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

- 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 monolithic
- 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
Creating winning teams.

**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 monolithic
- Every method’s description is home-grown
- Methods have no common ground

**SOLUTIONS**

1. Discover a standard Common Ground
2. Modularize the Methods
3. Free the Practices from their methods

Supported by all THREE solutions

ivarJacobsen International

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

Essence = Kernel + Language

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

- Visual language
- Simple
- Intuitive
THE ESSENCE LANGUAGE

Practice

described in terms of

Competency

requires

Activity

progress

produce

Key element to progress

Alpha
describe

Work Product
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
Creating winning teams.

ALPHAS HAVE STATES AND CARDS

The cards allow us to play many serious games

ALPHA STATES HAVE CHECKLISTS

The checks measure real outcome, not activities done or documents produced.
### MODULARIZE THE METHODS

<table>
<thead>
<tr>
<th>Essentialized Methods</th>
<th>Uses</th>
<th>Methods like SAFe, DAD, SPS, LeSS, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Essentialized Practices</td>
<td>Uses</td>
<td><strong>Generic practices:</strong> Scrum, User Story, Use Cases, etc.</td>
</tr>
<tr>
<td>The Essence Kernel</td>
<td>Uses</td>
<td><strong>Specific practices:</strong> Practices from SAFe, DAD, Nexus, LeSS, etc. – All in a practice eco-system</td>
</tr>
<tr>
<td>The Essence Language</td>
<td>Uses</td>
<td><strong>The Essence Common Ground</strong></td>
</tr>
</tbody>
</table>

*Creating winning teams.*
THUS THIS IS AN ESSENTIALIZED METHOD

A Method

Essence makes Methods Modular – not Monolithic
FREE THE PRACTICES FROM THEIR METHODS

Imagine a Practice Ecosystem

Your Existing Practices

Nexus

SAFe

Disciplined Agile Delivery
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

Active Guidance
provides expert advice just when needed

Active Review
monitors status, progress and quality (individual/mgmt)

Active Automation
delegates mundane tasks

Finally we are AI ready

I do use cases
I do architecture
I do Iterative planning
I do Embedded Systems
I do Business Systems

Virtual Pair
- Programmers
- Analysts
- Designer
- Tester
- Project Managers
WELCOME TO THE FUTURE
Get Started!