



**KILL ALL METHODS
FREE THE PRACTICES**

Abstract

Kill All Methods - Free the Practices

The way we develop software struggles to keep pace with changes in technology and business. Even with the rise of agile we still see people flip-flopping from one branded method (or to be more fashionable framework) to another throwing away the good with the bad, and behaving more like religious cultists than scientists.

Can we truly enable and empower our teams and become true learning organizations whilst we behave more like the fashion industry than an engineering profession? Can we really see our ourselves as an open, diverse and collaborative community whilst we continually attack one another and rebrand, reinvent and rename everything like old hipsters trying to stay in with the in-crowd. Are we doomed to be locked in a never ending method war in the hope that the one true way emerges to rule them all?

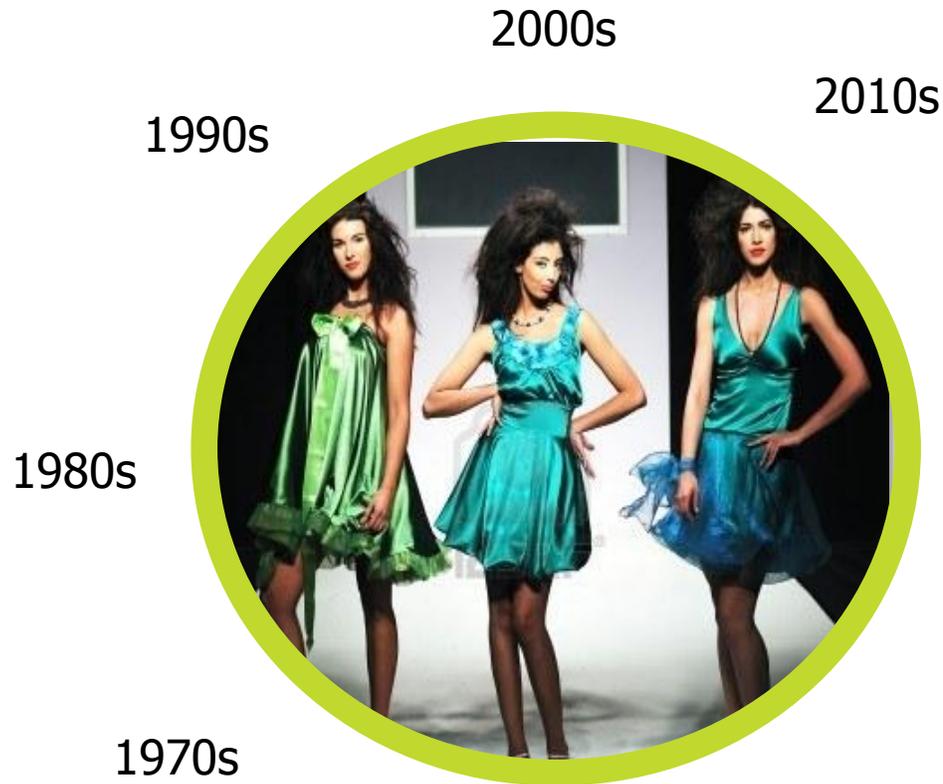
The answer is no. There is a simple way to break out of this cycle of unhealthy competition between methods that are more similar than they are different, and that is to free the practices. Free the practices to rise and fall based on their own merits. Free the practices so that teams can experiment, innovate and plug and play with proven practices to create the way-of-working that they need today and seamlessly evolve into the one they need tomorrow.

In this presentation Dr Ivar Jacobson will revisit the history of methods, explain why we need to break out of our repetitive dysfunctional behavior, and introduce Essence: a new way of thinking that promises to liberate the practices and enable true learning organizations.

AGENDA

1. How mature are we? 😊
2. What we need to do
3. What we get
4. What we do
5. Take aways

ARE WE DRIVEN BY FASHION?



Today?

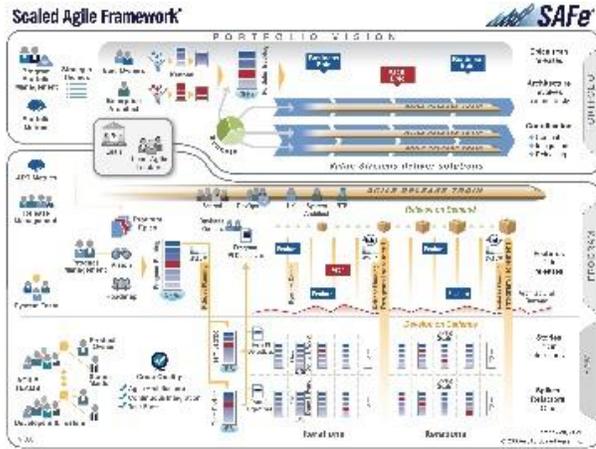
Future?

- The first published methods separated functions from data
- 25 years ago it was all about OO
- 20 years ago it was about components, UML, Unified Process (RUP)
- 15 years ago it was about CMMI
- Then we got Agile (XP, Scrum, Lean, Kanban, and more)
- Now it is Scaling Agile
- Tomorrow ???

All had/have something good – so that is not the problem

THE MOST ADOPTED METHODS FOR SCALING AGILE ARE:

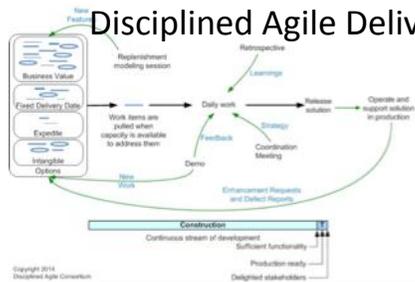
SAFe



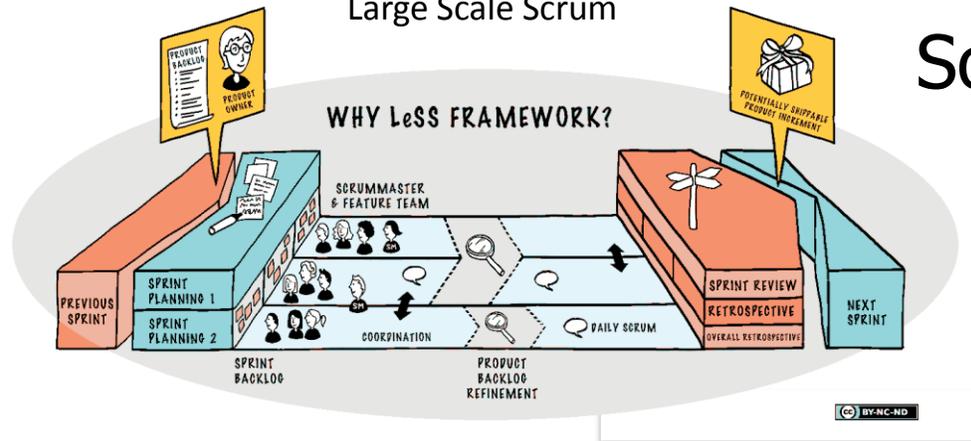
Nexus



Disciplined Agile Delivery



Large Scale Scrum

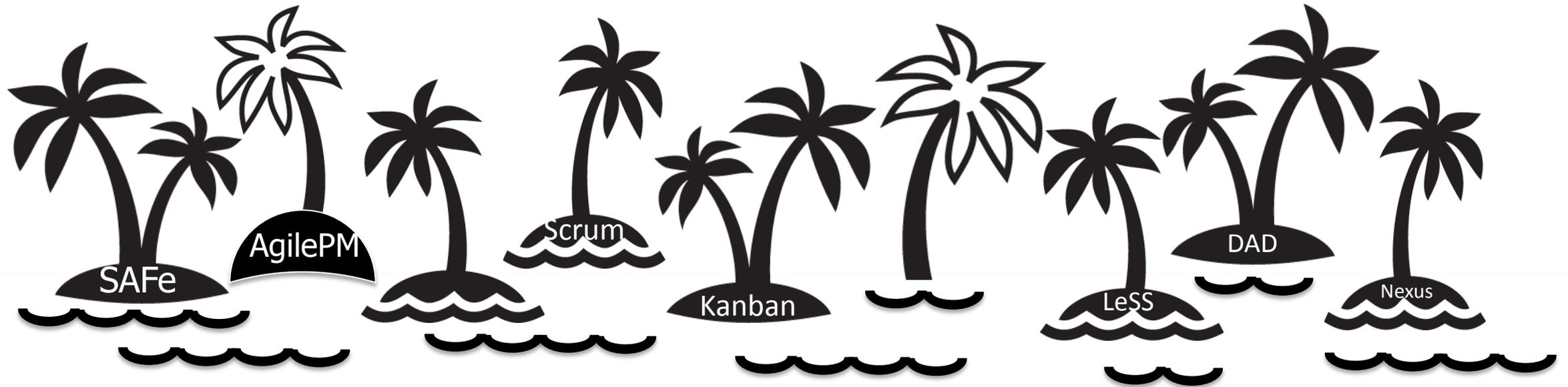


- All widely used
 - All provide great value when adopted
 - All have great practices
- So that is **not the problem**

So What is the Problem?

