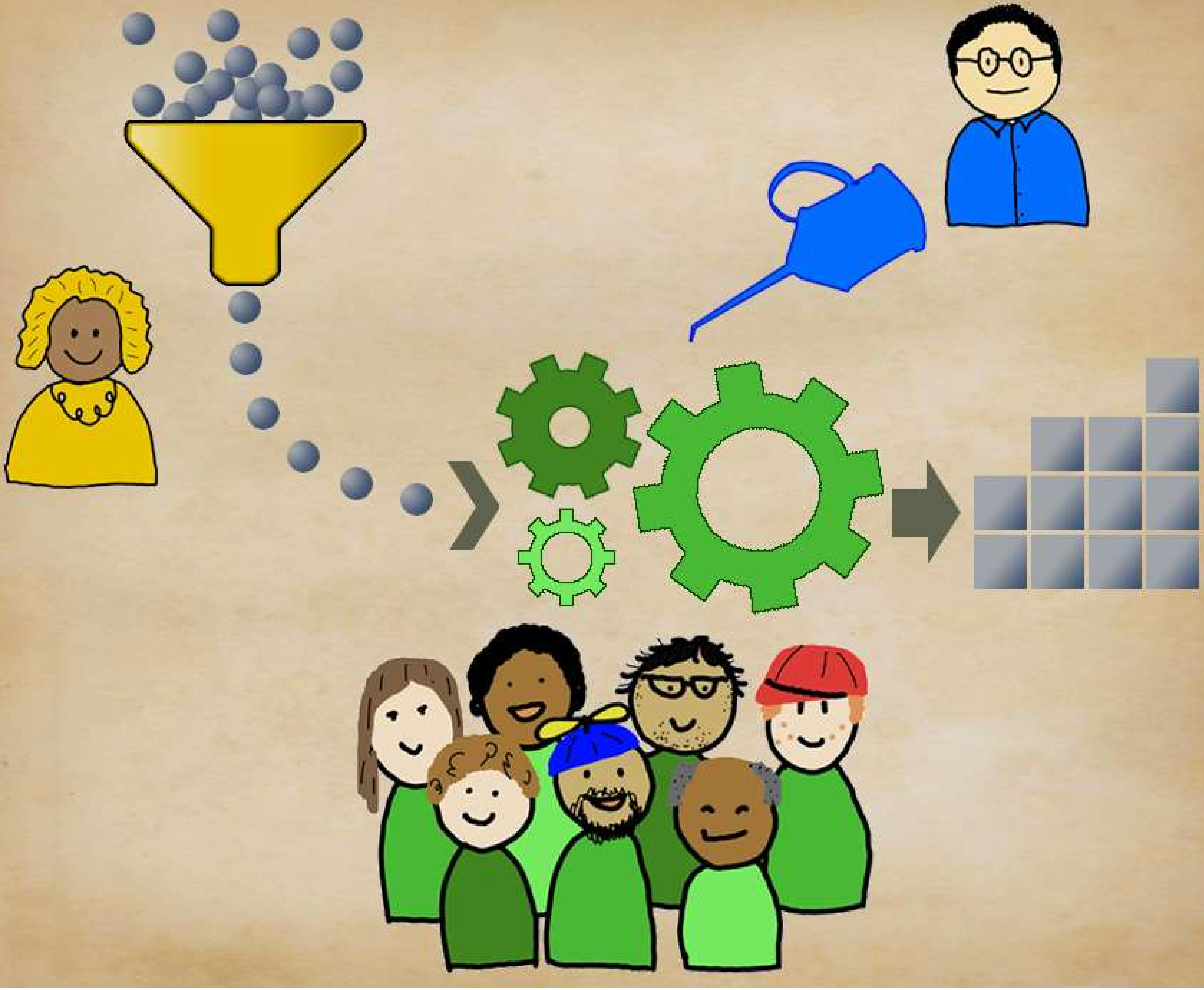
a lightweight, Agile framework for iterative development

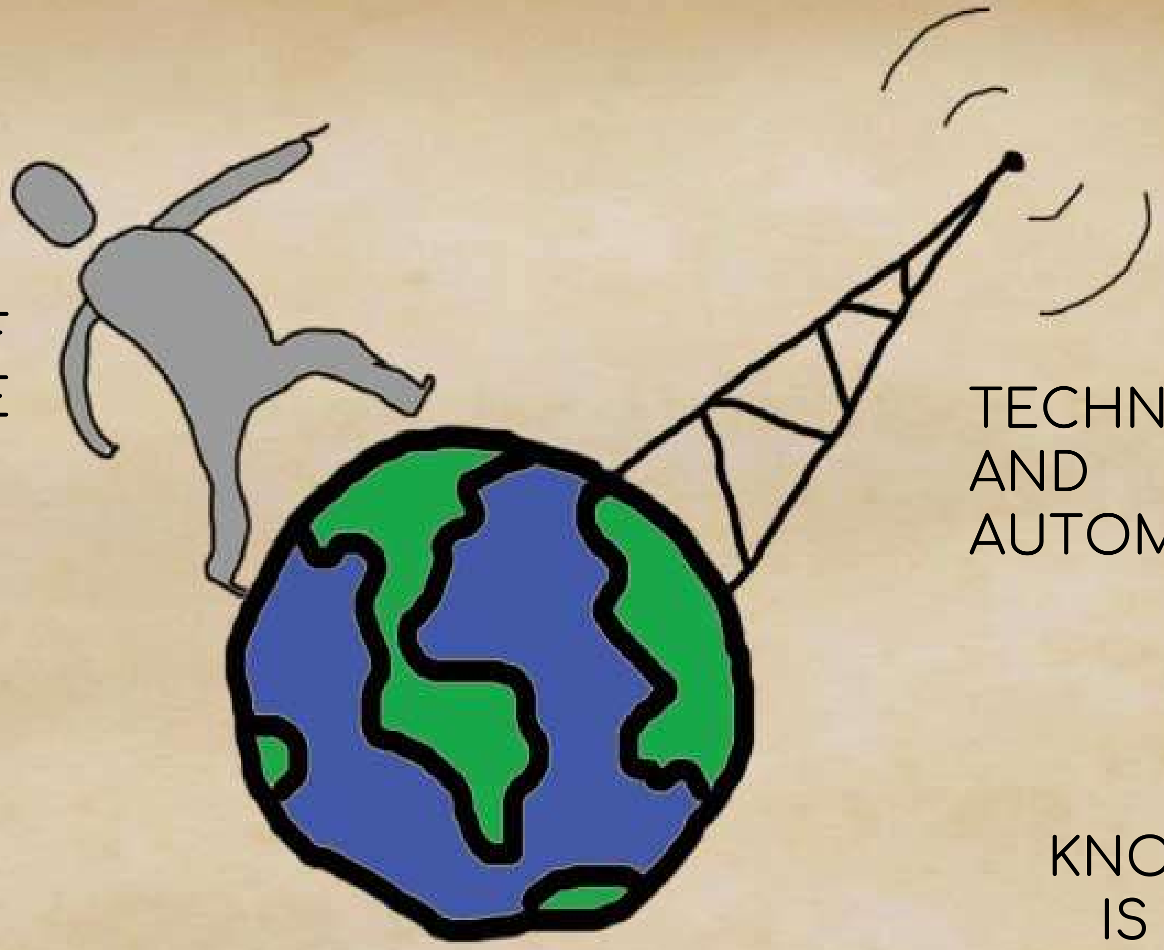
Lightweight - Set is small, not comprehensive

