

# Devoxx 2009 – Day 2

## Key Takeaways from Uncle Bob and Ivar Jacobson

### Introduction

This keynote is going to be delivered by two men that I do respect very much. I've used (and put to practice) a hell of a lot of material from both over the years, be it the design principles from Uncle Bob (Robert C. Martin) or the incredibly profound work from Ivar Jacobson on the process (and now practices) front.

### Background

I've used Uncle Bob's principles since about 15 years. In '92 I was cutting heavy multithreaded C++ for IVR platforms that I designed from the ground up and his ideas were key to my success with these. All of his macro and micro design principles still hold water.

Ivar's current line of thought around Practices (with a big P, since they can be Composed) and Aspect Oriented Development with Use Cases really make him a man with a straight mind. His approach makes it easy to tackle a lot of situations with a very « lean » mindset. Because by focusing on the « alphas » evolutions, all the rest becomes easy to figure out and help in dealing with « fashionistas » by harnessing them in the right scope and having them deliver.

I am also quite « brainwashed » (in a good way, mind you) with UP (for Unified Process), not to be confused with RUP (Rational Unified Process), which is really too large for a mere mortal (read process practitioner) do digest (okay, this makes me not that normal, and I do agree with that – as in « People do not read process books », which I do read – and with pleasure at that. But I am just finding too many nuggets of gold in these books to stop. I had a chat with Scott Ambler on books and he may be even worse than me, he moved about 6 tons (yes, tons) of books around when moving.).

“So what?” will you say. Well, I am coaching people on how to envision, shape, build, and deliver solutions using technologies. Given the state I see in the field, these ideas have quite a future. Lots of projects are shooting themselves in the foot (even worse, I do think that a lot of times, they do hammer one foot in the ground and run in circles around it, complaining that it hurts!). Psychology would have a lot to say on that, especially on the denial aspect.

The light is turning down, we are about to start.

# Keynote 1: Ivar on « Do we really know how to develop software? »

Ivar mentions that it is the first time that he gives this new talk.

State of the industry:

- Every developer knows how to develop software but as an industry, we don't.
- We have no widely accepted foundation

Are we seen as a trustworthy profession? There are doubts about us, especially the business side...

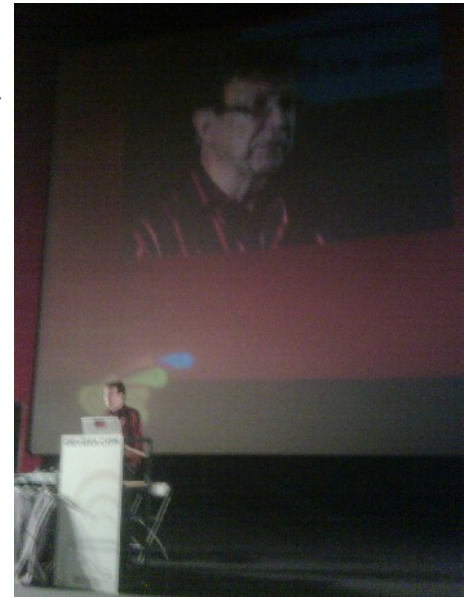
The press.... not always kind:



Business Week's: SOFTWARE HELL – any way out of this mess? A lot of people consider that we do not know what we are doing.

Are we seen as a fashion industry? You bet! OO, UML, Components, CMMI, XP, SCRUM.

UML is now back, because there is really no other solid modeling notation.



About CMMI: you can be a level 5 garbage developer.  
What don't you get out of CMMI: good software! (wow, that's a good line).

Scrum: there are good things in Scrum. But a two day class @ 2000 \$ per person, delivering you a certified scrum master without testing... this is ridiculous.

## ***Industry vs Academics***



## ***Industry vs Academics vs Methodologists***

This is getting more complex indeed :-)



The same story repeats: each company builds its own methodology and borrows from here and there. We end up with soups. We do reinvent the wheel over and over again.

Things look like new, but are not. What was really new in UP for example?

Whatever we have today, we have the feeling that tomorrow, it will be out of fashion.

**“Are you in my sect or outside my sect????”**

We can learn using books. But, they are not consistent with one another. How can we know who is « right ». We have a MESS here.

We need a theory! (some kind of BASE, just as tech is powered by science, software engineering must be founded by a sound BASIS).

