IVAR JACOBSON
INTERNATIONAL

Creating winning teams.

# 50 years of Software Engineering, so now what?

Ivar Jacobson

# What is Software Development Really?

- Is it an Engineering Discipline or
- Is it a Craft?

The Answer is of course that it is both.

More of a Craft than an Engineering Discipline

# AGENDA

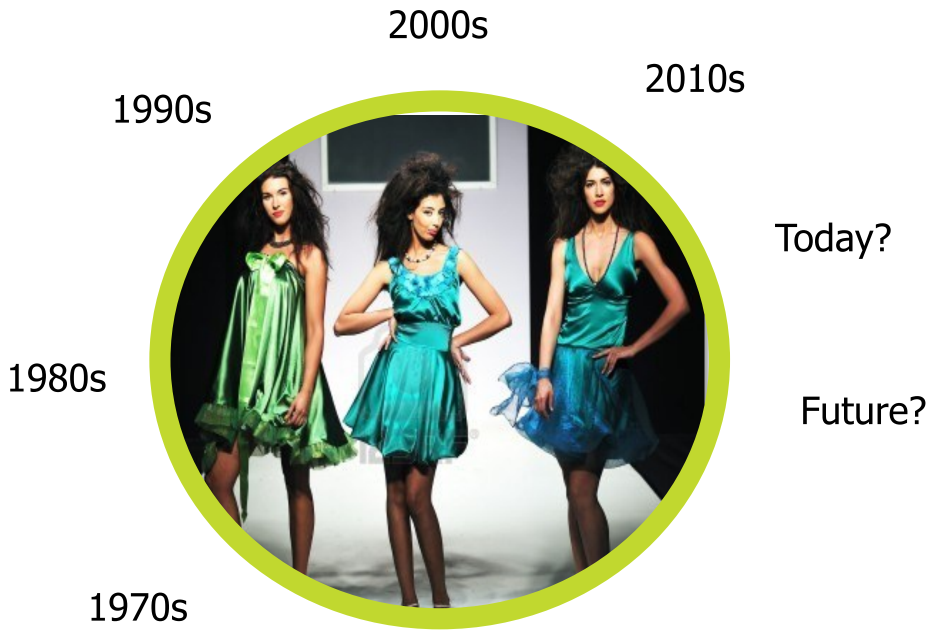
1. How mature are we? ☺
2. What we need to do
3. What we get
4. What we do
5. Take aways

# ARE WE DRIVEN BY FASHION?

- 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 (SAFe, DAD, LeSS, Nexus)
- Tomorrow ???

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

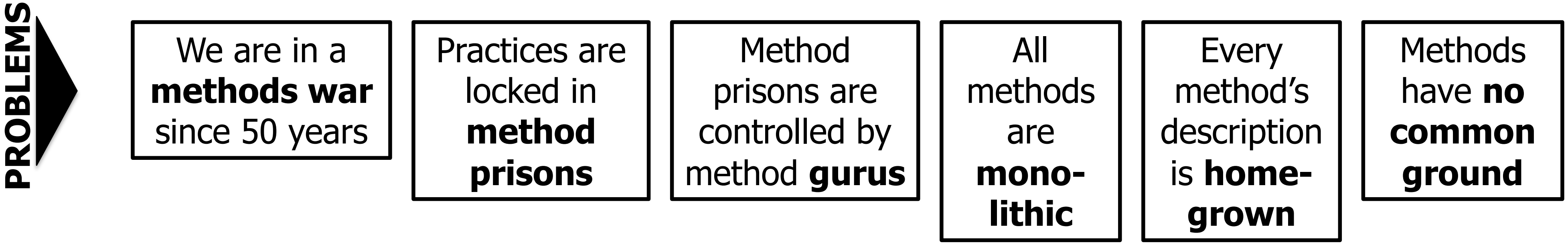
# ARE WE DRIVEN BY FASHION?



- 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 (SAFe, DAD, LeSS, Nexus)
- Tomorrow ???