ALL METHODS ARE MONOLITHIC

- They are non-modular
- You can't mix and match practices from different methods

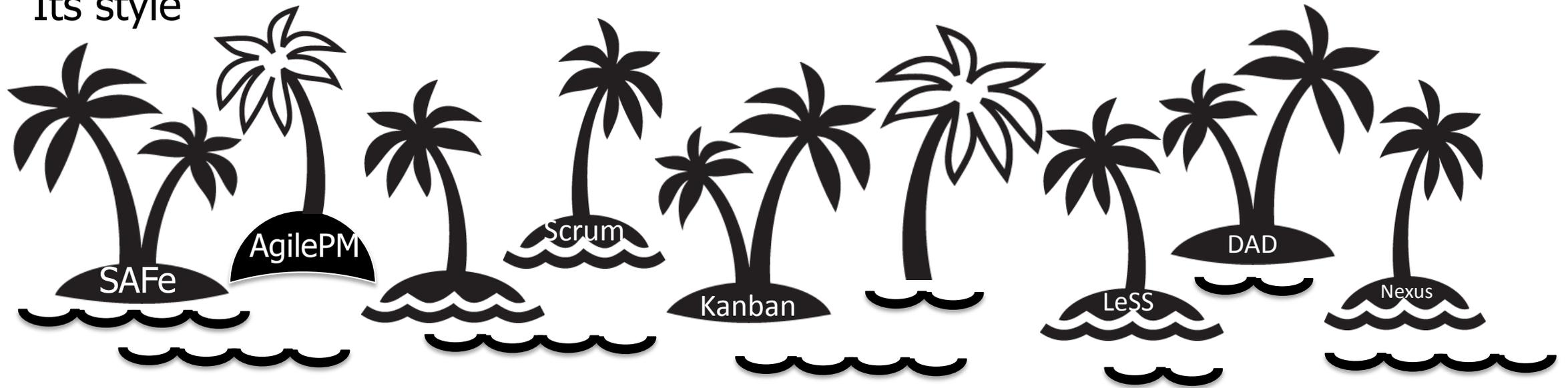


Methods are like isolated islands

EVERY METHOD IS HOMEGROWN

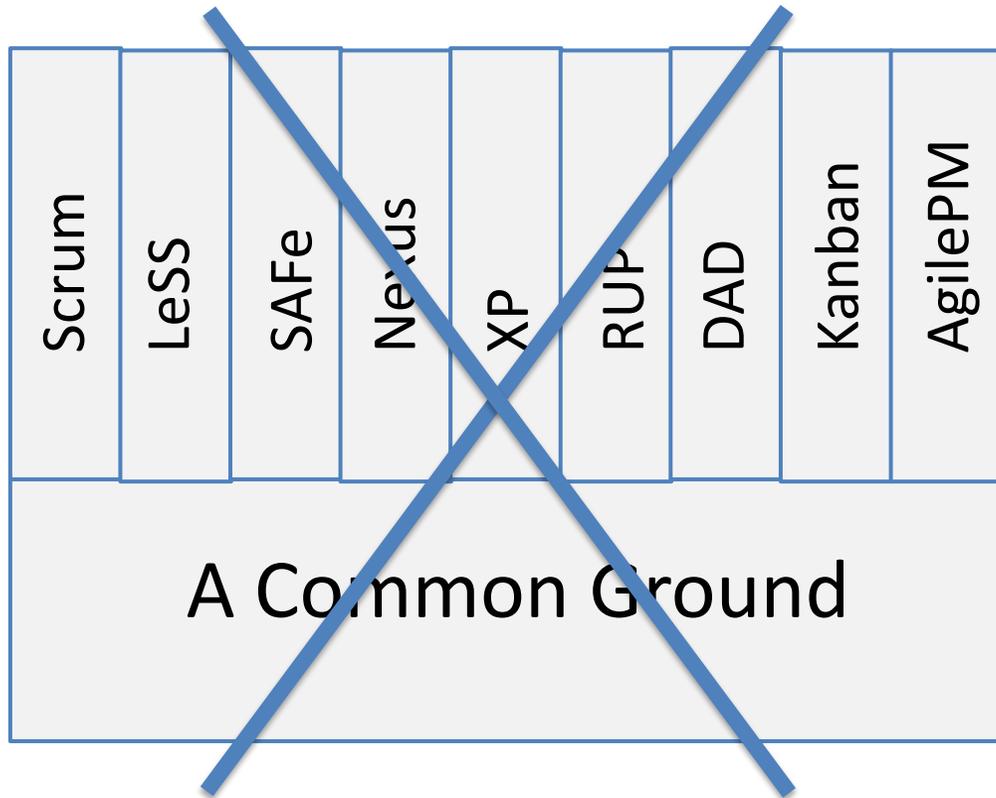
Everything about the method is unique

- Its user experience
- Its structure
- Its terminology
- Its style



Comparing/mixing methods is like
comparing/mixing cultures

METHODS HAVE NO COMMON GROUND



- They all deal with software so they should share a lot
- Fact is, they share almost nothing, not even the basics:
 - What is Software?
 - What is Software development?
 - What is Requirements, Design, Test?
 - What is Team, Way of Working?

PRACTICES ARE LOCKED IN METHOD PRISONS



- There are 100,000+ methods in the world
- Every method is a group of some smaller mini-methods, called practices
- There are only a few 100s of practices
- Like the ingredients in a soup the practices are not separable and not reusable – they are stuck in the method

Practices can't be reused to create other methods

METHOD PRISONS ARE CONTROLLED BY METHOD WARDENS - GURUS



- The guru controls which practices goes into her method
- Is forced to “steal/borrow” practices from other methods
- Is forced to rewrite “stolen/borrowed” practices to make them fit within her method and
- in the process “improves/misunderstands” them

- No one is an expert on everything (in a good method)
- But you can be an expert on something (such as a practice)

WE ARE AT A METHODS WAR FOR 50 YEARS



- Lots of good practices never make it
 - Because not seen/understood by any guru
- Winner is the best Marketeer
 - It is 80% Marketing, 20% Technology

Method gurus are forced to “steal” instead of collaborate

THESE ARE SOME PROBLEMS

- All methods are monolithic
- Every method is homegrown
- Methods have no common ground
- Practices are locked in method prisons
- Method prisons are controlled by method gurus
- We are at a methods war for 50 years

This is immature and foolish!!!

AGENDA

1. How mature are we? 😊
2. What we need to do
3. What we get
4. What we do
5. Take aways

HOW DO WE STOP THIS CRAZINESS?

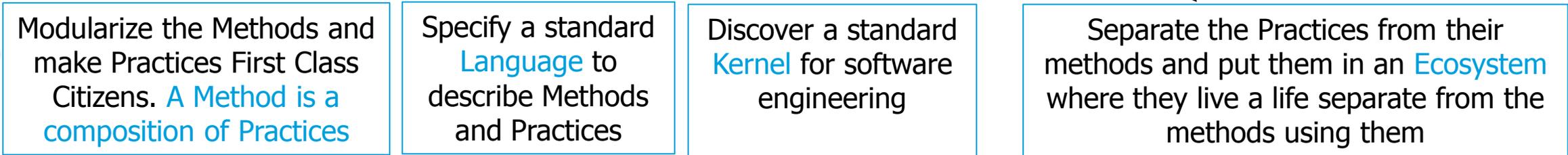
PROBLEMS



Free the Practices

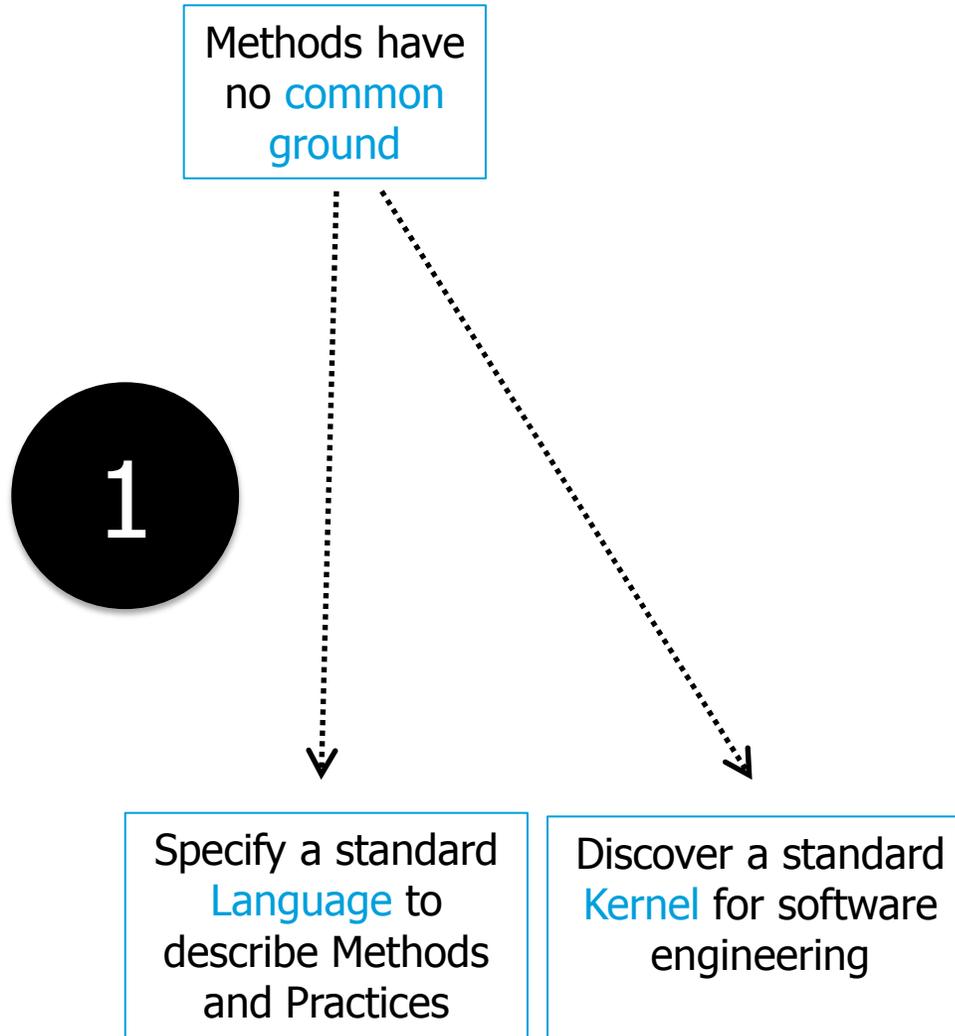
Kill the Methods

SOLUTIONS



1) START WITH THE COMMON GROUND!