Agile - Reflects the *Agile Manifesto*

Process Framework - A specific set of practices

Iterative - Deliver results in frequent increments





PACE OF CHANGE

TECHNOLOGY AND AUTOMATION

CUSTOMER EXPECTATIONS

KNOWLEDGE IS NOW A COMMODITY

GLOBALIZATION

PRODUCT COMPLEXITY

INCREASED REGULATIONS

HYPER-COMPETITION

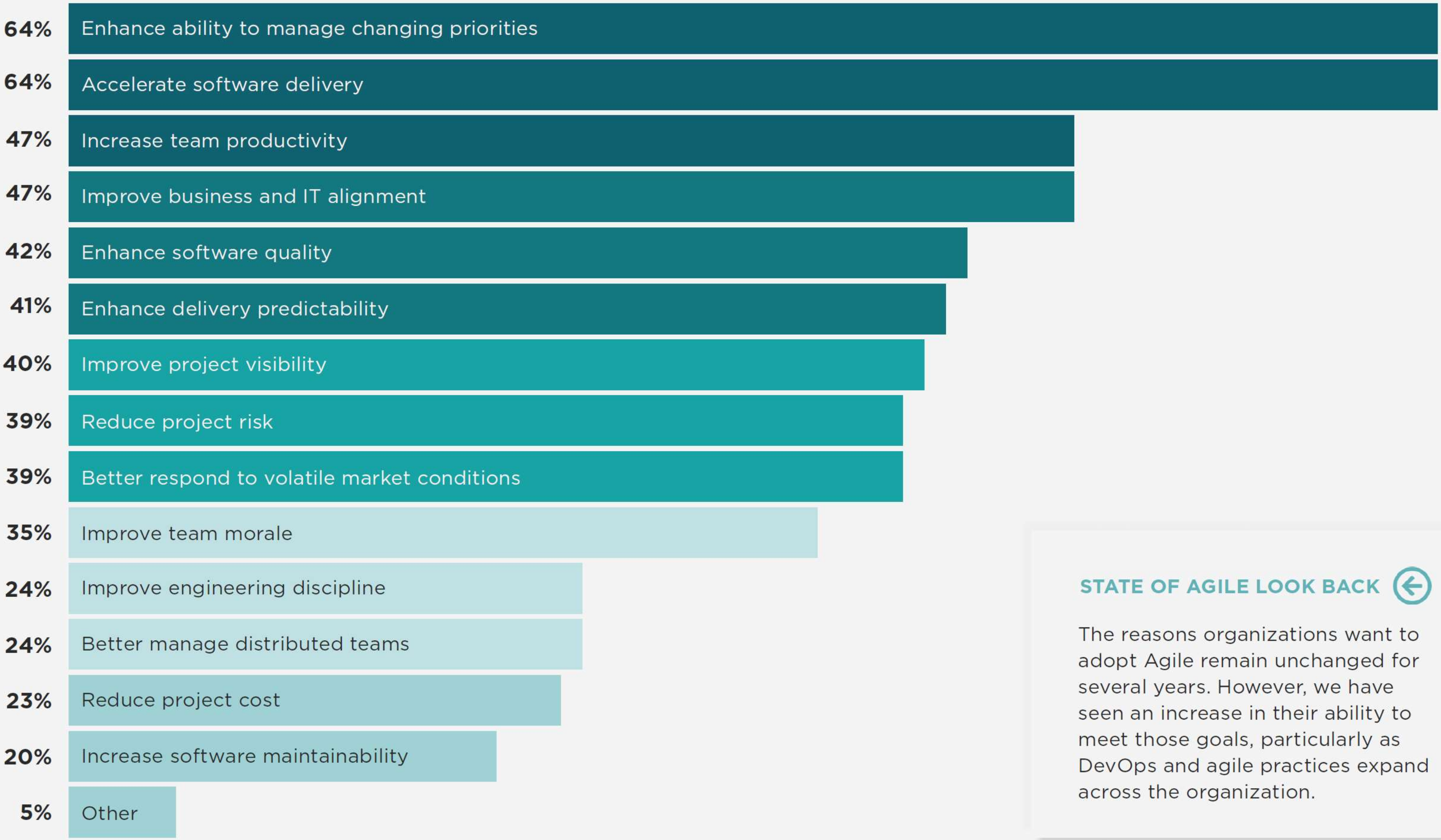
Agile Adoption

15th State of Agile Report (2021)

stateofagile.com

What were the most important reasons for adopting Agile within your team or organization?

**Respondents were able to select multiple responses to this question.*



STATE OF AGILE LOOK BACK

The reasons organizations want to adopt Agile remain unchanged for several years. However, we have seen an increase in their ability to meet those goals, particularly as DevOps and agile practices expand across the organization.

What are a few common organizational impediments outside the scope of a team that can affect the effectiveness of Scrum Teams?

(Remember to Stretch)

Root Cause Analysis

Helps answer **why** a problem or impediment has occurred:

- what happened?
- why did it happened?
- what can be done to reduce the likelihood of the problem happening again?

Root Cause Analysis

Define the Problem:

- *What is happening? What are specific symptoms?*

Collect Data:

- *What proof do you have that the problem exists?*
- *How long has it existed? What is the impact?*

Identify Possible Causal Factors:

- *What sequence of events lead to the problem?*
- *What conditions allow the problem to occur?*

Identify the Root Cause:

- *Why does the causal factor exist?*
- *What is the real reason the problem occurred?*

Recommend and Implement Solutions:

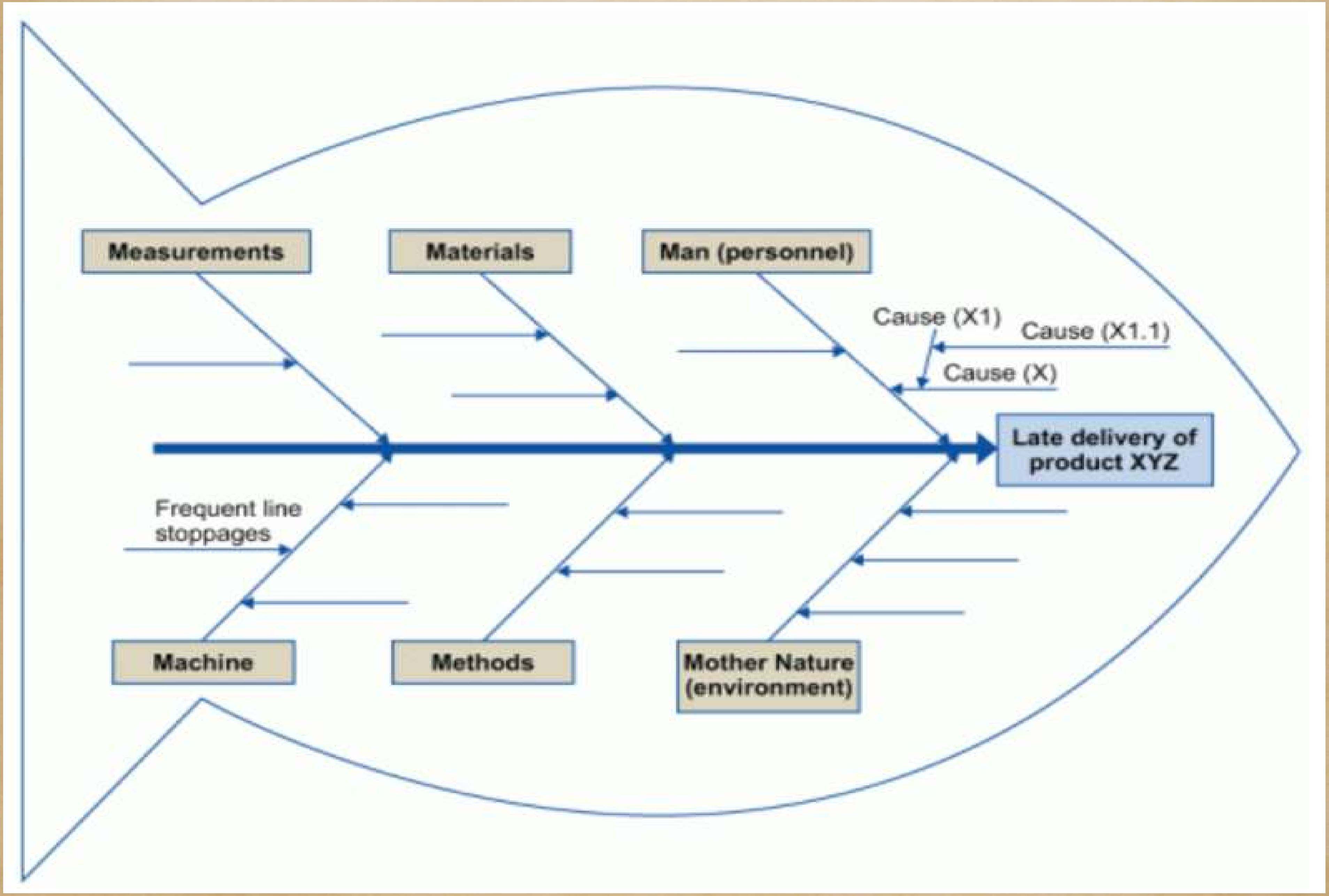
- *What can be done to prevent the problem from happening again?*
- *How will the solution be implemented?*
- *Who will be responsible?*

The 5 Whys

- An iterative technique that repeats the question "Why?" to identify the root cause of a problem
- Formally developed and used within Toyota (Japan)
- No hard and fast rules about lines of questioning or how long to keep looking for additional root causes
- Analysis can be tested by reversing the order and using "Therefore"

The 5 Whys at Leno's Garage

Fishbone Diagram



www.isixsigma.com/tools-templates/the-fundamentals-of-cause-and-effect-aka-fishbone-diagrams/

Group Exercise

- Let’s see if we can use Root Cause Analysis to help a classmate drill down on an impediment or issue
- We need 1 volunteer willing to present a problem or impediment to the group
- Together, we’ll then use the 5 Whys and/or Fishbone analysis to get to the potential root cause of the issue

Assignment #7:

When you get back to work, use Fishbone analysis and/or the 5 Whys to find the root cause of an organizational impediment, then resolve the issue.