## So What are the Problems?

# SO WHAT ARE THE **PROBLEMS** – HERE ARE SOME



**PROBLEMS**

We are in a **methods war** since 50 years

Practices are locked in **method prisons**

Method prisons are controlled by method **gurus**

All methods are **mono-lithic**

Every method's description is **home-grown**

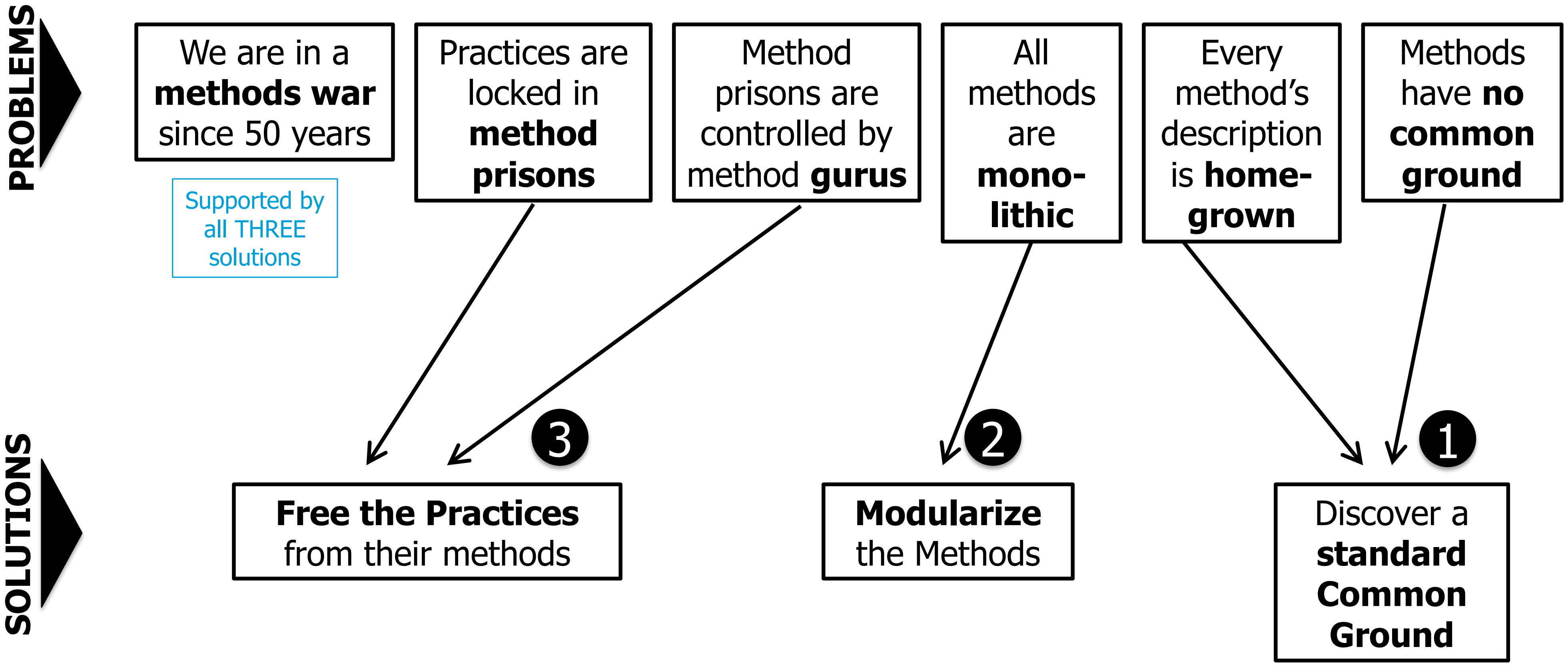
Methods have **no common ground**

## 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 FOOLISHNESS?



PROBLEMS

We are in a **methods war** since 50 years

Supported by all THREE solutions

Practices are locked in **method prisons**

Method prisons are controlled by method **gurus**

All methods are **mono-lithic**

Every method's description is **home-grown**

Methods have **no common ground**

SOLUTIONS

3 **Free the Practices** from their methods

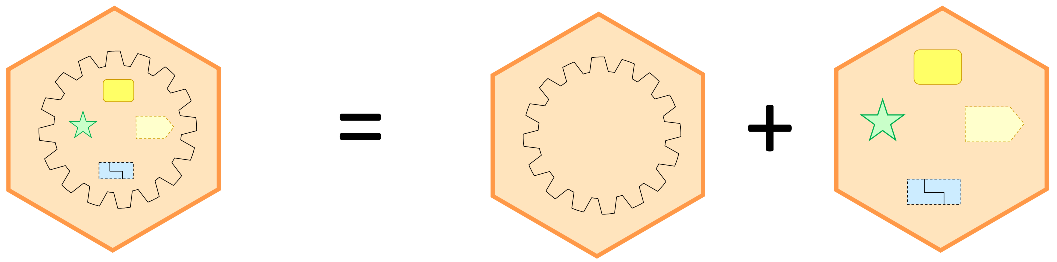
2 **Modularize** the Methods

1 Discover a **standard Common Ground**

# 1 DISCOVER A STANDARD COMMON GROUND

For the first time in the 60+ year history of software engineering, we have got a Common Ground (in 2014)