PROBLEMS



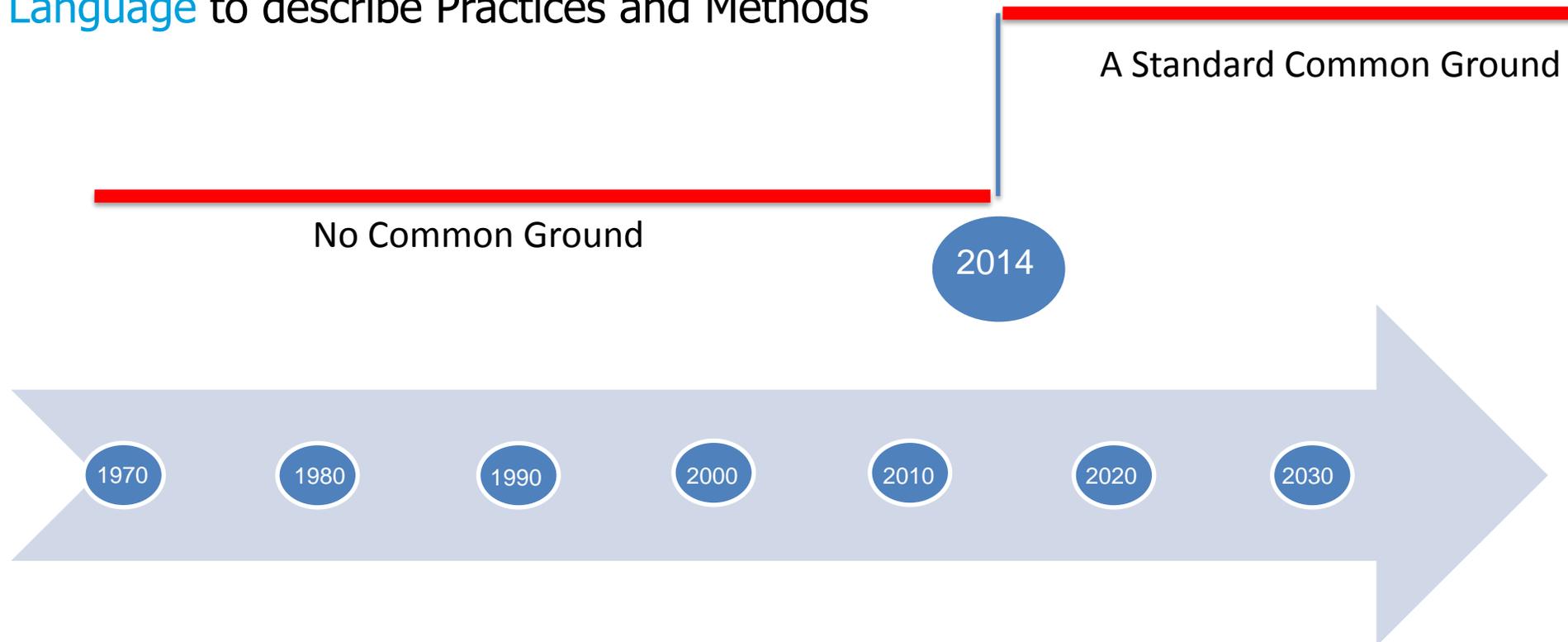
SOLUTIONS

DISCOVER A COMMON GROUND

LANGUAGE + KERNEL

What is a Common Ground?

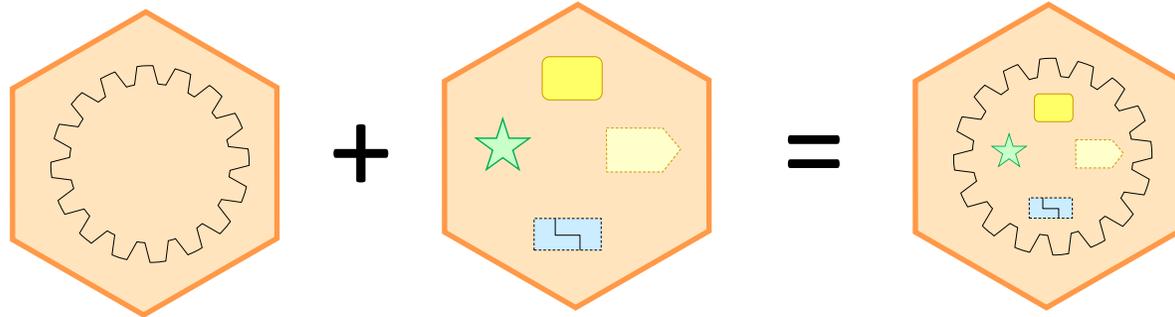
- It is a **Kernel** of elements including the essential things to work with, the essential things to do and the essential competencies you need.
- It is a **Language** to describe Practices and Methods



THERE IS A STANDARD COMMON GROUND: ESSENCE

For the first time in the 60+ year history of software engineering, we have got a Common Ground

Kernel Language Essence

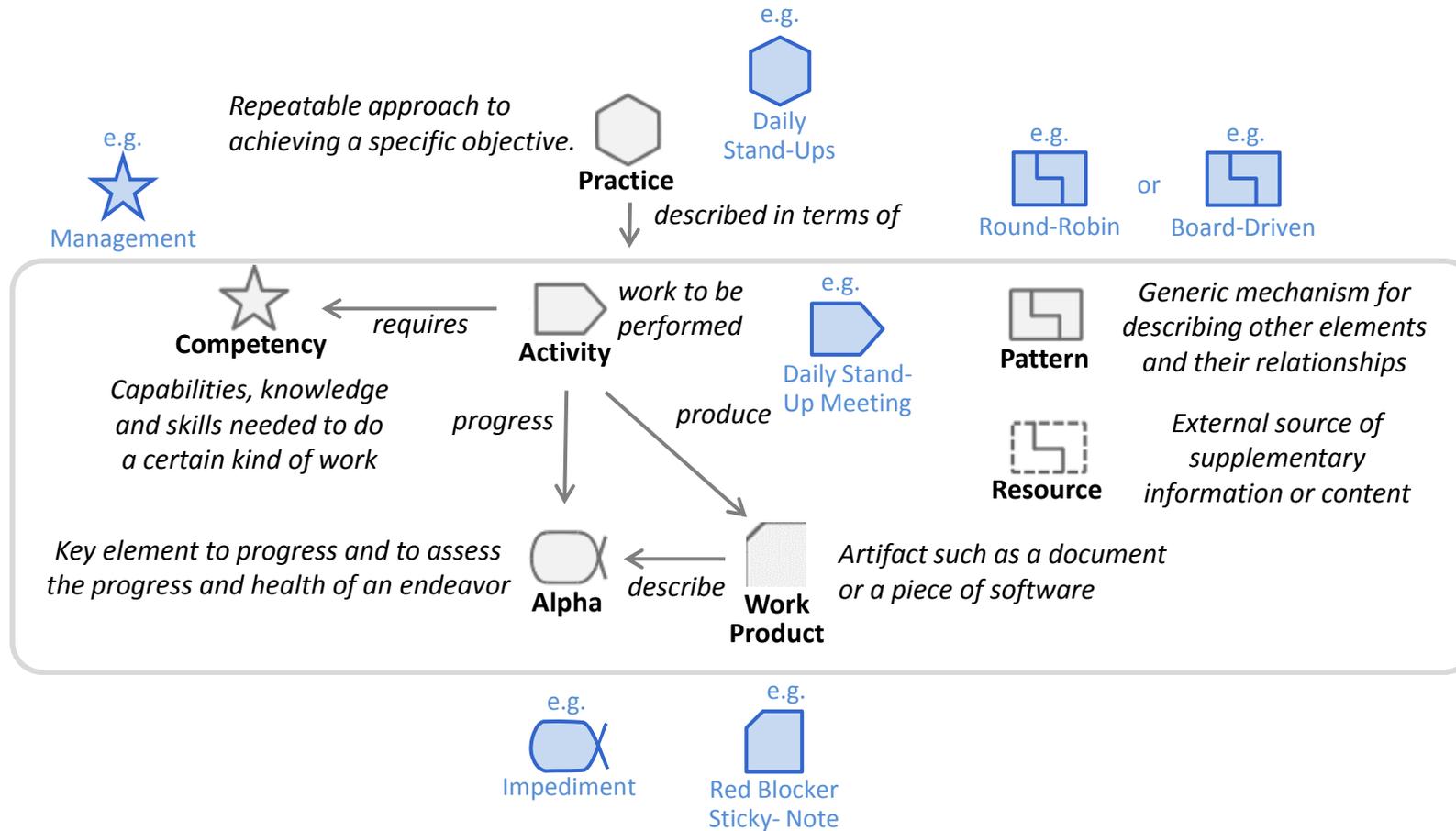


- Essential Things to Work with
- Essential Things to Do
- The Essential Competencies you need

- Visual language
- Simple
- Intuitive

THE COMMON
GROUND

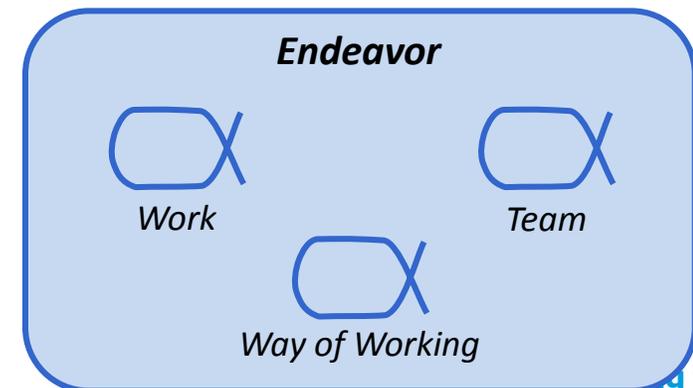
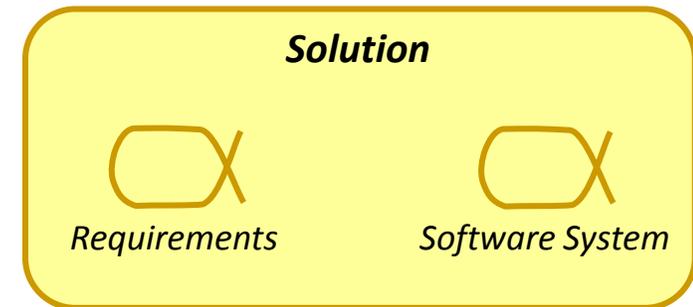
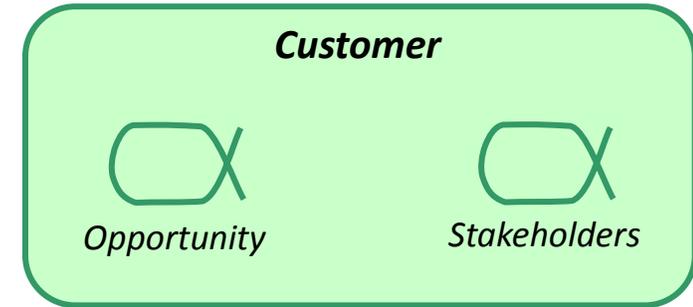
THE ESSENCE LANGUAGE: DESCRIBE METHODS AND PRACTICES