Document this (and your reactions, learnings) in your personal journal.

TO DO	DOING	DONE
Scrum Leadership	Scaling Scrum / Agile Organization	Why & What of Agile and Scrum
		Scrum Leadership
		Facilitation
		Coaching
		Organizational Change
		Self-managing Teams
		XP & Kanban
		Agile Product Development
		Teaching

Scaling Scrum

Agile Organizations

What is Scaling?

What do we even mean by “Scaling” Scrum:

- Across a large product?
- Across an entire organization?
- Across distributed, geographic locations?
- All of these at once?!?

Why Scale?

What do we need to scale Scrum?

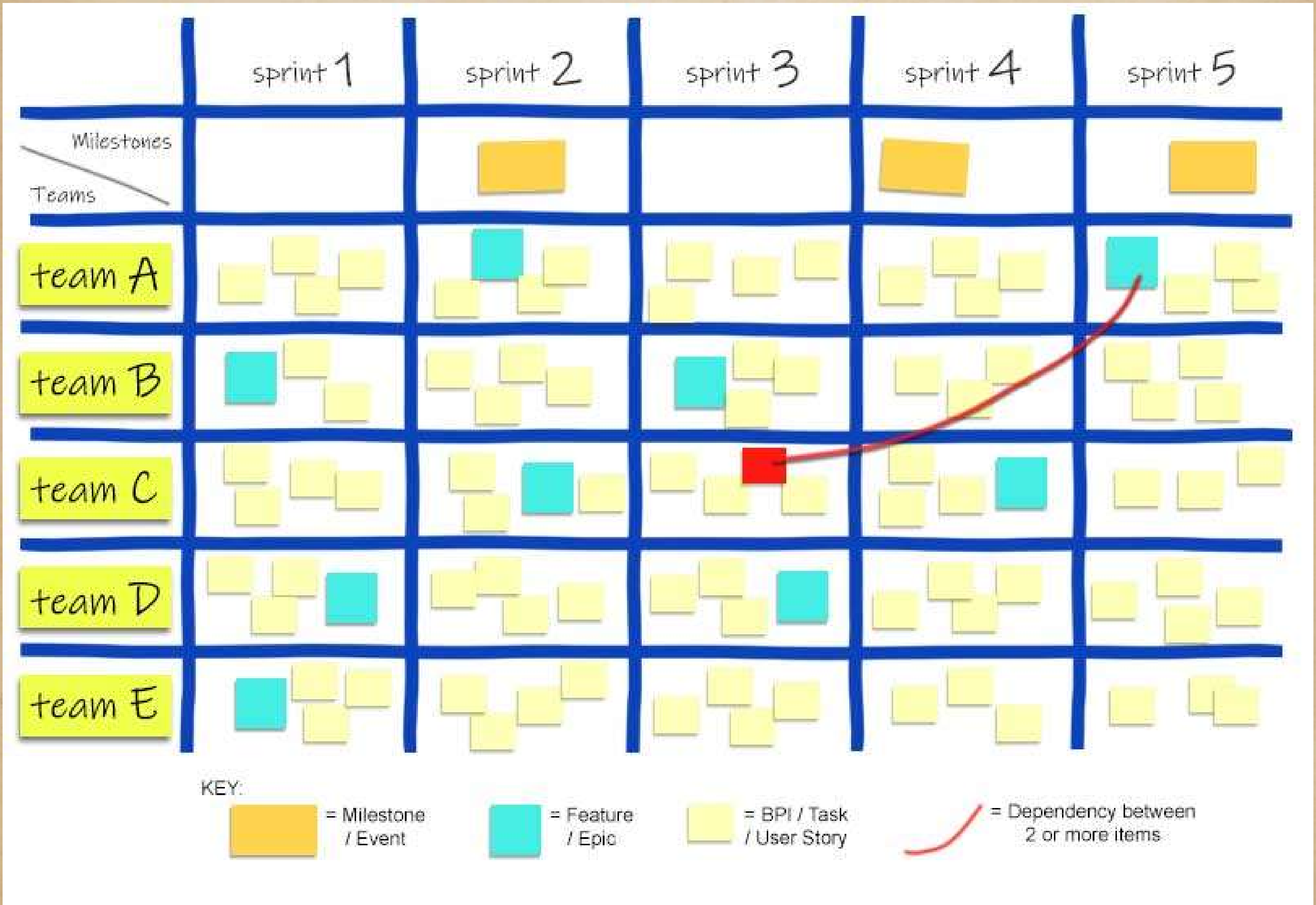
- Across a large product?
- Across an entire organization?
- Across distributed, geographic locations?
- All of these at once?!?

Why might an organization decide to NOT scale Scrum?

Dependencies

What are some examples of dependencies?

Make Dependencies Visible



Make Dependencies Visible

Source: @michael_p_stump

✓ Make Dependencies Visible

✓ Manage Dependencies

✓ Reduce Dependencies

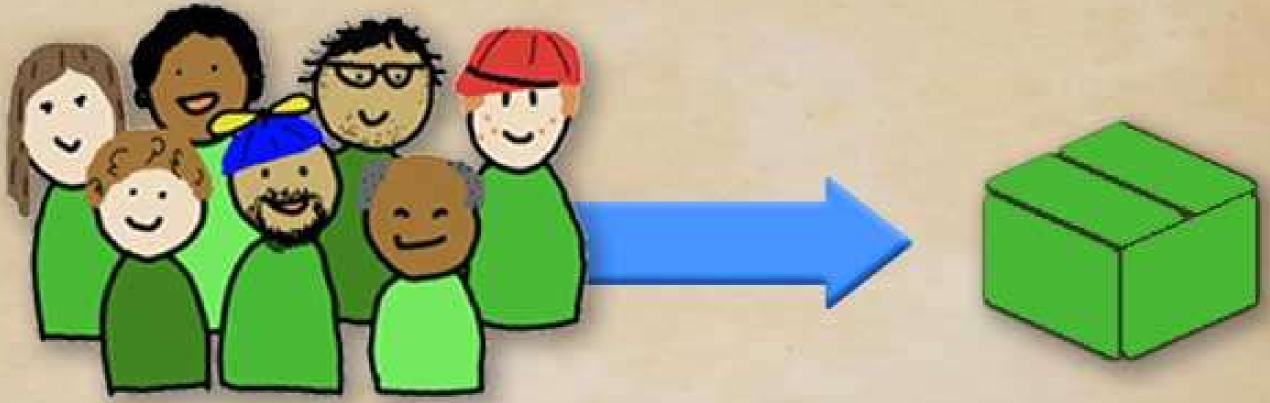
Feature vs. Component Teams

- A **feature team** is a cross-functional team that can pull end-customer features from the Product Backlog and complete them.