**Essence** **Kernel** **Language**



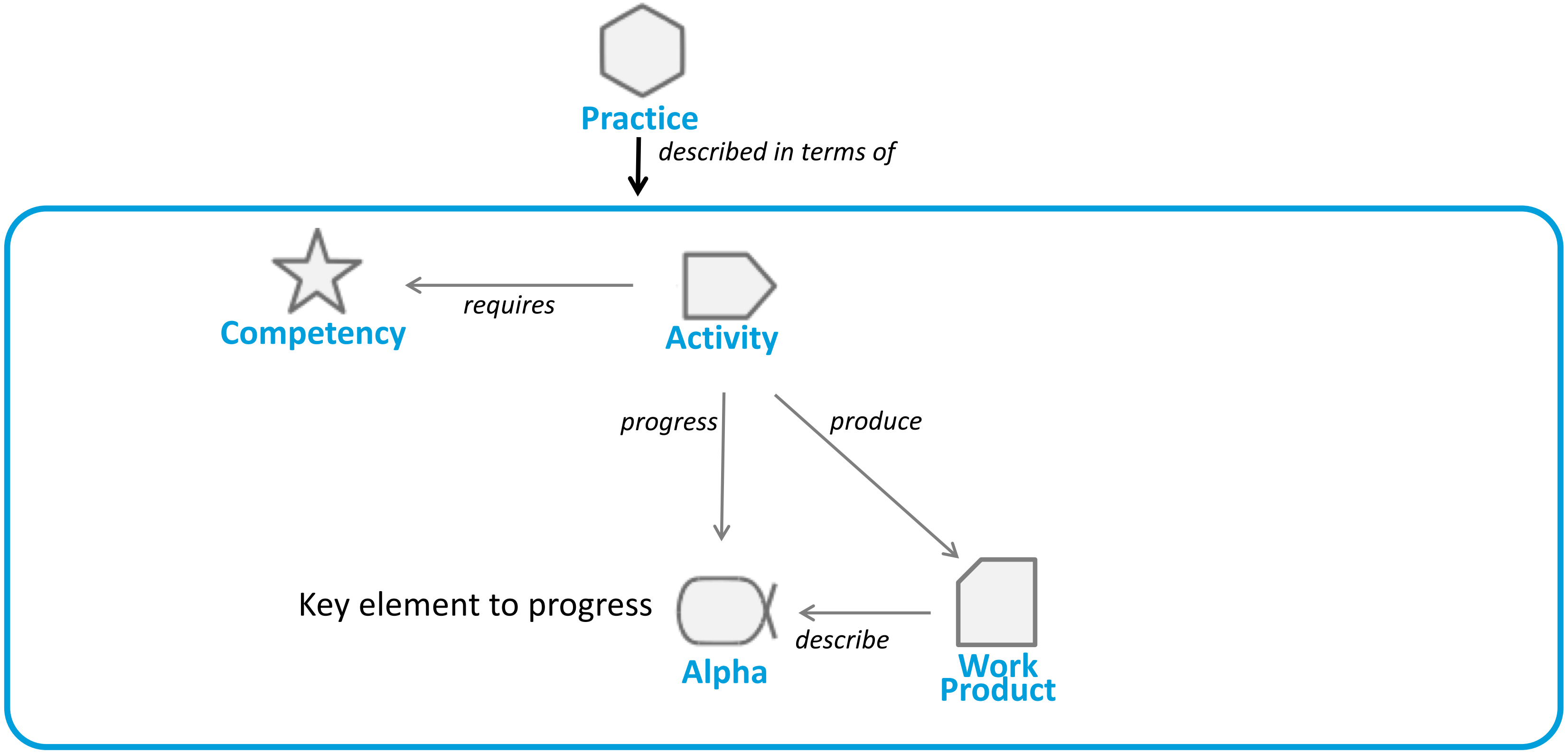
THE COMMON GROUND

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

- Visual language
- Simple
- Intuitive

WE SET THE STANDARD

# THE ESSENCE LANGUAGE



Key element to progress

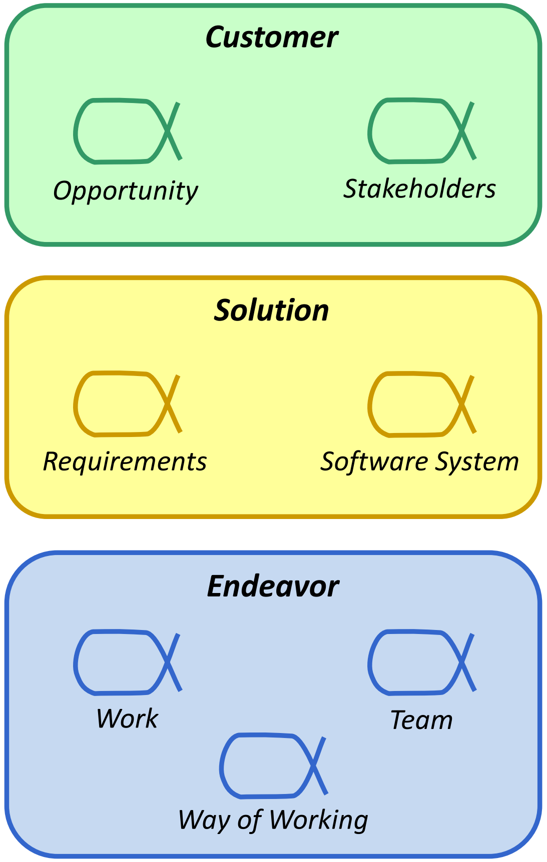
# THE ESSENCE KERNEL

## The Alphas: The Most Important Things

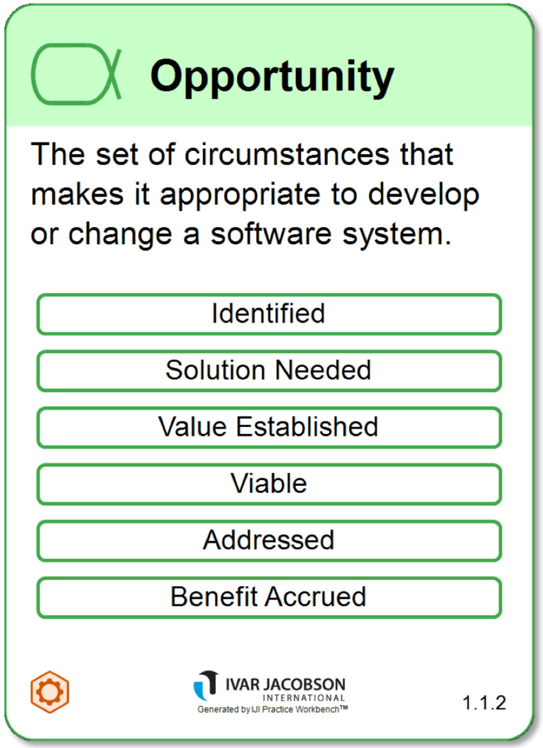
- There are **customer** needs to be met
  - Someone has a problem or ***Opportunity*** to address
  - There are other ***Stakeholders*** who will fund, use and benefit from the solution produced