THE KERNEL FOR SOFTWARE ENGINEERING

The Alphas: The Most Important Things

- There are **customer** needs to be met
 - Someone has a problem or **Opportunity** to address
 - There are other **Stakeholders** who will fund, use and benefit from the solution produced
- There is a **solution** to be delivered
 - There are certain **Requirements** to be met
 - There'll be a **Software System** to develop
- There is an **endeavor** to be undertaken
 - We need to kick off the **Work** ...
 - Build an empowered **Team** of good people ...
 - With a good, responsive **Way of Working**



ALPHA CARDS

Opportunity

The set of circumstances that makes it appropriate to develop or change a software system.

Identified

Solution Needed

Value Established

Viable

Addressed

Benefit Accrued



 **IVAR JACOBSON**
INTERNATIONAL
Generated by IJ Practice Workbench™

1.1.2

Software System

A system made up of software, hardware, and data that provides its primary value by the execution of the software.

Architecture Selected

Demonstrable

Usable

Ready

Operational

Retired



 **IVAR JACOBSON**
INTERNATIONAL
Generated by IJ Practice Workbench™

1.1.2

“Product development is the process of converting uncertainty to knowledge”

ALPHA STATE CARD CHECKLISTS



2) HOMEGROWN METHODS ARE GONE

PROBLEMS

Every method is
homegrown

2

WE HAVE A STANDARD LANGUAGE

SOLUTIONS

Specify a standard
Language to
describe Methods
and Practices

3) MODULARIZE THE METHODS

AND MAKE PRACTICES FIRST CLASS CITIZENS

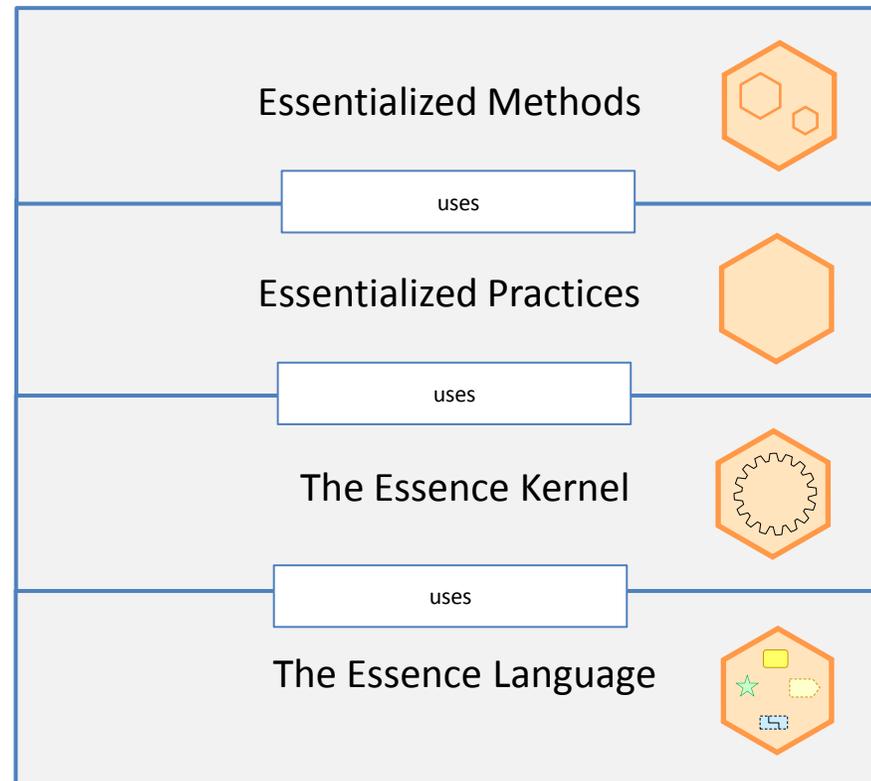
PROBLEMS

All methods are monolithic

3

SOLUTIONS

Modularize the Methods and make Practices First Class Citizens. A Method is a composition of Practices



Methods like SAFe, DAD, SPS, LeSS, etc.

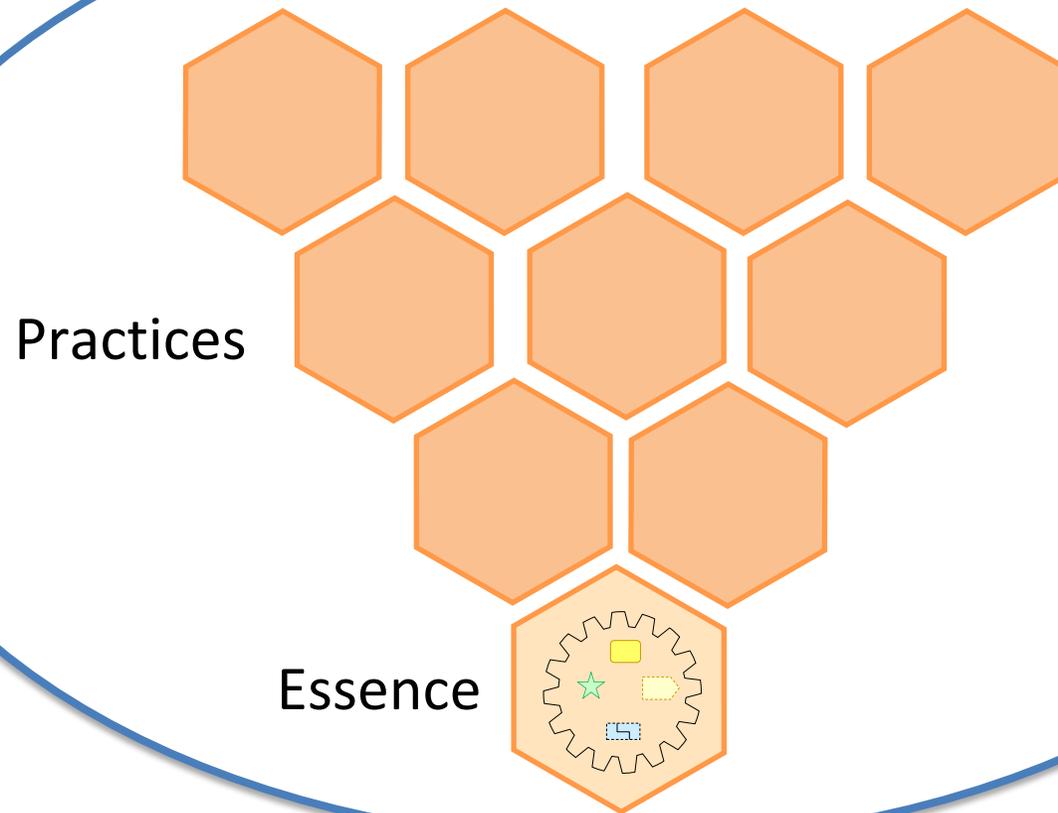
Generic practices: Scrum, User Story, Use Cases, etc.

Specific practices: Practices from SAFe, DAD, SPS, LeSS, etc. – All in a practice library

The Essence Common Ground

THUS THIS IS AN ESSENTIALIZED METHOD

A Method



Essence makes Methods Modular – not Monolithic

KILL ALL METHODS – FREE THE PRACTICES?

- All methods not yet essentialized give you huge problems (method prisons, method gurus, method wars, etc.) so they need to be killed...
 - By freeing the practices locked into them - essentializing them, and
 - By letting the methods resurrect by composing its essentialized practices - resulting in essentialized methods.

Of course we want all the good methods back,
but deliberated from the existing problems