- A **component team** focuses on development of a component or subsystem that can be used to create only part of the end-customer feature
  - Component teams are also referred to as *asset* or *subsystem teams*

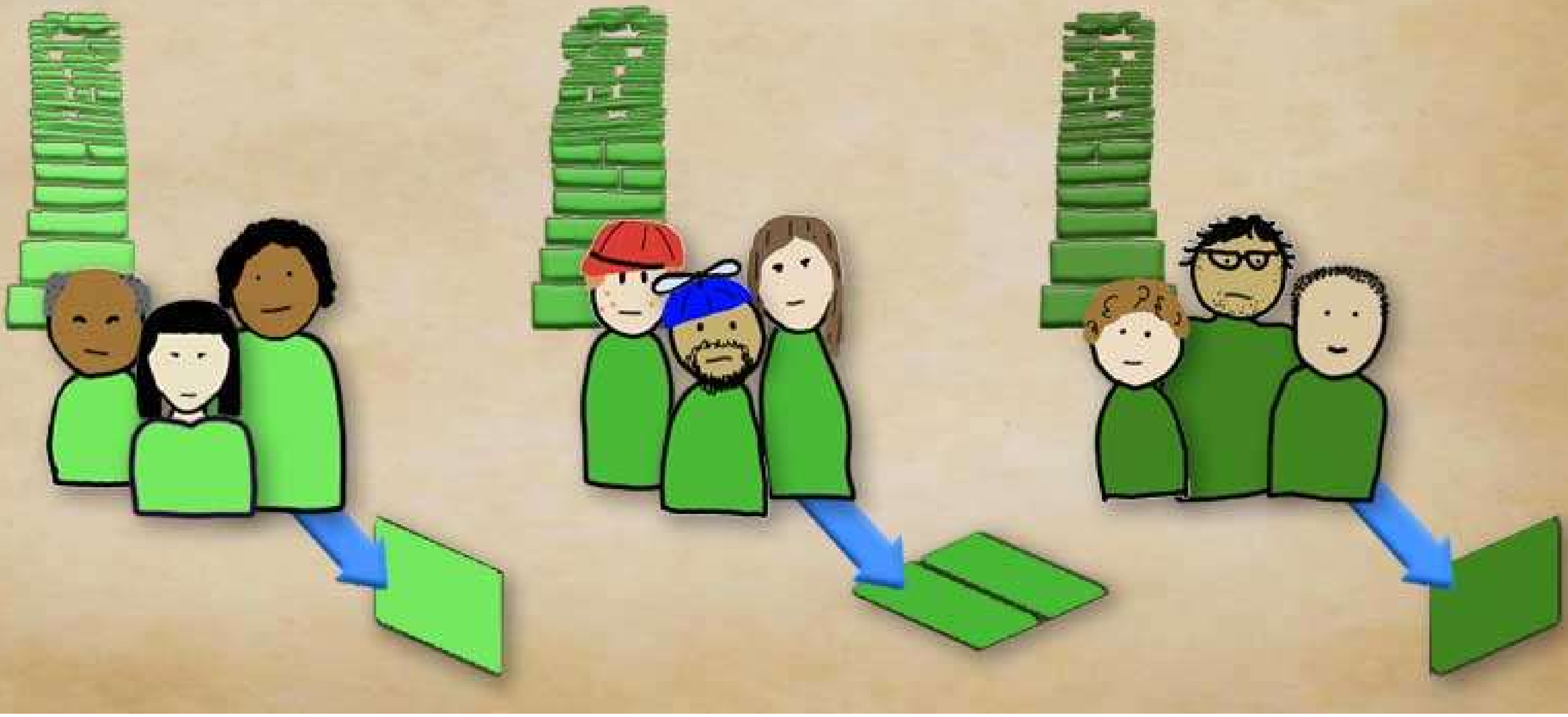
Feature Teams

- Feature teams have all the skills necessary to perform task-level work to deliver the entire PBI
- Team members would work on tasks related to the complete Product Backlog Item (PBI):
  - UI, middle-tier, database, etc.

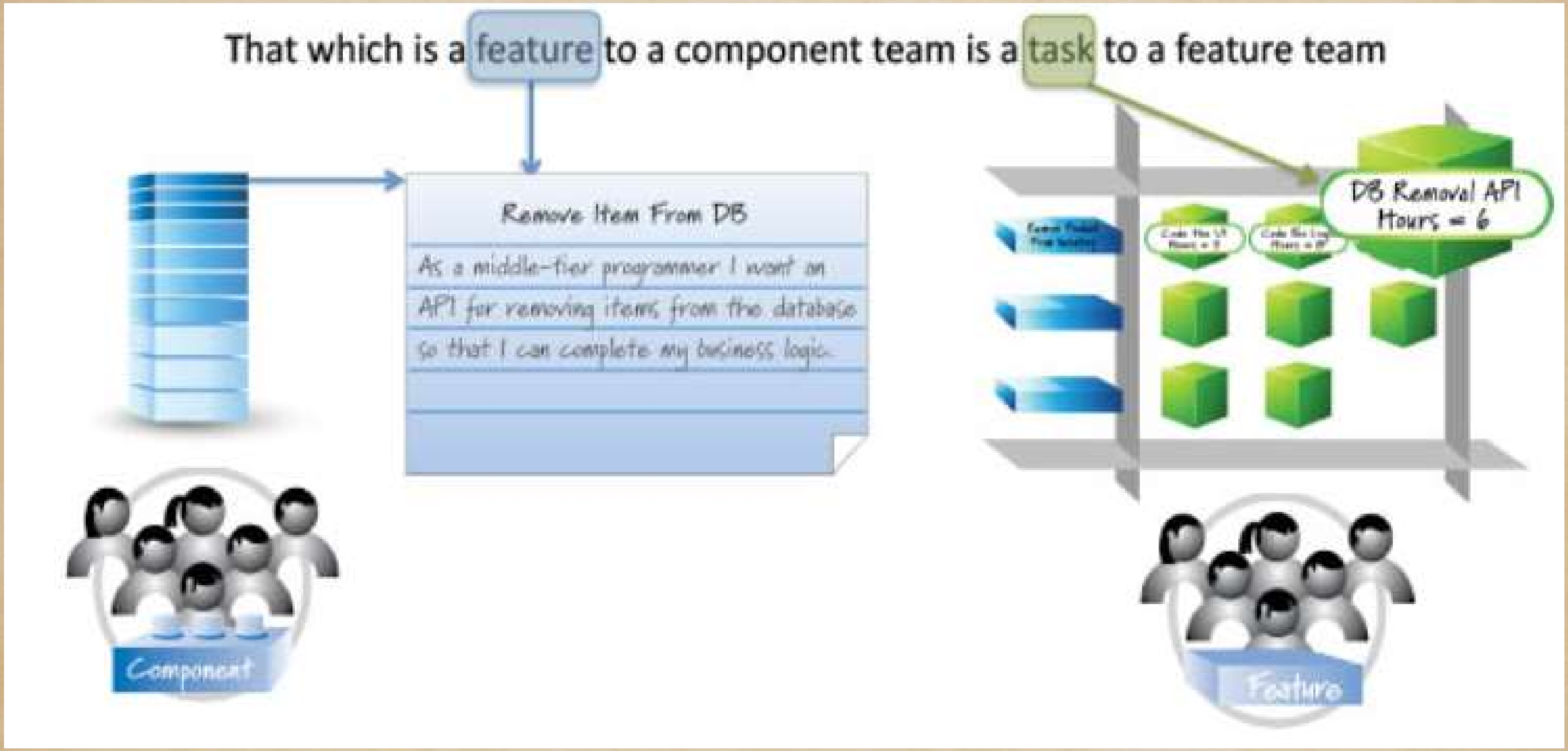


Component Teams

- No one team could deliver a given feature
- PBIs are created for each component team’s work
- Further task breakdown occurs to deliver the item