- 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***



# ALPHAS HAVE STATES AND CARDS



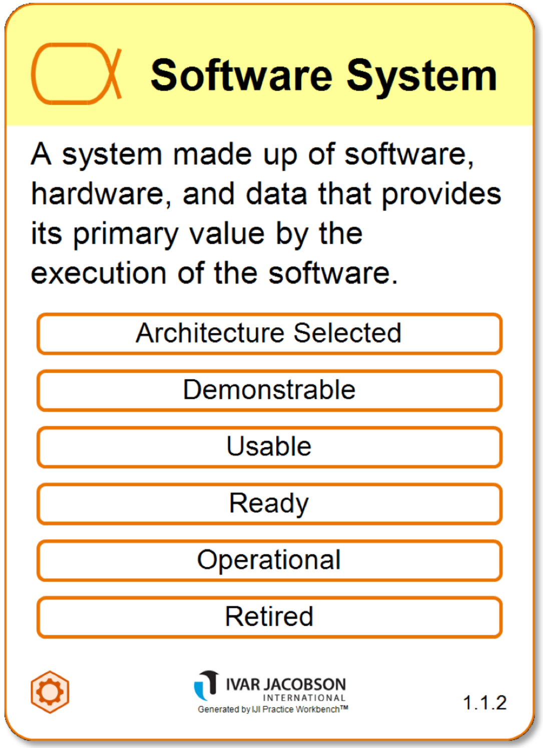
## Opportunity

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 IJI 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

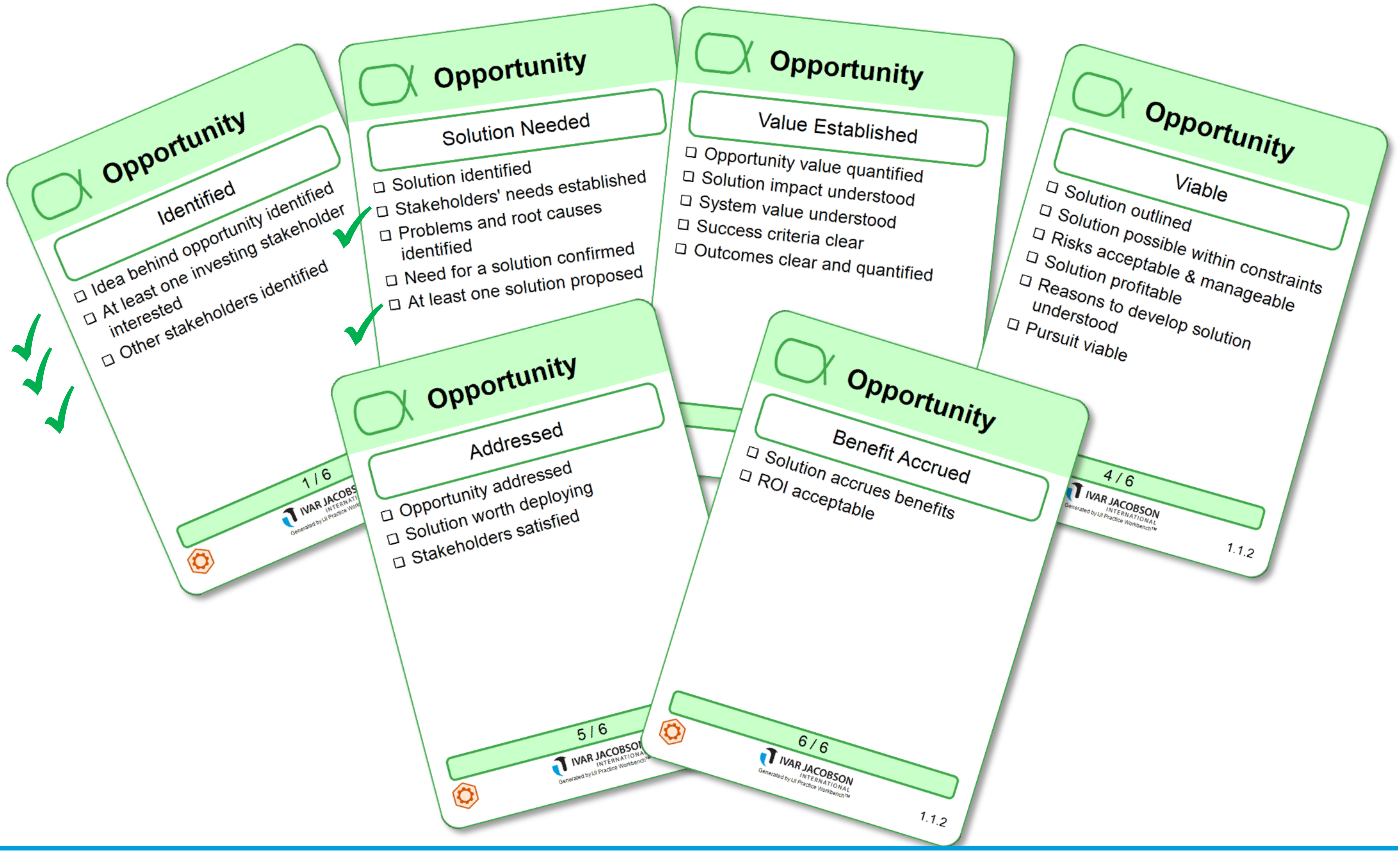
IVAR JACOBSON INTERNATIONAL
Generated by IJI Practice Workbench™

