

What we should teach you about software at school

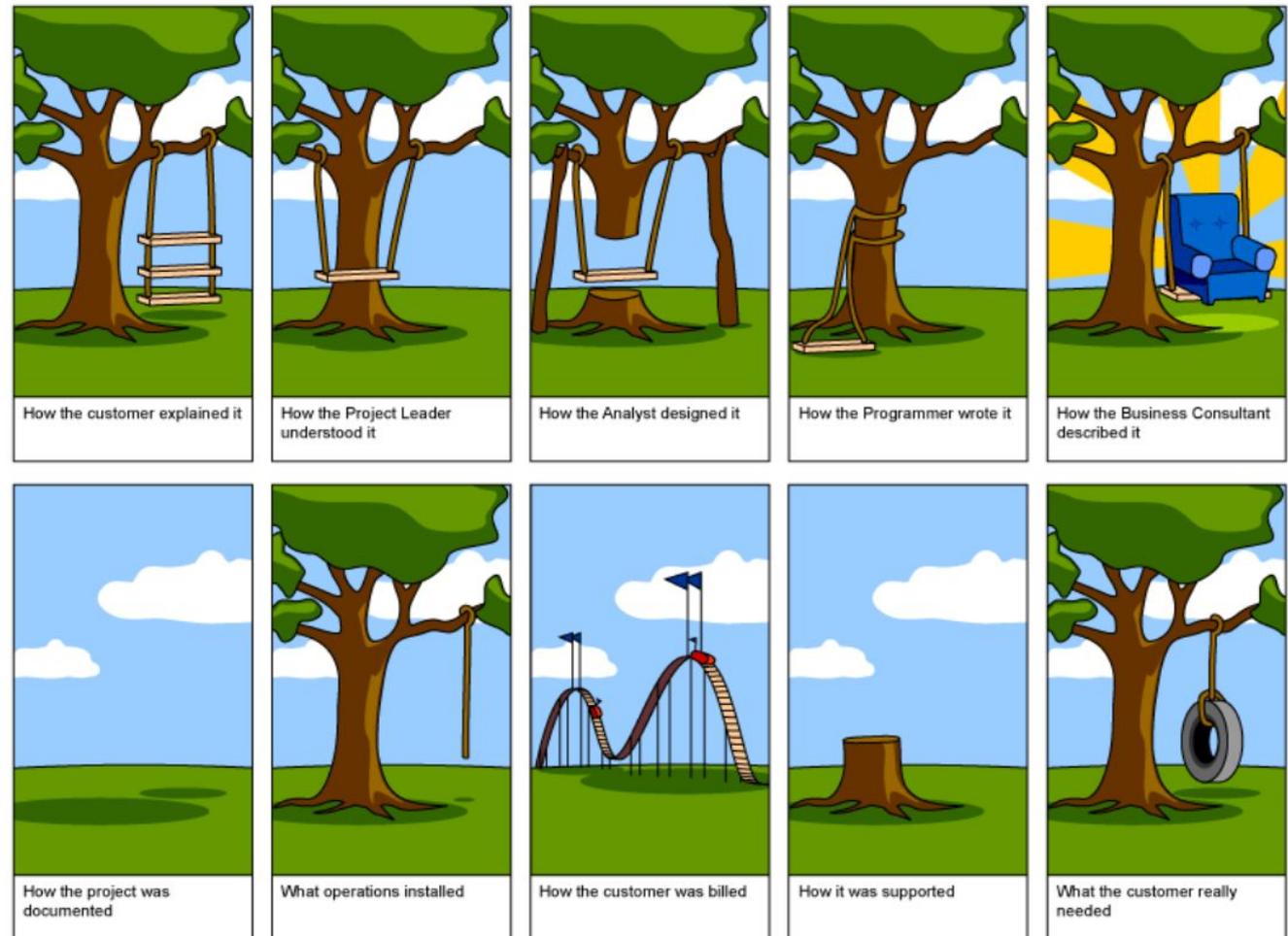
The Essence

Ivar Jacobson
ivar@ivarjacobson.com

Agenda

- The State of the Industry
- Specific Problems
- A Case for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

Are we seen as a trustworthy industry?