Component Teams



That which is a feature to a component team is a task to a feature team

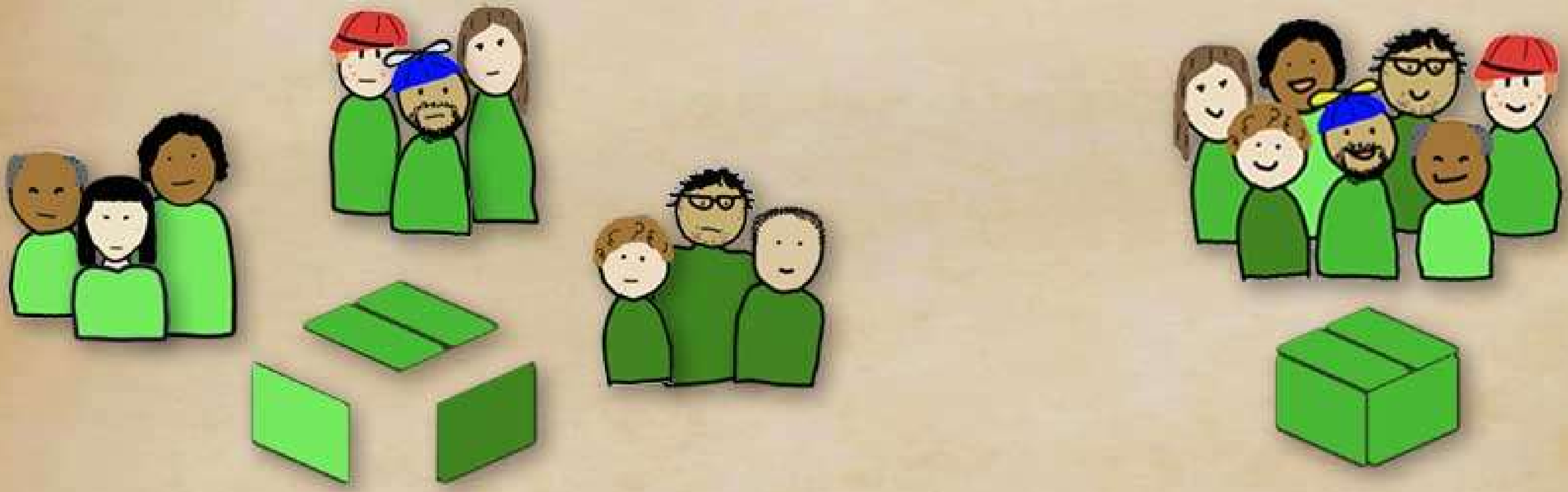
Remove Item From DB

As a middle-tier programmer I want an API for removing items from the database so that I can complete my business logic.

That which is a ***feature*** to a Component Team is a ***task*** to a Feature Team

Component vs. Feature Teams

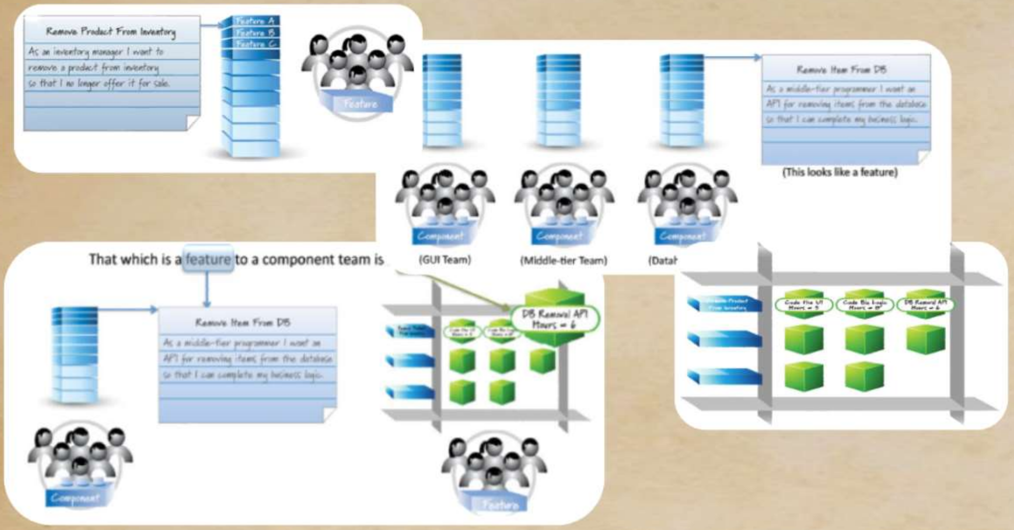
What are the trade-offs?

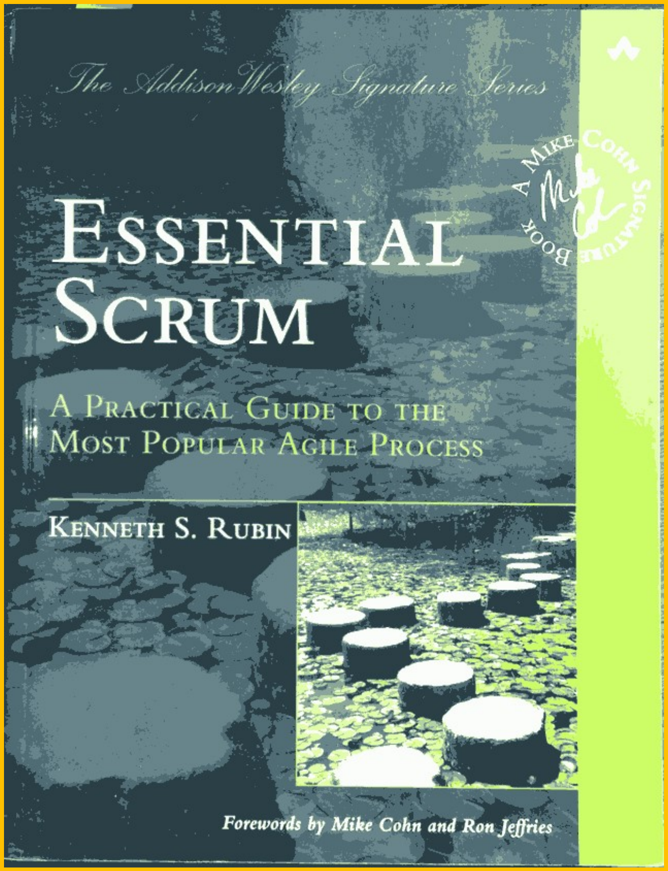


Feature vs. Component Teams

- Which structure optimizes for delivery: Feature Teams or Component Teams?
- Is the organization willing to form Feature Teams to enable delivery of product to customers?
- Is the organization willing to change existing team structures?
- What are the trade-offs?

Feature vs. Component Teams


Remove Product From Inventory
As an inventory manager I want to remove a product from inventory so that I no longer offer it for sale.
Feature A
Feature B
Feature C
Feature
Remove Item From DB
As a middle-tier programmer I want an API for removing items from the database so that I can complete my business logic.
(This looks like a feature)
Component
Component
Component
(GUI Team)
(Middle-tier Team)
(Datab
That which is a feature to a component team is
Remove Item From DB
As a middle-tier programmer I want an API for removing items from the database so that I can complete my business logic.
Component
DB Removal API
Hours = 6
Feature
Code the UI
Hours = 5
Code Biz Logic
Hours = 8
DB Removal API
Hours = 6