1.1.2

The cards allow us to play many serious games

*Quote from: "The Lean Machine: How Harley Davidson Drove Top-Line Growth and Profitability with Revolutionary Lean Product Development" by Dantar P. Oosterwal*

# ALPHA STATES HAVE CHECKLISTS



**Opportunity**

Identified

- □ Idea behind opportunity identified ✓
- □ At least one investing stakeholder interested ✓
- □ Other stakeholders identified ✓

1 / 6

**Opportunity**

Solution Needed

- □ Solution identified
- □ Stakeholders' needs established
- □ Problems and root causes identified ✓
- □ Need for a solution confirmed
- □ At least one solution proposed ✓

**Opportunity**

Value Established

- □ Opportunity value quantified
- □ Solution impact understood
- □ System value understood
- □ Success criteria clear
- □ Outcomes clear and quantified

**Opportunity**

Viable

- □ Solution outlined
- □ Solution possible within constraints
- □ Risks acceptable & manageable
- □ Solution profitable
- □ Reasons to develop solution understood
- □ Pursuit viable

4 / 6

1.1.2

**Opportunity**

Addressed

- □ Opportunity addressed
- □ Solution worth deploying
- □ Stakeholders satisfied

5 / 6

**Opportunity**

Benefit Accrued

- □ Solution accrues benefits
- □ ROI acceptable

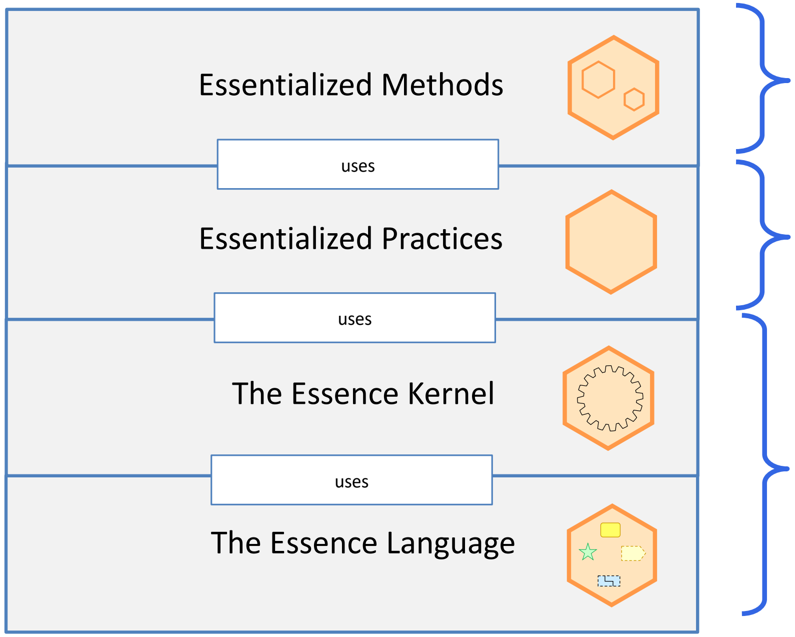
6 / 6

1.1.2

IVAR JACOBSON INTERNATIONAL
Generated by IJI Practice Workbench™

The checks measure real outcome, not activities done or documents produced

# 2 MODULARIZE THE METHODS

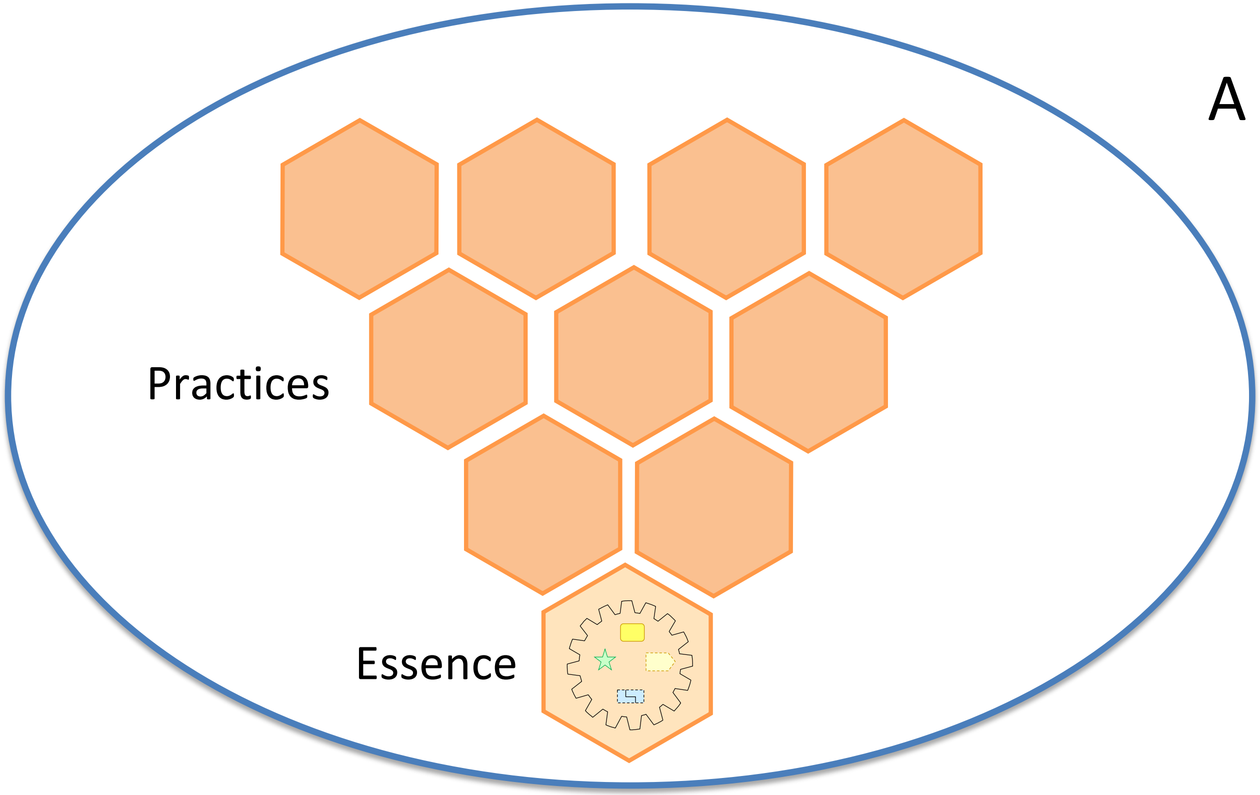


Methods like SAFe, DAD, SPS, LeSS, etc.

**Generic practices**: Scrum, User Story, Use Cases, etc.
**Specific practices**: Practices from SAFe, DAD, Nexus, LeSS, etc. – All in a practice eco-system