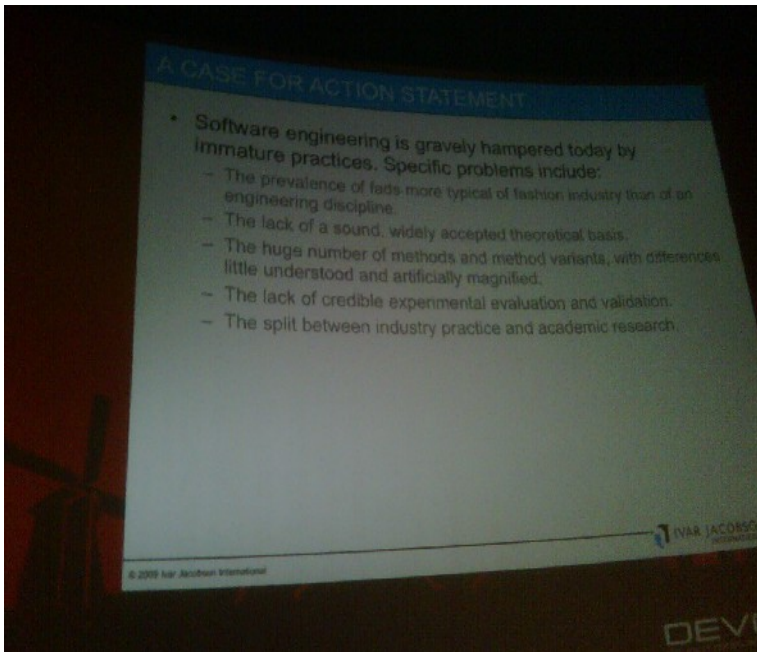
## ***A call to arms***

Many of us have felt that this is just CRAZY.

It is gone too far to continue like this.

## ***So: A CALL FOR ACTION STATEMENT***

Software engineering is gravely hampered today by immature practices.



So, unsmart!

We support a process to refund software engineering based on a solid theory, proven principles and best practices that:

- are based on a kernel
- addresses tech and people
- are supported by industry, academia, researchers, and users
- support extension in the face of changing requirements and technologies

This is smart.

We cannot continue to have this distance between industry, academia, researchers, and users.

This is the SEMAT initiative. Remember that name.

So: 25 top signatories.

Scott Ambler, Alistair Cockburn, Ellen G., Ivar Jacobson, Barry Boehm. Ken Schwaber..

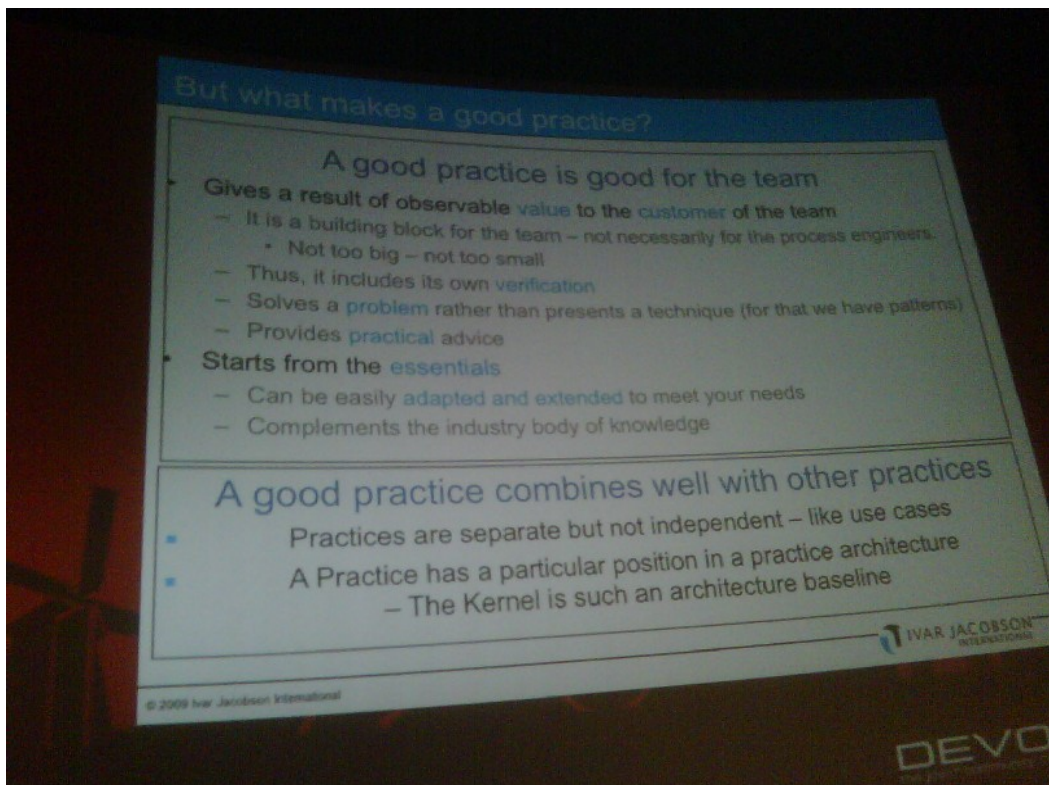
So, we want to change the world. We may fail. This is about building a community.