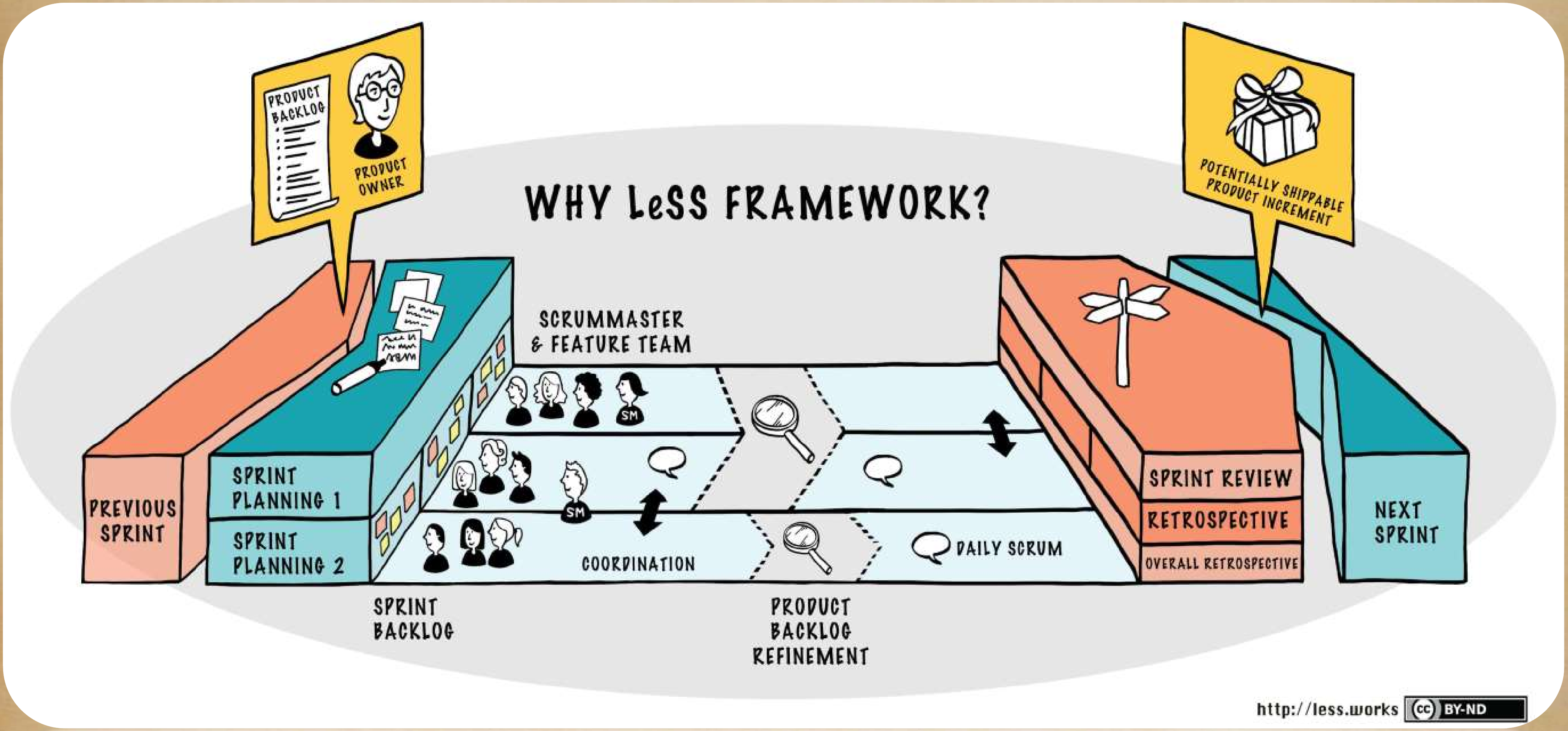

The Addison Wesley Signature Series
A Mike Cohn Signature Book
Essential Scrum
A Practical Guide to the Most Popular Agile Process
Kenneth S. Rubin
Forewords by Mike Cohn and Ron Jeffries

RECOMMENDED READING

Scaling Challenges

- How to be consistent with the Scrum Guide
- Slower delivery?
- Descaling – the elimination of bad habits – followed by incremental rollout of functioning team is critical
- Minimal viable bureaucracy (much coordination)

Large Scale Scrum (LeSS)



#LeSSWorks @less_works https://less.works

Large Scale Scrum (LeSS)

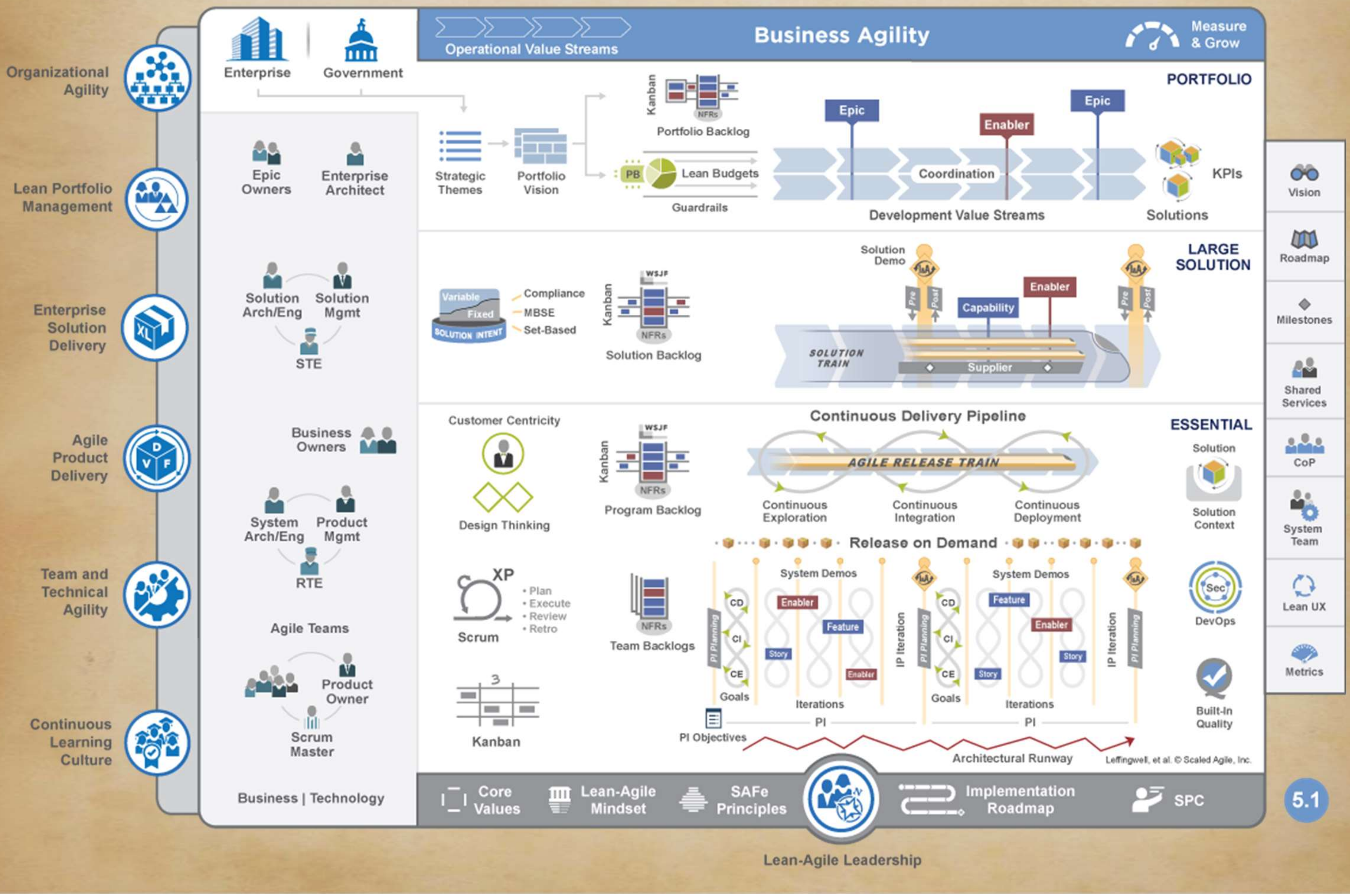
What is the Same as One Scrum Team?

- No separate analysis group, testing group, architecture group, user experience group, platform group, etc.
- No “tester” or “architect” within the team
- That implies the dissolution of existing single-function groups and the management supervising roles, and the elimination of traditional career paths and job titles
- The concept of “it is not ready until the end” dissolves
- Scrum is not for the programming phase after the analysis and before testing – the sequential lifecycle is eliminated
- There is no team lead or project manager that directs or tracks team members