**The Essence Common Ground**

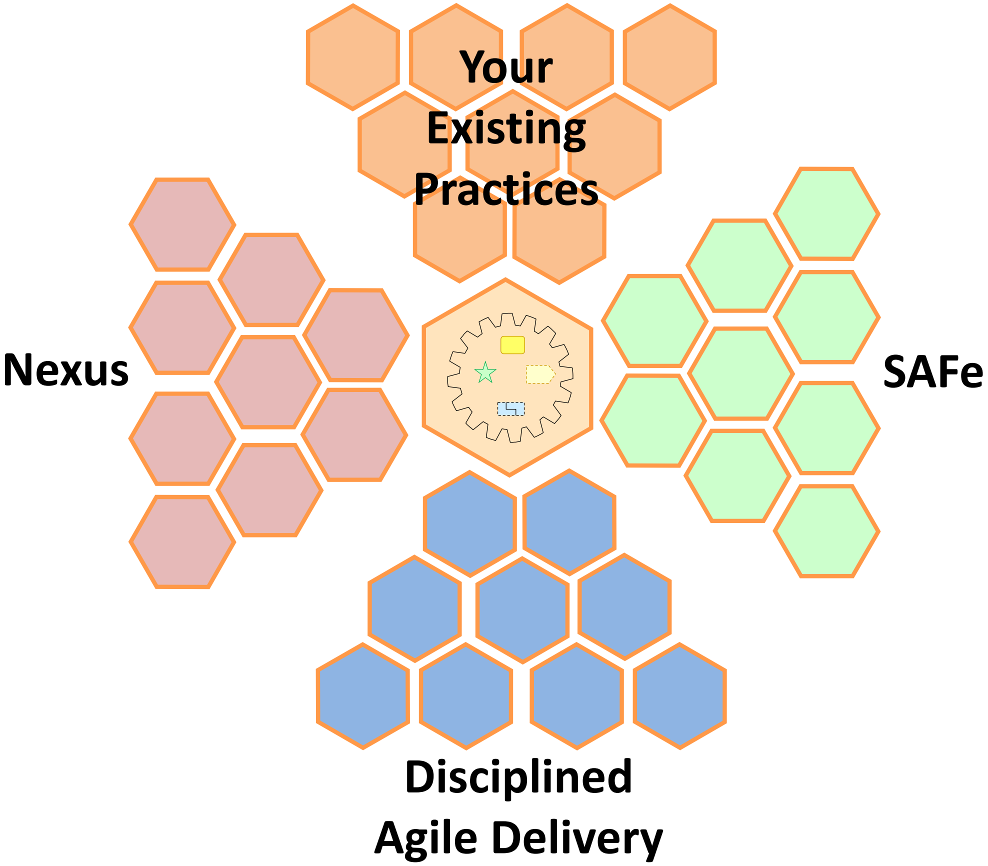
# THUS THIS IS AN ESSENTIALIZED METHOD



Essence makes Methods Modular – not Monolithic

# 3 FREE THE PRACTICES FROM THEIR METHODS

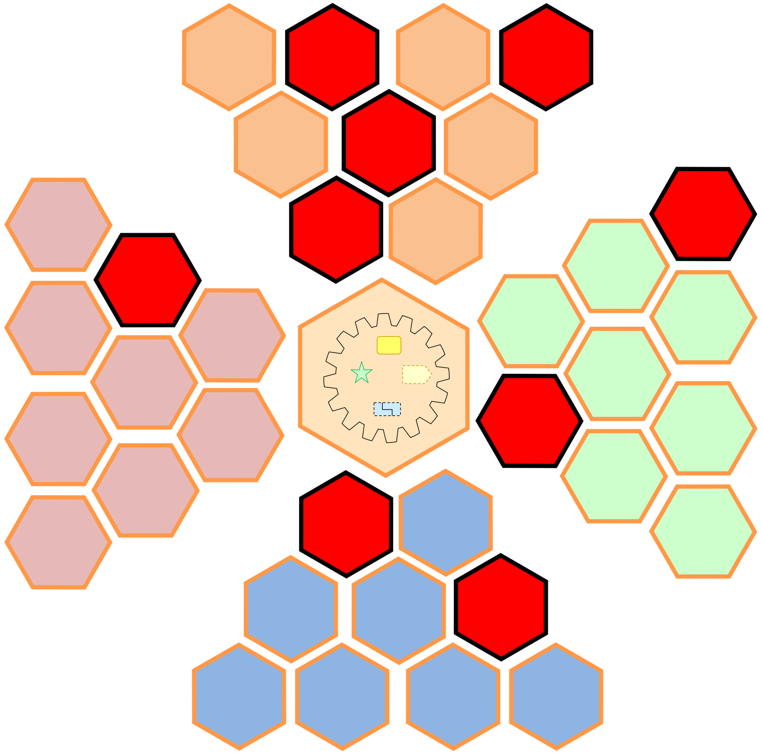
Imagine a Practice Ecosystem



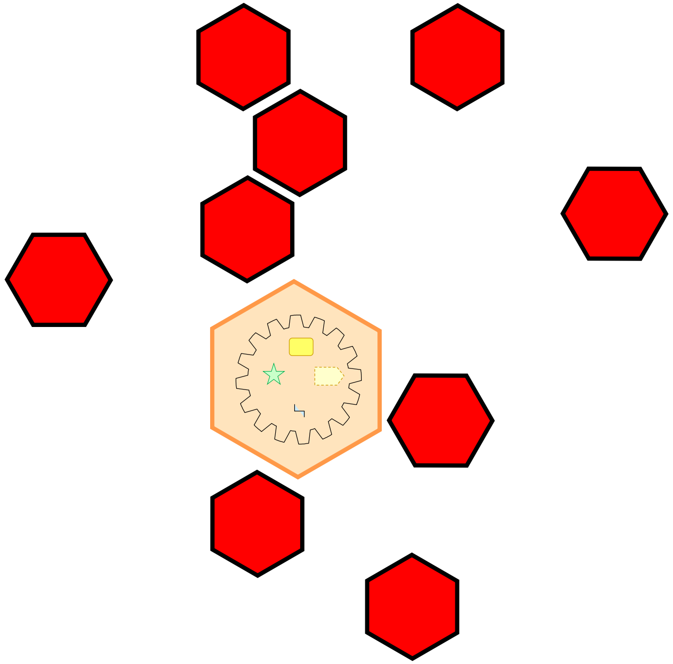
IVAR JACOBSON INTERNATIONAL

Creating winning teams.

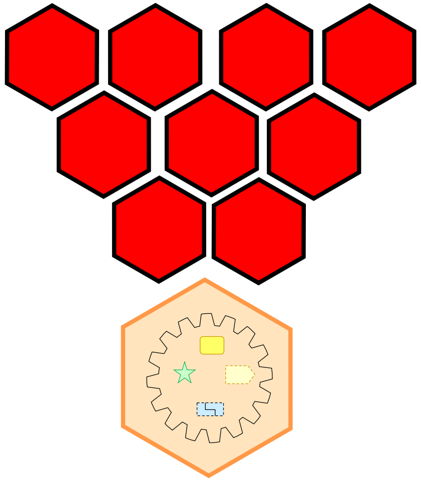
# YOU SELECT THE PRACTICES YOU LIKE



# 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. Then what more?

# Industry - Value Proposition

1. Every team can select their own preferred method
2. Progress and health of projects are measurable in a method agnostic way
3. All developers speak the same language independent on which methods they use
4. Your organization is forever learning
5. Your organization moves from primarily being a craft to primarily being an engineering discipline

# Academic- Value Proposition

1. Students learn what is generic to all methods, not just specific methods
2. Students can learn specific methods based on a common ground so they can compare methods on a detailed level
3. Students learn game playing in software development
4. Researchers have got a standard model of software engineering to build upon

# AGENDA

1. How mature are we? ☺
2. What we need to do
3. What we get
4. What we do
5. Then what more?

# 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) say they have essentialized all their methods.
  - RedHat is using Essence for consulting engagements and are now scaling up to other areas.
  - Etc.
- Jeff Sutherland has essentialized Scrum
- Scott Ambler is essentializing DAD