## ***Ivar Jacobson Internation work***

So, let's move to the IJI work. This is not going to be the SEMAT result but this is showing how we can move towards a solution. Show something credible.

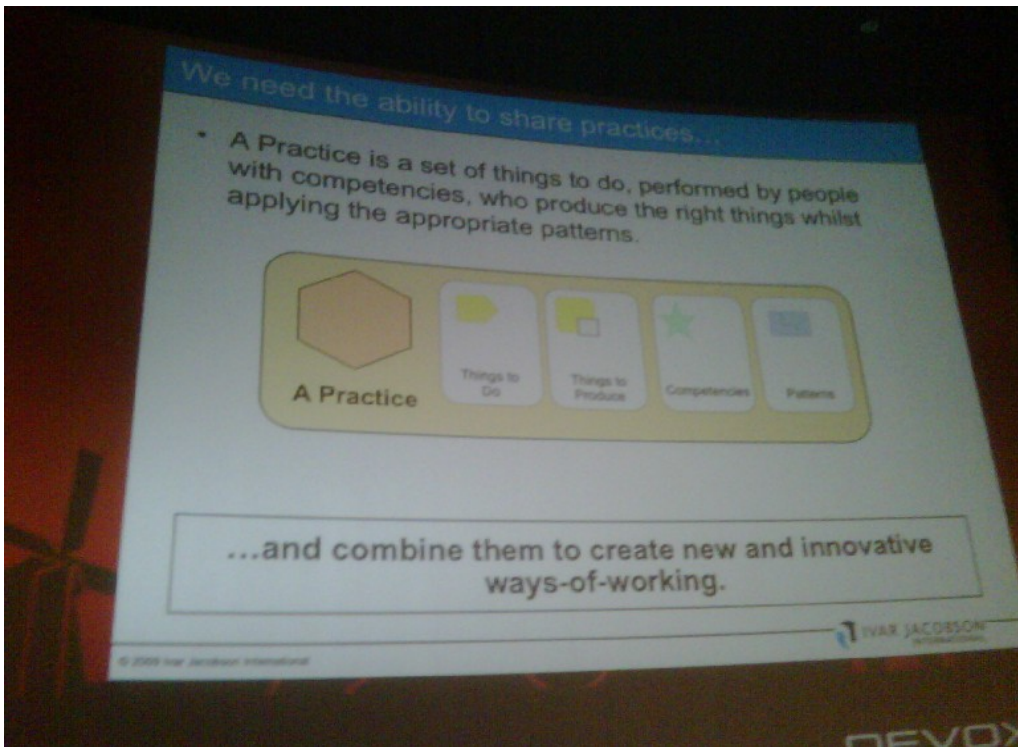
As an industry, we have stopped talking about methodologies. Enough process, let's talk about practices. All big companies are now pushing for practices. You do not hear about RUP anymore, but about practices. Same at Microsoft. A method becomes a packaging of practices.

A practice is an aspect in the space of methodology. So that you can discuss practices, and compose practices. They can be merged, and overlaid. Practices are not components, they are **aspects**.



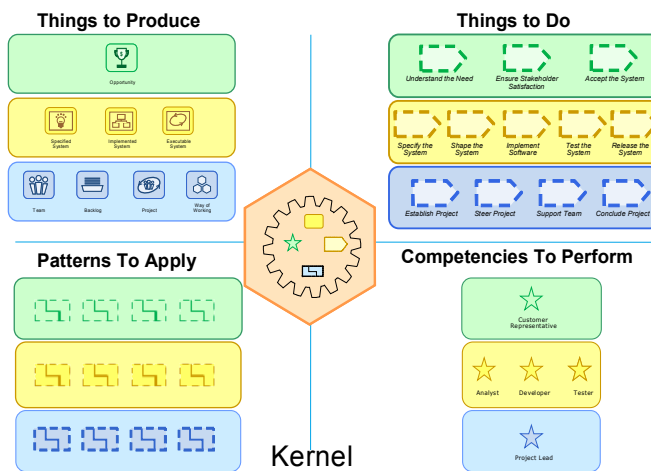
The idea is to get practices from everywhere, and compose them through merge ops.