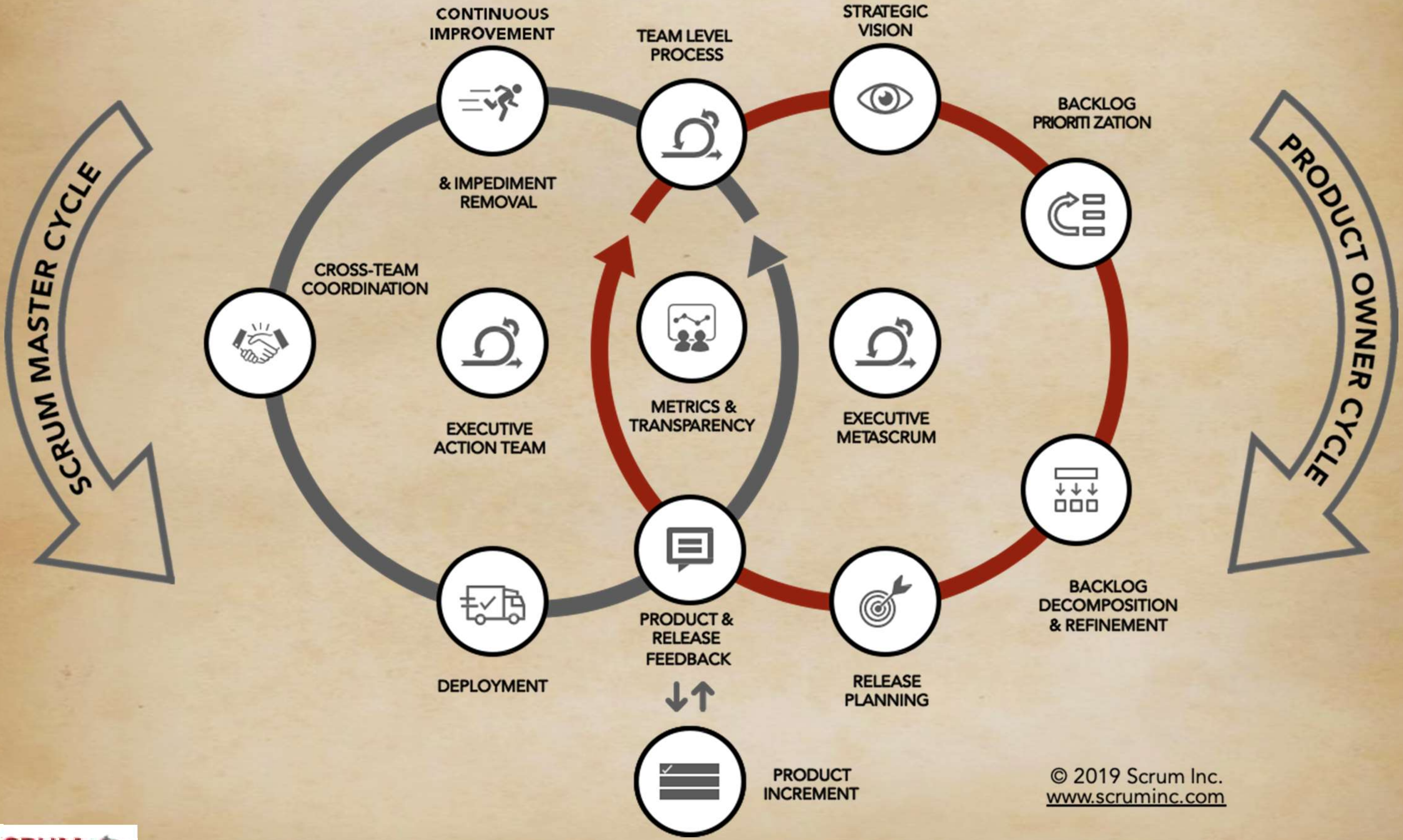
What is Different?

- For the roles, nothing
- Meetings and artifacts may change slightly

Scaled Agile Framework (SAFe)



Scrum at Scale (S@S)



© 2019 Scrum Inc.
www.scruminc.com

SCRUM @SCALE

www.scrumatscale.com

Scrum at Scale (S@S)

- Jeff Sutherland's 2016 approach to scaling Scrum
- A incremental approach to scaling Scrum that preserves Scrum's contextual and fractal nature
- A scale-free architecture for promoting organizational agility
- Includes a set of patterns that are effective for small scale Scrum and how to use effectively "larger scale"

www.scrumatscale.com

What are your (new) thoughts on scaling at your organization?

TO DO

- Scrum Mastery revisited

DOING

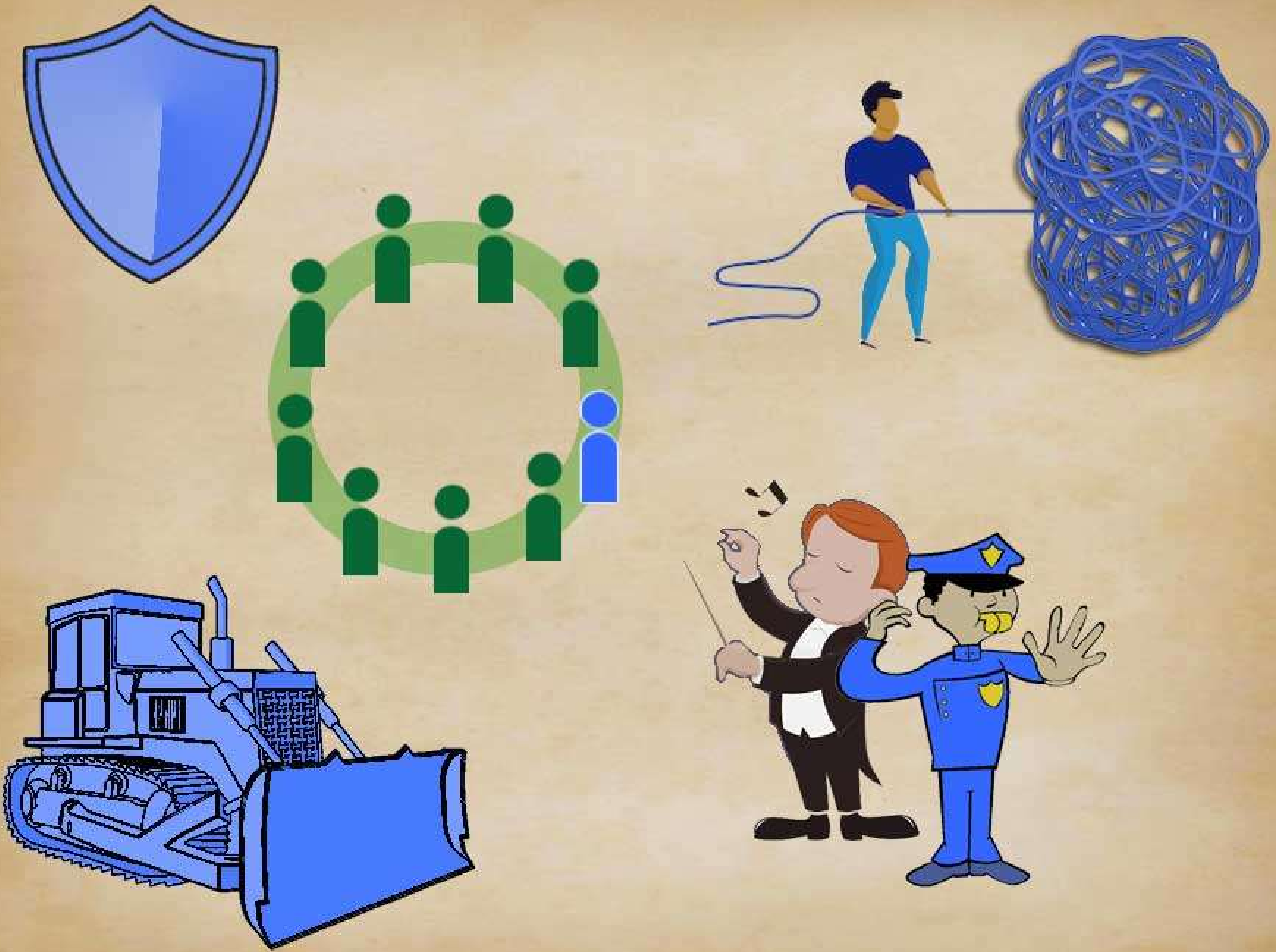
DONE

- Why & What of Agile and Scrum
- Scrum Leadership
- Facilitation
- Coaching
- Organizational Change
- Self-managing Teams
- XP & Kanban
- Agile Product Development
- Teaching
- Scaling Scrum / Agile Organization