# ACADEMIC WORLD

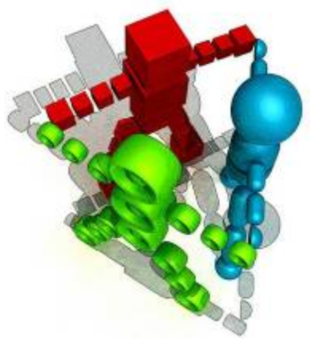
- Quote by Prof Pekka Abrahamsson, NUST: "…we have successfully taught Essence in Software Engineering course to 460 students… We have finally moved beyond Scrum and Kanban…my Software Engineering education in the future will be driven by Essence."
- More than 10 universities around the world are already teaching Essence.
- A project 'Software Engineering Essentialized' for first year students started two years ago.  About 25 professors from around the world are actively involved to create a new way to teach software engineering

# AGENDA

1. How mature are we? ☺
2. What we need to do
3. What we get
4. What we do
5. Then what more?

# Codified Practices are the Beginnings of Smart Practices

Virtual Pair
- Programmers
- Analysts
- Designer
- Tester
- Project Managers



## Active Guidance

provides expert advice just when neec

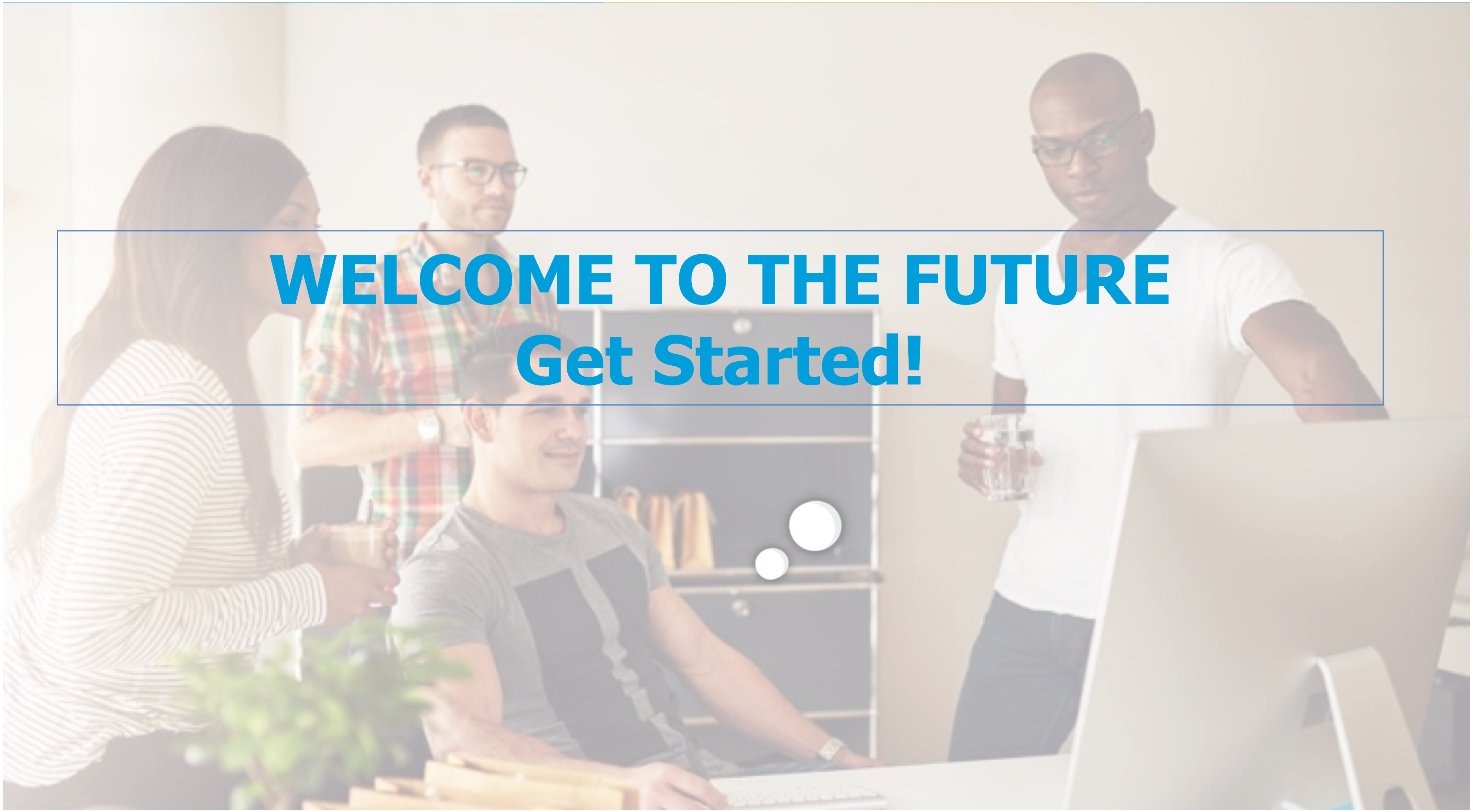
## Active Review

monitors status, progress and quality (individual/mgmt)

## Active Automation

delegates mundane tasks

Finally we are AI ready



# WELCOME TO THE FUTURE

## Get Started!