Underneath the practices, there is a kernel.

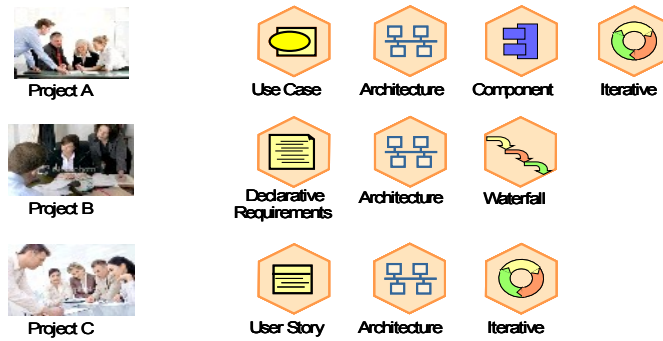
Kernel -> Practice -> Way of Working



Kernel: learnable in 2 hours. We use a day, because of the examples.

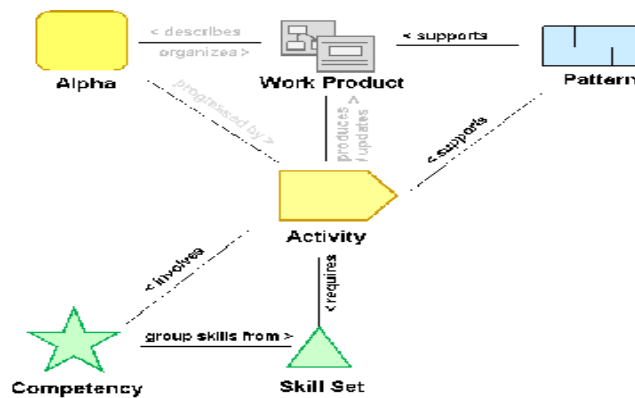
The KERNEL ensures common understanding across teams in a minimal way.

The kernel allows practices to be shared.



All centered around the kernel

Ivar shows a sample of how to slot in a particular requirements practice in an activity space.



Meta-model overview.

Ivar: I've seen so many companies that threw all away when Scrum came. The managers where not able to resist the popularity of a new fad. This is absolutely senseless. This costs a lot of money.

Instead, you can use this approach to improve your way of working stepwise.

You add a practice to what people are doing.

## Academics

The THEY, THEY, THEY communication : they do not know, they do not provide something useful...

So much energy is spend on doing PhD theses that do not add real value.

People buy the books... but they do not read them. That's why I travel the world, telling people about what is in the books. Why do you still write books? Hey, people buy them!

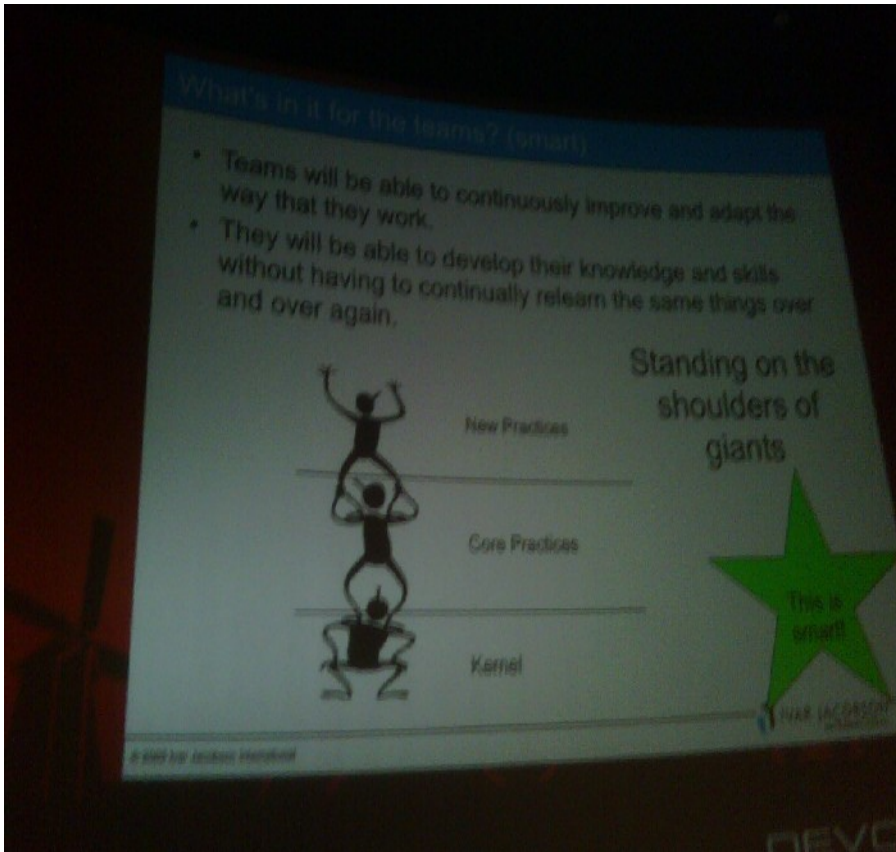
We want the world to create new practices, but not rewrite EVERYTHING.