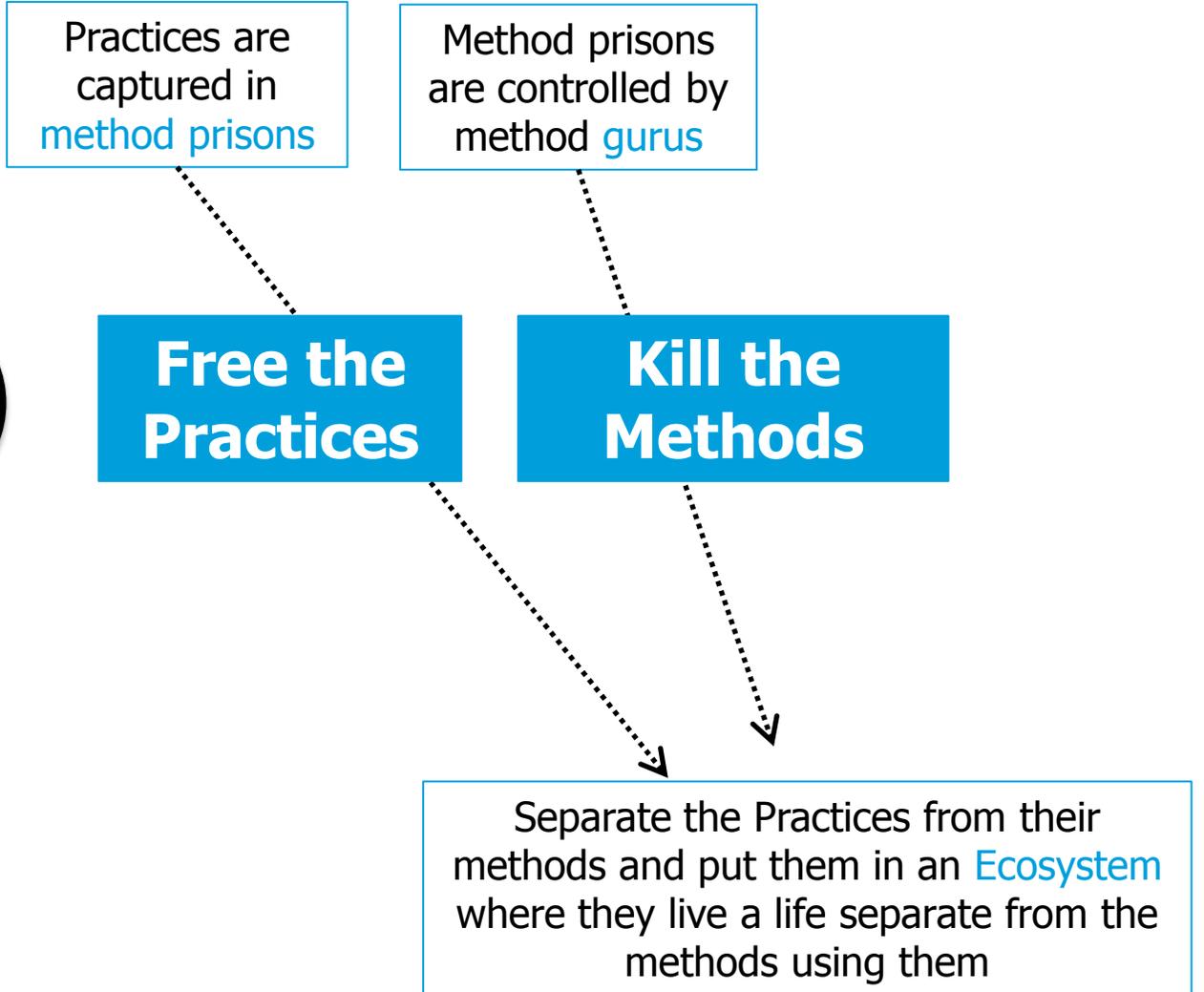
4) CREATE AN ECOSYSTEM

PROBLEMS

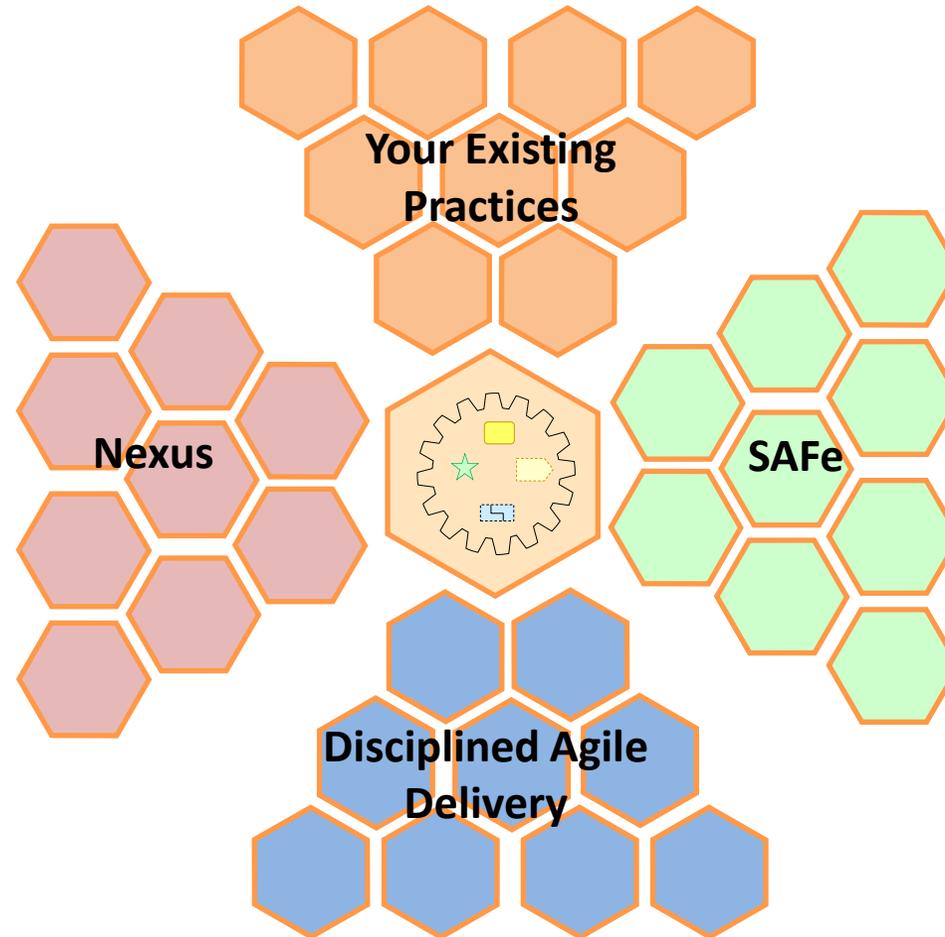
**PRACTICES LIVE
SEPARATE FROM
THE METHODS
USING THEM**

SOLUTIONS

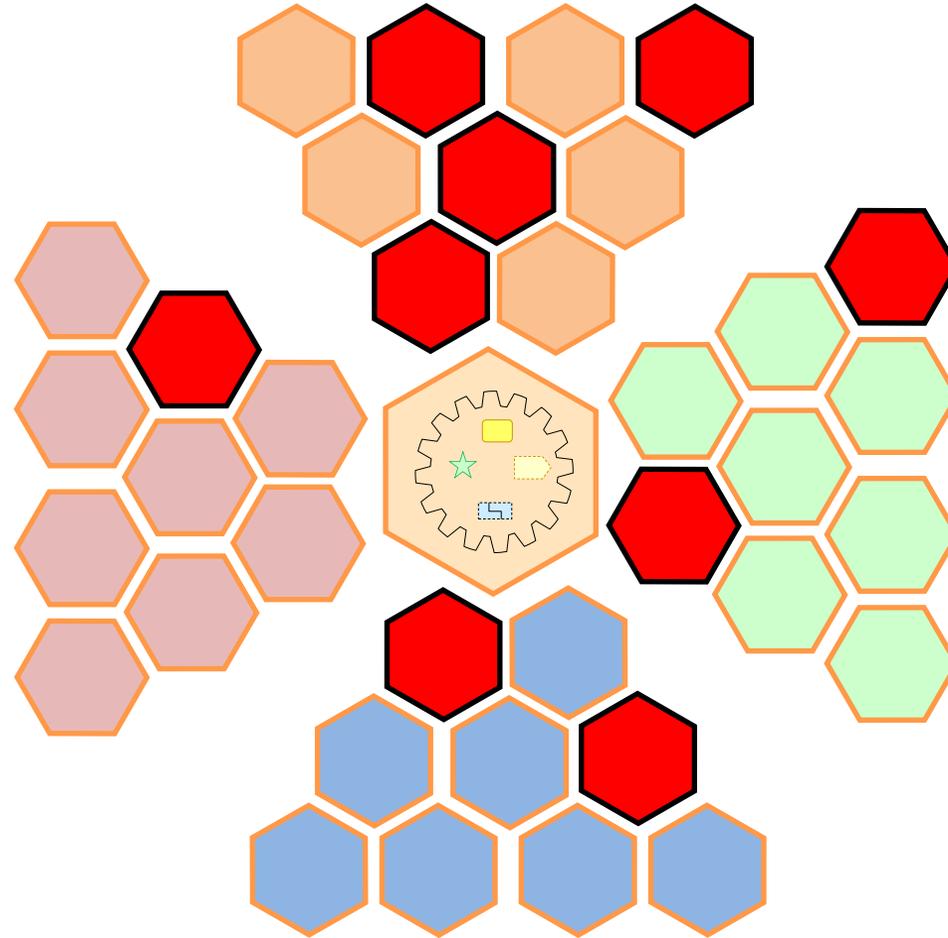
4



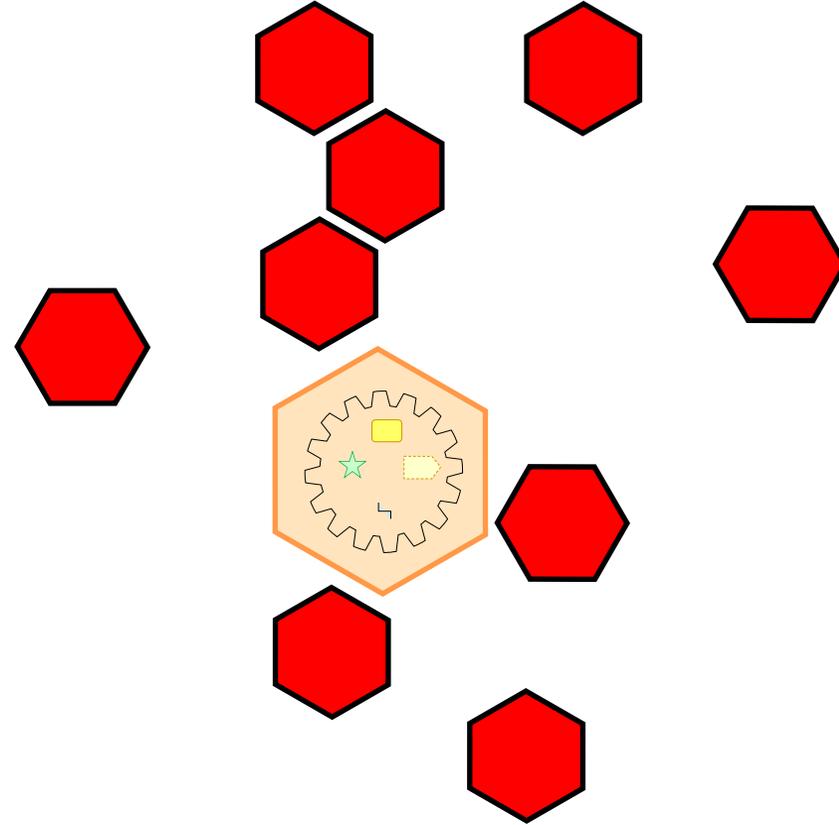
IMAGINE A PRACTICE ECOSYSTEM



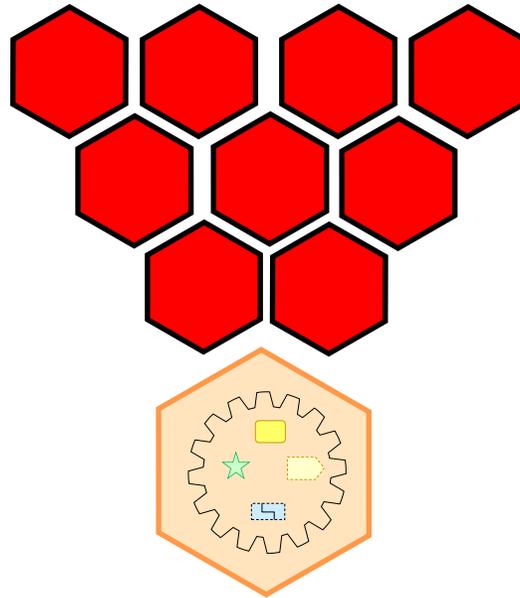
YOU SELECT THE PRACTICES YOU LIKE



IGNORE THE ONES YOU DON'T NEED



YOU CREATE YOUR OWN METHOD



...mixing and matching practices to create your own method

AGENDA

1. How mature are we? 😊
2. What we need to do
3. What we get
4. What we do
5. Take aways

EVERY TEAM CAN SELECT THEIR OWN PREFERRED METHOD