Scrum Mastery

Module: 9

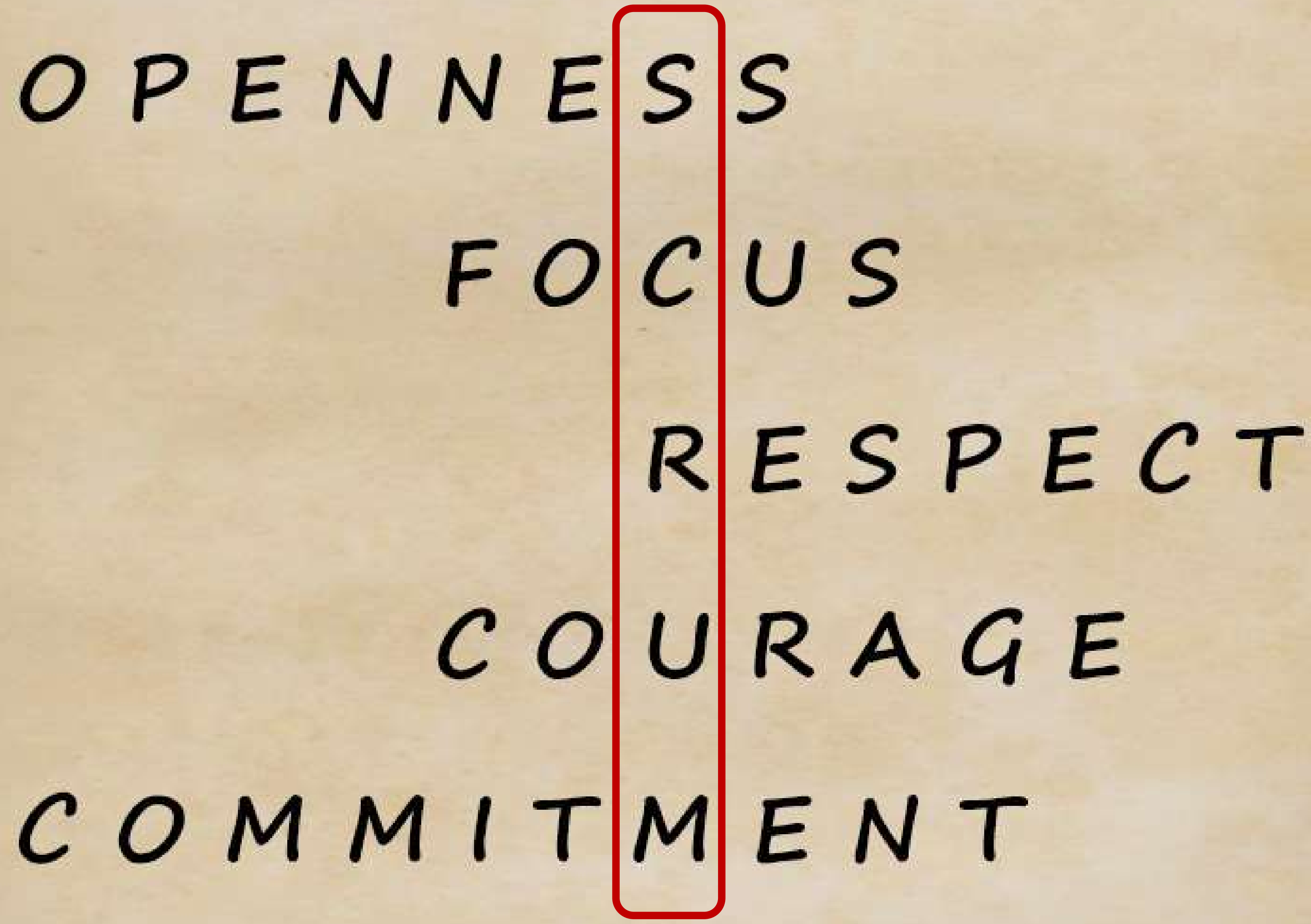
So you’re a Scrum Master …



Serving the Organization

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

Source: Scrum Guide



OPENNESS

FOCUS

RESPECT

COURAGE

COMMITMENT

https://www.scrumalliance.org/about-scrum/values

Self-Reflection

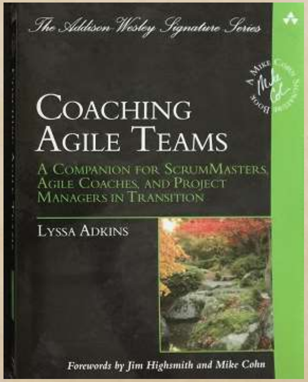
How well do you fulfill/embody the Scrum values?

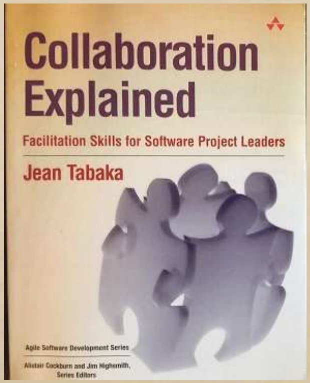
How you could improve?

Wrapping Up

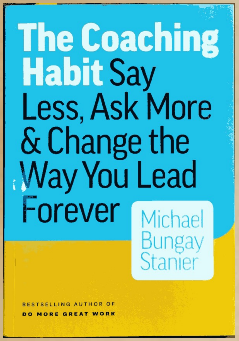
Recommended Reading

Coaching Agile Teams:
A Companion for
ScrumMasters, Agile
Coaches, and Project
Managers in Transition
- Lyssa Adkins


The Addison-Wesley Signature Series
Coaching Agile Teams
A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition
Lyssa Adkins
Forewords by Jim Highsmith and Mike Cohn

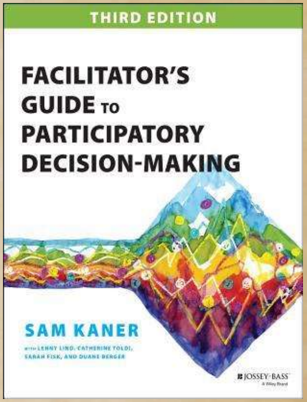

Collaboration Explained
Facilitation Skills for Software Project Leaders
Jean Tabaka
Agile Software Development Series
Alistair Cockburn and Jim Highsmith, Series Editors

Collaboration Explained:
Facilitation Skills for
Software Project Leaders
- Jean Tabaka

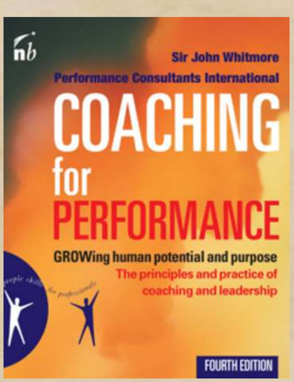

The Coaching Habit Say Less, Ask More & Change the Way You Lead Forever
Michael Bungay Stanier
BESTSELLING AUTHOR OF DO MORE GREAT WORK

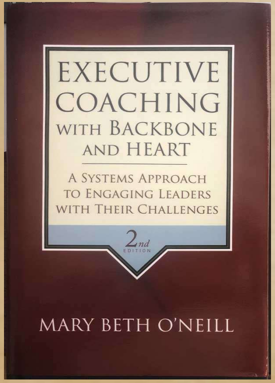
The Coaching Habit:
Say Less, Ask More & Change
the Way You Lead Forever
- Michael Bungay Stanier

Facilitator's Guide to
Participatory Decision-Making
- Sam Kaner



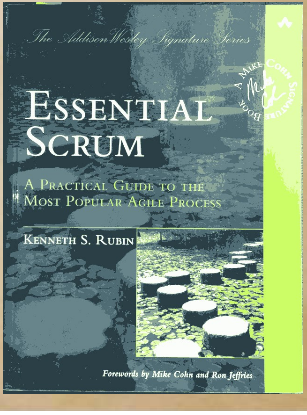
Coaching for Performance:
GROWing Human Potential and Purpose
- the Principles and Practice
of Coaching and Leadership
- John Whitmore


nb
Sir John Whitmore
Performance Consultants International
COACHING for PERFORMANCE
GROWing human potential and purpose
The principles and practice of coaching and leadership
FOURTH EDITION


EXECUTIVE COACHING WITH BACKBONE AND HEART
A SYSTEMS APPROACH TO ENGAGING LEADERS WITH THEIR CHALLENGES
2nd EDITION
MARY BETH O'NEILL

Executive Coaching with
Backbone and Heart:
A Systems Approach to Engaging
Leaders with Their Challenges
- Mary Beth A. O'Neill

Essential Scrum:
A Practical Guide to the
Most Popular Agile Process
- Kenneth S. Rubin


The Addison-Wesley Signature Series
Essential Scrum
A Practical Guide to the Most Popular Agile Process
Kenneth S. Rubin
Forewords by Mike Cohn and Ron Jeffries

Parking Lot

ROCKET NINE
SOLUTIONS

Do your Homework and Keep in Touch

Victor Bonacci
Certified Scrum Trainer

victor@rocketninesolutions.com

Agile
Coffee