## Wrap up

I start to get to the end now...

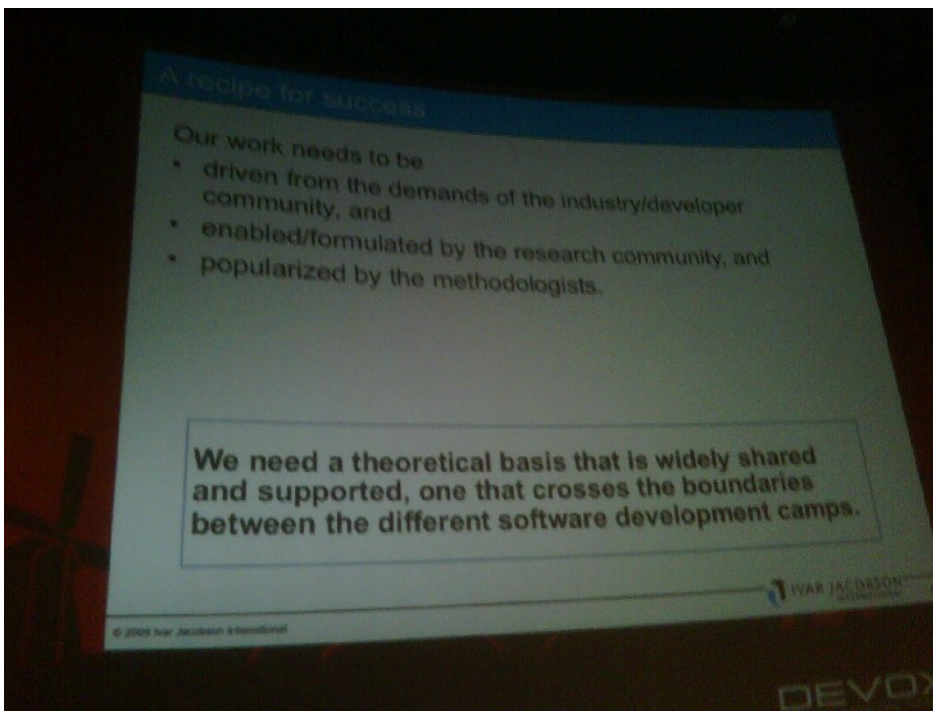
People tell.. « I already have a solution » but it is in their minds.

So, demand driven of the industry/developer community! Enabled/formulated by the research community, and popularized by methodologist!



What we try to do is not rocket science and will require a lot of work to build consensus.





[a\\_theory@ivarjacobson.com](mailto:a_theory@ivarjacobson.com)

Meyer was selected because he has a solid research background  
Another has very solid standards experience

Whatever theory we come up with, it has to be extensible.

[www.semat.org](http://www.semat.org)

Sign up to support us.

« There is nothing as practical as a good theory » --Kurt Lewin

A good theory helps you doing things.  
Unfortunately, a lot of theories are not very good.

So, a sound foundation to make us move forward as an industry.

So, a very good talk that shows us the future.

--Philippe Back  
November 2009

web: <http://philippeback.eu> | blog: <http://philippeback.be>