METHODS BECOME LIGHT

ORGANIZATION BECOMES
A LEARNING ORGANIZATION,

It becomes easy to:

- Teach
- Learn
- Get up to speed
- Change

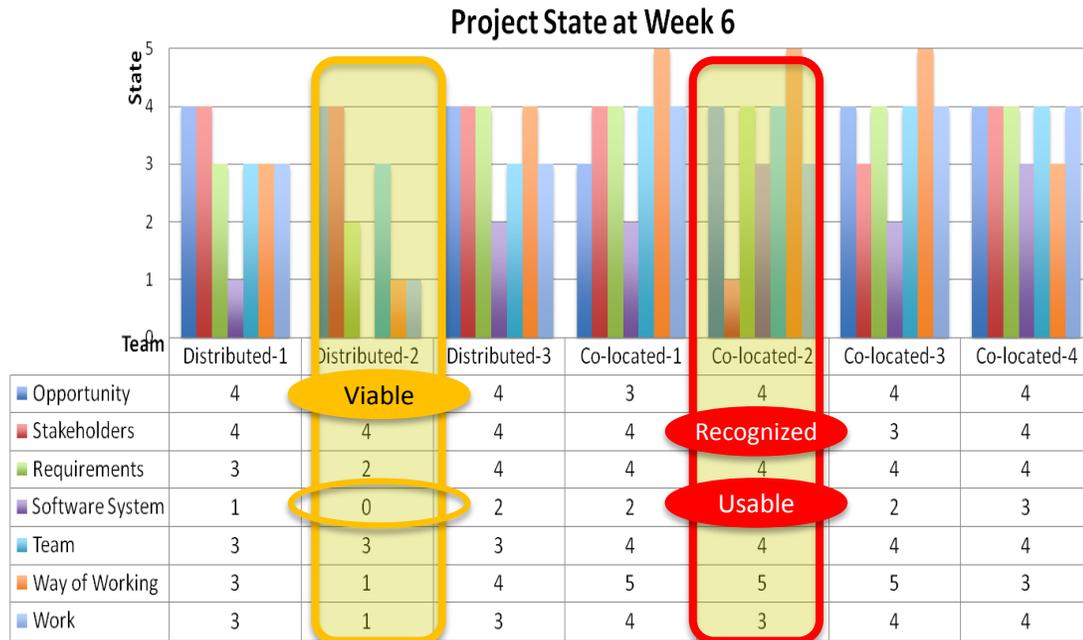
**CONTINUOUS
IMPROVEMENT**

Practices improved
continuously by users
from everywhere

Practices can come from
everywhere

Selected from a library of
practices

PROGRESS AND HEALTH OF PROJECT ARE MEASURABLE

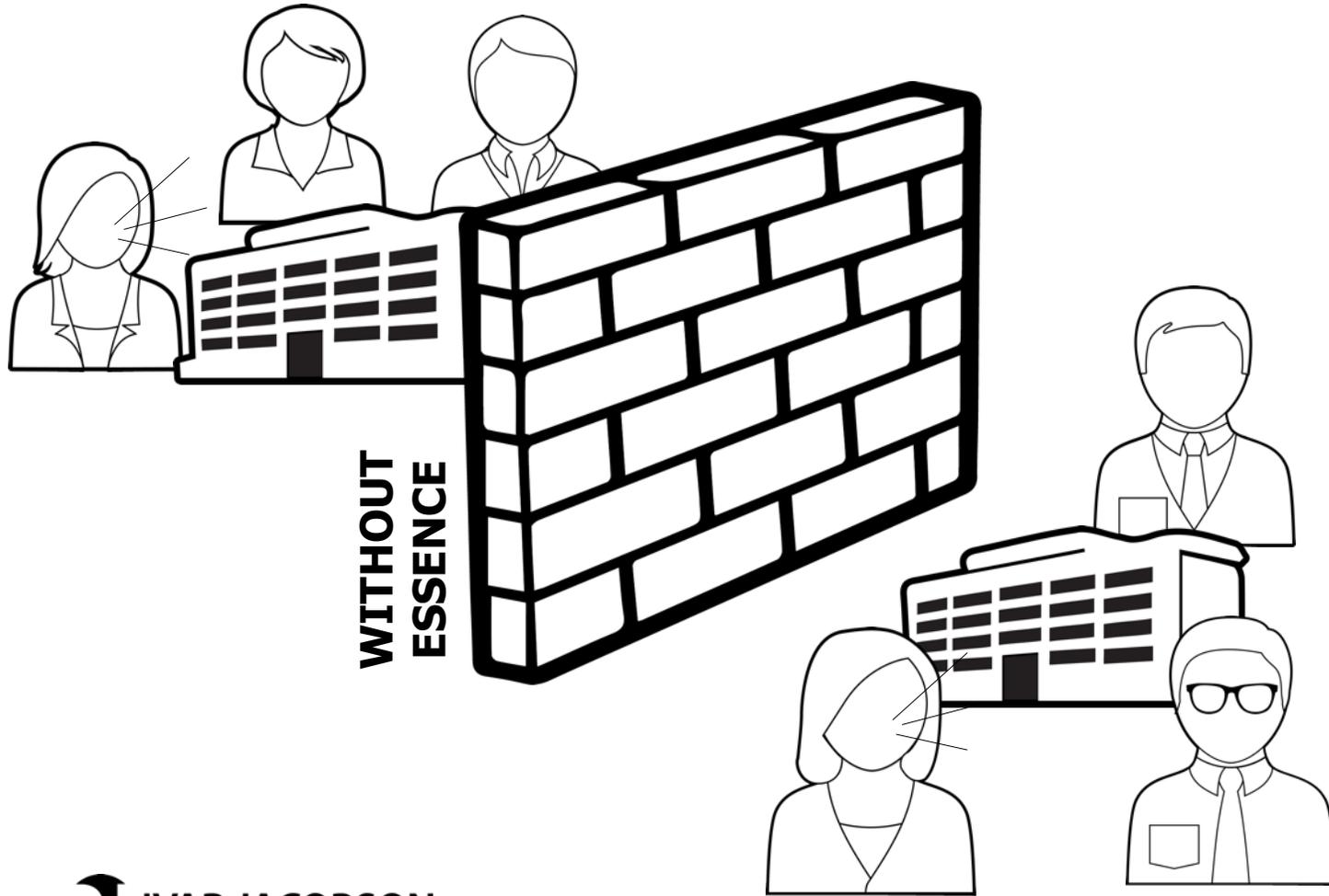


- **INDEPENDENT ON WHICH METHOD IS BEING USED**
- Imagine a project manager with many teams
- Imagine a product manager with many projects
- Imagine a program manager with many products