- How a Software Company would design a Treeswing

Are we seen as a trustworthy industry?

Blackberry users complain of

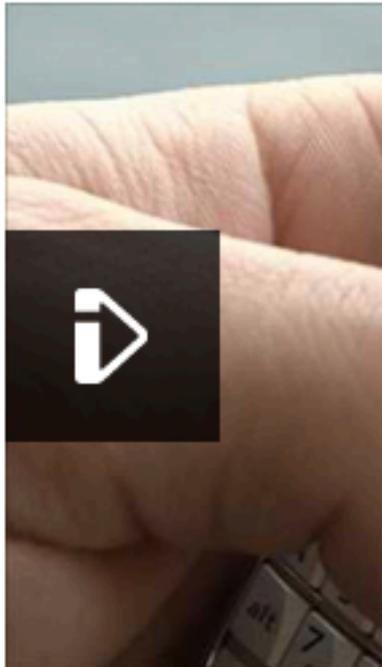
COMMENTS (552)

Sydney cash machines give out extra money

Queues formed at cash machines in the Australian city of Sydney after they malfunctioned and began giving out sums of money.

More than 40 Commonwealth Bank cash machines across the city were affected by the problem.

Officials said they were operating in stand-by mode, so could not identify the account balances of customers.



BBC Radio 1 Newsbeat

Blackberry users have complained about the company which makes them "operating normally".



The bank said the glitch was caused by software problems with cash machines going offline

Source: Top 10 Software Failures Of 2011
By Phil Codd



Are we seen as a trustworthy industry?

Blackberry users complain

COMMENTS (552)

Sydney

Software engineering is gravely hampered today by immature practices.

BBC Radio 1

Blackberry users have a company which makes "operating normally".

Officials said they were operating in stand-by mode, so could not identify the account balances of customers.

The bank said the glitch was caused by software cash machines going offline

Source: Top 10 Software Failures Of 2011
By Phil Codd

Agenda

- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

The prevalence of fads typical of fashion industry

Software Development seems to be driven by fashions and fads

- Twenty years ago it was all about OO
- Twelve years ago it was about components, UML, Unified Process (RUP)
- Ten years ago it was about CMMI and Spice
- Four years ago it was about XP
- Yesterday it was about Scrum
- Now it is about Lean and Kanban

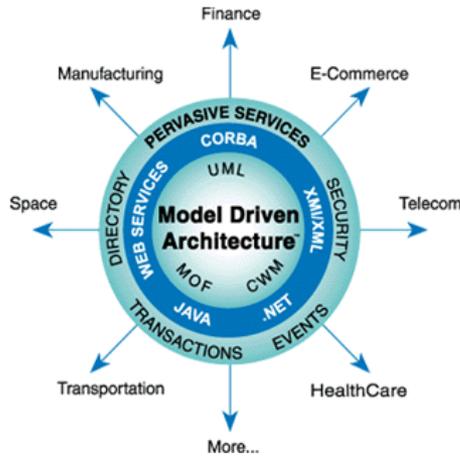


All good, but none has all you need!

The lack of a sound, widely accepted theoretical basis.

Everyone of us knows how to
develop **our** software,
but as a community we have **no**
widely accepted common ground

The huge number of methods...



- The problem is not the huge number but
 - We cannot compare them so we cannot learn from them
 - We cannot select the best practices from them
 - Not even within a large company do we know which methods we have.
 - For the individual practitioner she has no solid knowledge which she can take from work to work.

The lack of credible experimental evaluation and validation.

We throw out our old method and
replace it with a new popular one
without fully understanding the
consequences

---\$\$\$\$

The split between industry practice and academic research.

Industry Practice



Academic Research

We need to do something about it!

CALL FOR ACTION – part 1

- Software engineering is gravely hampered today by **immature practices**. Specific problems include:
 - The prevalence of fads more typical of **fashion industry** than of an engineering discipline.
 - The lack of a sound, **widely accepted theoretical basis**.
 - The **huge number of methods** and method variants, with differences little understood and artificially magnified.
 - The lack of credible experimental evaluation and validation.
 - The split between industry practice and academic research.