Carnegie Mellon®

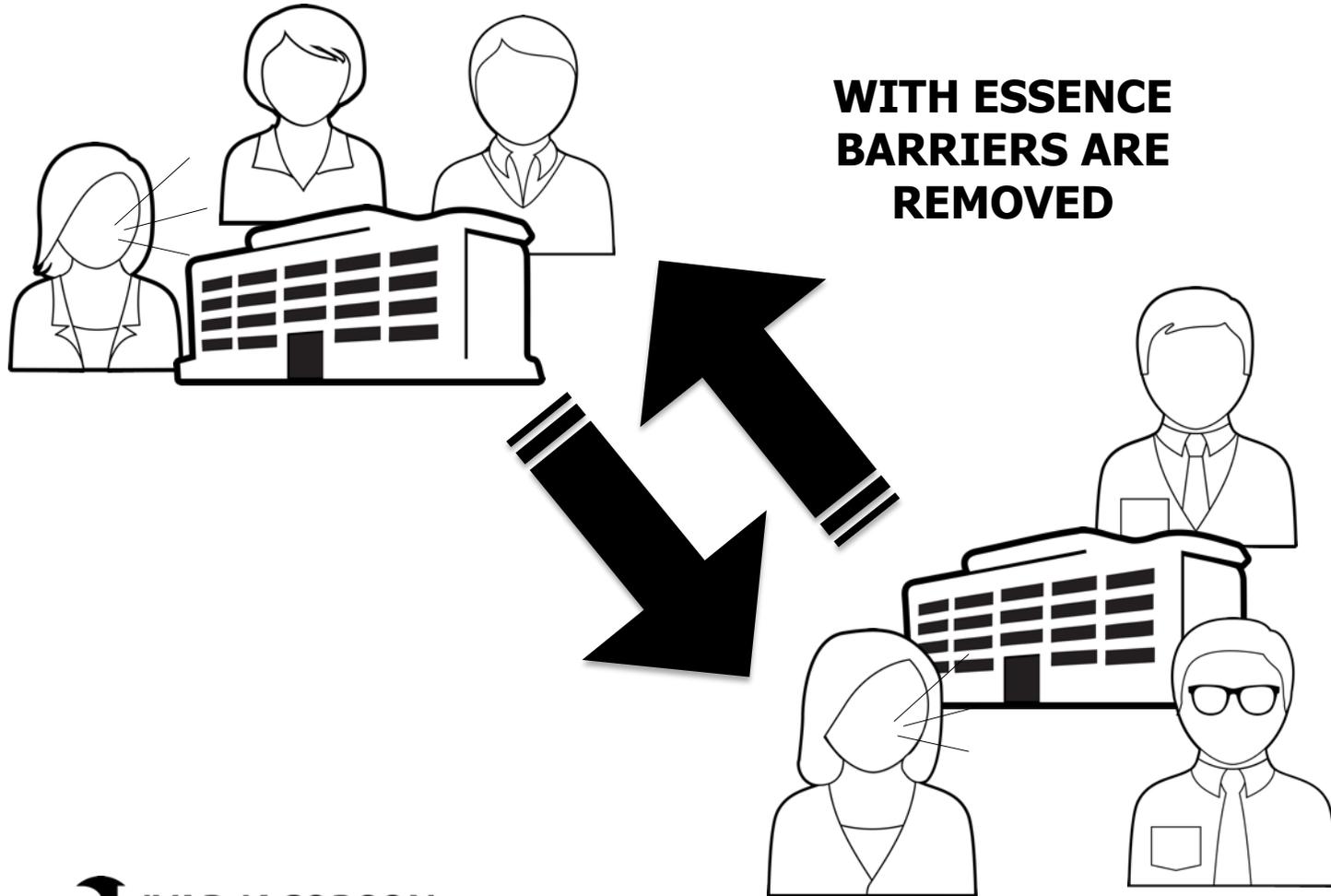
ALL 7 CMU-SV PRACTICUM TEAMS USED DIFFERENT SETS OF PRACTICES

ALL DEVELOPERS SPEAK THE SAME LANGUAGE INDEPENDENT ON WHICH METHOD THEY USE



- Moving from one organization to another is like converting culture

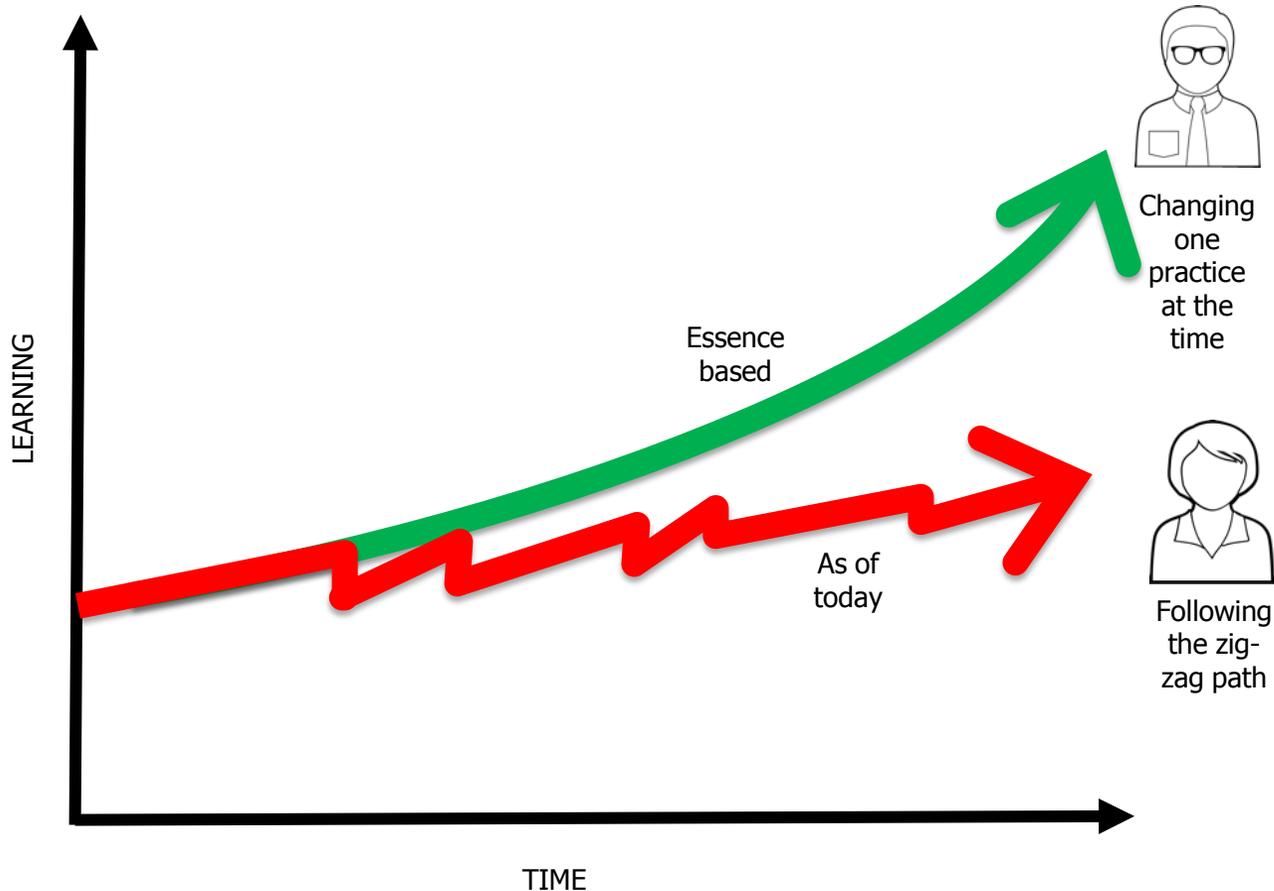
ALL DEVELOPERS SPEAK THE SAME LANGUAGE INDEPENDENT ON WHICH METHOD THEY USE



- **WITH ESSENCE:** Development can move very quickly
- Everyone is talking the same language!

YOUR ORGANIZATION IS FOREVER LEARNING

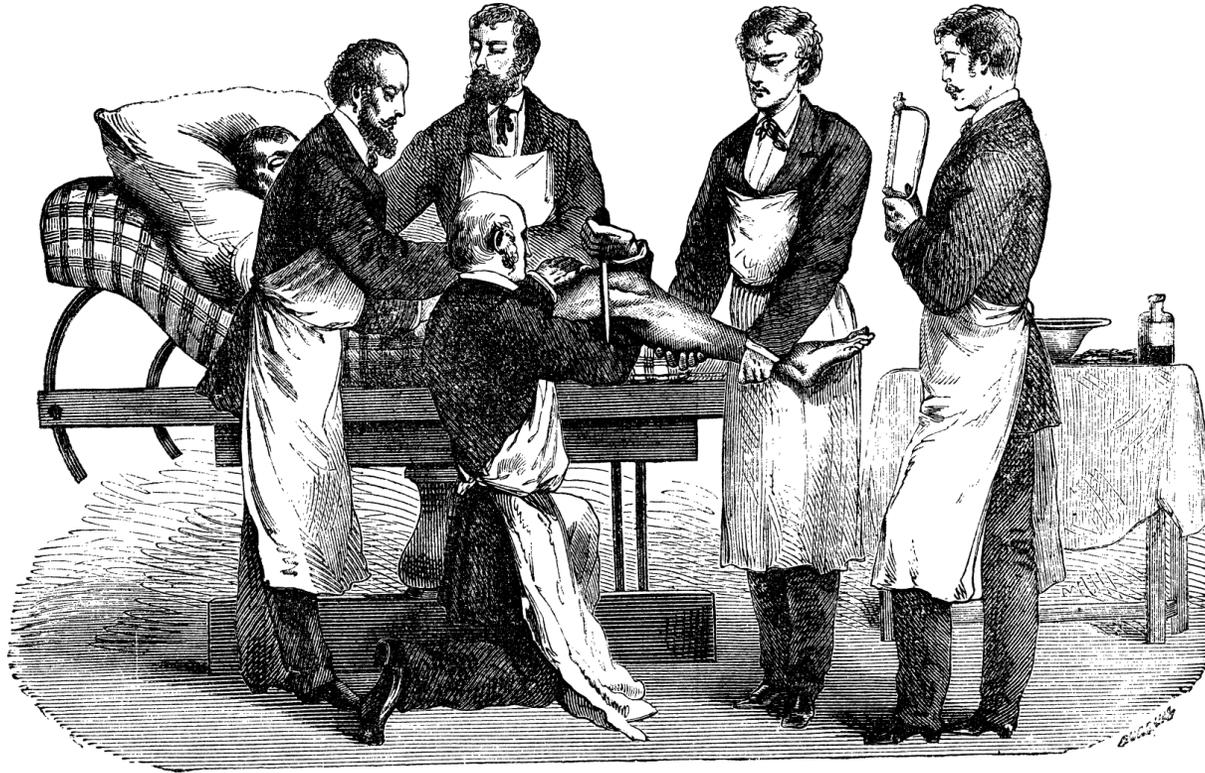
Learning for life instead learning one particular method, followed by the next



- Common language/common culture
- Increased competency of every individual
- More competent people will
 - develop better software faster and cheaper with happier customers.
 - be able to innovate more



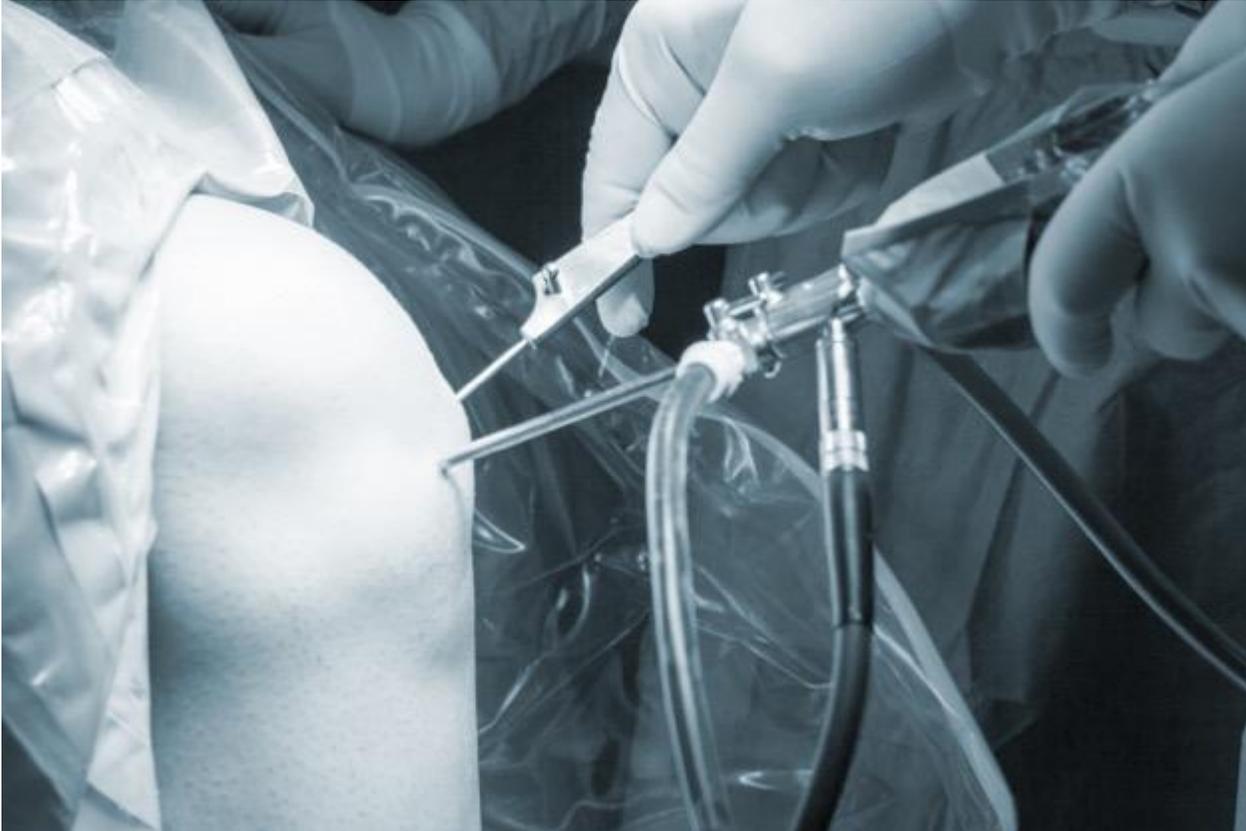
YOUR ORGANIZATION ADOPTS INDUSTRIAL SCALE ENGINEERING



TODAY SOFTWARE DEVELOPMENT IS MORE OF A CRAFT

- Software development will always need craftsmanship, craftsmanship that can stand on more or less science, more or less engineering, and more or less structured knowledge, but
- Today we are at a less mature level than where we will be standing after adopting Essence

YOUR ORGANIZATION ADOPTS INDUSTRIAL SCALE ENGINEERING



TOMORROW SOFTWARE DEVELOPMENT IS LESS OF A CRAFT AND MORE OF AN ENGINEERING DISCIPLINE

- Essence addresses systematically methods to allow for dramatic efficiency and quality improvements through tooling and techniques
- Application of many engineering practices
- Working systematically and less relying on heroics



AGENDA

1. How mature are we? 😊
2. What we need to do
3. What we get
4. What we do
5. Take aways

INDUSTRIAL WORLD

- *Fujitsu UK* and *Munich Re* have been using Essence for many years
- Several of the *largest and most prestigious service companies* are on a path to essentialization
 - *Tata Consulting Services (TCS)* are essentializing all their methods.
 - *RedHat* is using Essence for consulting engagements and are now scaling up to other areas.
 - *2e Consulting* in Korea has rewritten their methodologies in Essence.
- *A large telecom vendor in East Asia* has adopted Essence in their IT organization. It is now spreading nicely to other organizations within the same company.
- The largest project within *New Zealand government* uses Essence as a platform for their methods. Another *government organization in South Africa* is doing the same thing.
- Many other organizations are in the early phases on the road to adoption of Essence.
- Tools are already here

Finally companies get out of their method prisons and can build forever learning organizations

ACADEMIC WORLD

- Quote by Prof Pekka Abrahamsson, NUST: “...we have successfully taught Essence in Software Engineering course to 460 students... Essence empowered students to gain control of their project, work methods and practices. We have finally moved beyond Scrum and Kanban...my Software Engineering education in the future will be driven by Essence.”
- Universities around the world are already teaching Essence, eg CMU West, Florida Atlantic University, Copenhagen, Oslo, Stockholm, Vienna, Seoul, Beijing, Johannesburg, Medellin, Sao Paolo, Mexico City, St Petersburg, Wellington.
- A project ‘Software Engineering Essentialized’ for first year students started two years ago. About 25 professors from around the world.

Finally universities get something systematic, well-developed and futuristic to teach on software engineering

AGENDA

1. How mature are we? 😊
2. What we need to do
3. What we get
4. What we do
5. Take aways

SOMETHING TO ASK OURSELVES

Can we truly enable and empower our teams and become true learning organizations whilst we behave like the fashion industry?

Can we really see our ourselves as an open, diverse and collaborative community whilst we continually attack one another and rebrand, reinvent and rename everything?

Are we doomed to be locked in a never ending method war?



SOMETHING TO ASK OURSELVES

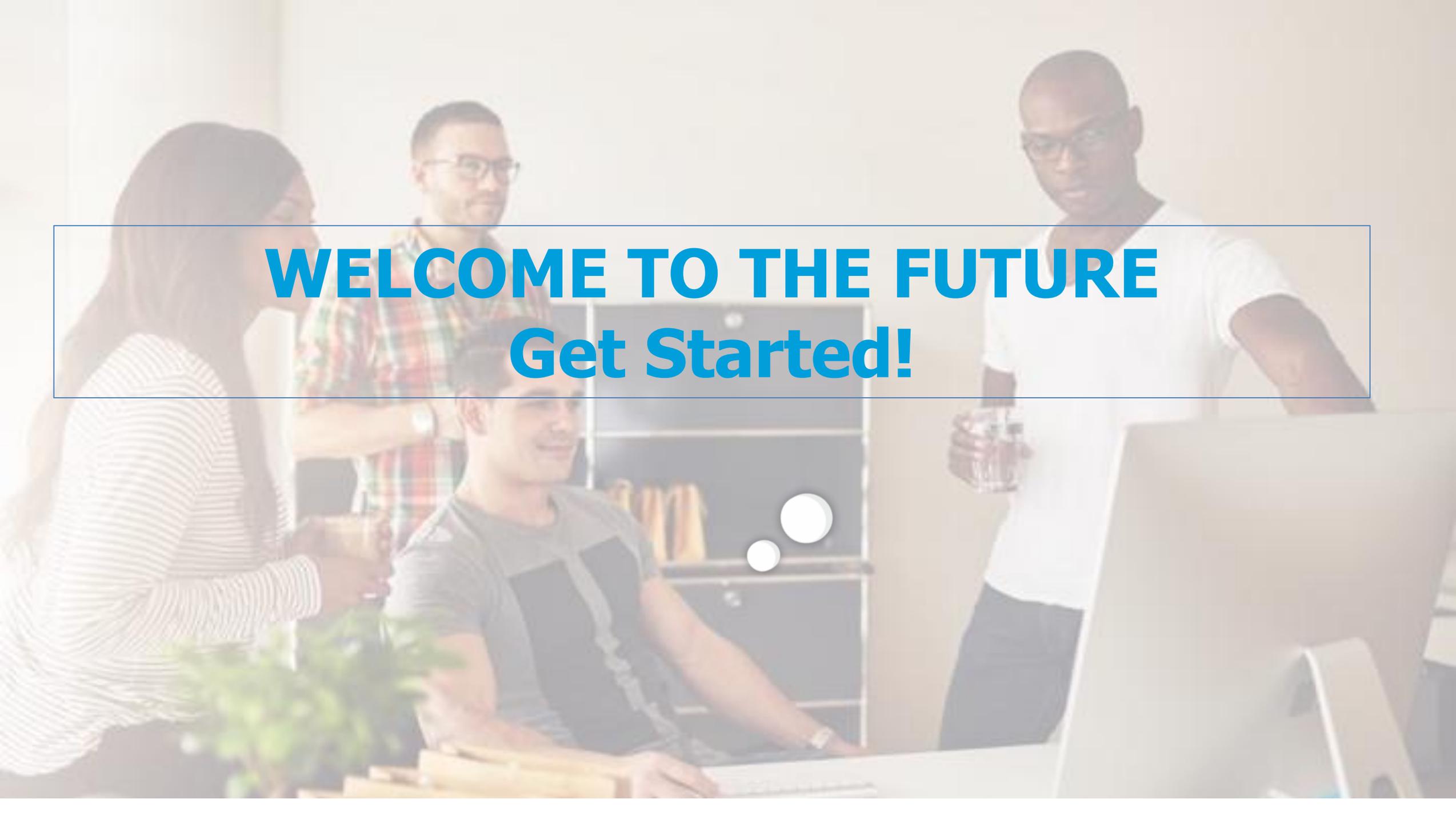
Can we truly enable
empower
be

Can we really see our own
as an open, di
collaborati

**Of Course – Noooo!
We can do better than that**

Are we doomed to be locked in
a never ending method war?





WELCOME TO THE FUTURE
Get Started!