Agenda

- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

The Semat initiative

SOFTWARE ENGINEERING METHOD AND THEORY



www.semat.org

Founded by the Troika in

Sep 2009:

Ivar Jacobson
Bertrand Meyer
Richard Soley



What are we going to do about it?

CALL FOR ACTION – part 2

- We support a process to **refound software engineering** based on **a solid theory, proven principles and best practices** that:
 - Include **a kernel of widely-agreed elements**, extensible for specific uses
 - Addresses both **technology** and **people** issues
 - Are supported by industry, academia, researchers and users
 - Support extension in the face of changing requirements and technology

This is the Grand Vision

What are we going to do about it?

CASE FOR ACTION - part 2

•
Standing on a
Solid Theory

nd softw
and be
eed elem
nd peopl
ademia,
of chang

Including a
Kernel of
Widely-Agreed
Elements

This is the Grand Vision

What Semat is not

~~A New
Unfilled
Method~~

~~Competing
With Existing
Methods~~

Signatories

For current list, please see www.semat.org

- Pekka Abrahamsson,
- Scott Ambler,
- Victor Basili,
- Jean Bézivin,
- Robert V. Binder
- Dines Bjorner,
- Barry Boehm,
- Alan W. Brown,
- Larry Constantine,
- Steve Cook,
- Bill Curtis,
- Donald Firesmith,
- Erich Gamma,
- Carlo Ghezzi,
- Tom Gilb,
- Ellen Gottesdiener,
- Sam Guckenheimer,
- Robert Glass,
- David Harel
- Brian Henderson-Sellers,
- Martin Griss,
- Capers Jones,
- **Ivar Jacobson**,
- Philippe Kruchten,
- Harold Lawson,
- Robert Martin,
- **Bertrand Meyer**,
- James Odell,
- Meilir Page-Jones,
- Dieter Rombach,
- Ken Schwaber,
- Alec Sharp,
- **Richard Soley**.
- Andrey Terekhov
- Fuqing Yang
- Ed Yourdon



Corporate and Academic Signatories

- ABB, Switzerland
- Chalmers, Sweden
- Ericsson, Sweden
- Fujitsu, UK
- Huawei, China ←
- IBM, USA
- Microsoft, Spain
- KAIST, Korea
- Peking University, China ←
- Royal Institute of Technology, Sweden
- SAAB, Sweden
- Samsung SDS, Korea
- Swedish Institute of Computer Science, Sweden
- SINTEF, Norway
- Software Engineering Center, Korea
- SEI, USA
- Telecom Italia, Italy
- City of Toronto, Ontario, Canada
- Wellpoint, USA

For current list, please see www.semat.org

Agenda

- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

Standing on a Solid Theory

Examples:

- Darwin's theory of natural selection,
- Maxwell's equations,
- Einstein's theory of relativity.

There is nothing so practical as a
good theory!

-- Kurt Lewin

Maxwell's Equations – Now, an established theory



This is the kind of result we desire for Semat –
but most likely of very different nature

Software Engineering doesn't lack theories

There are an abundant number of theories in software engineering:

- Boehm's The Constructive Cost Model – COCOMO,
- Parnas' principle of information hiding,
- Constantine's Cohesion and Coupling,
- Conway's Law,
- Dijkstra's theory of cognitive limits ("Go to statement considered harmful"),
- Wirth's stepwise refinement,
- Meyer's Design by Contract,
- Etc., etc., etc..

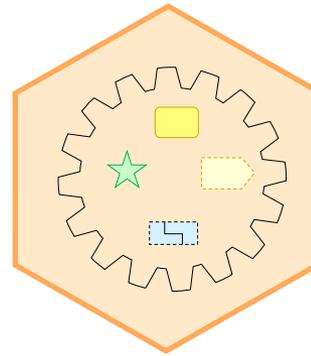
But none of these theories is foundational

Agenda

- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

Include a kernel of widely-agreed elements

- The Kernel should be harvested from a large number of methods
- The Kernel is practice and method agnostic.
- The Kernel includes elements which are universal for all software development efforts.

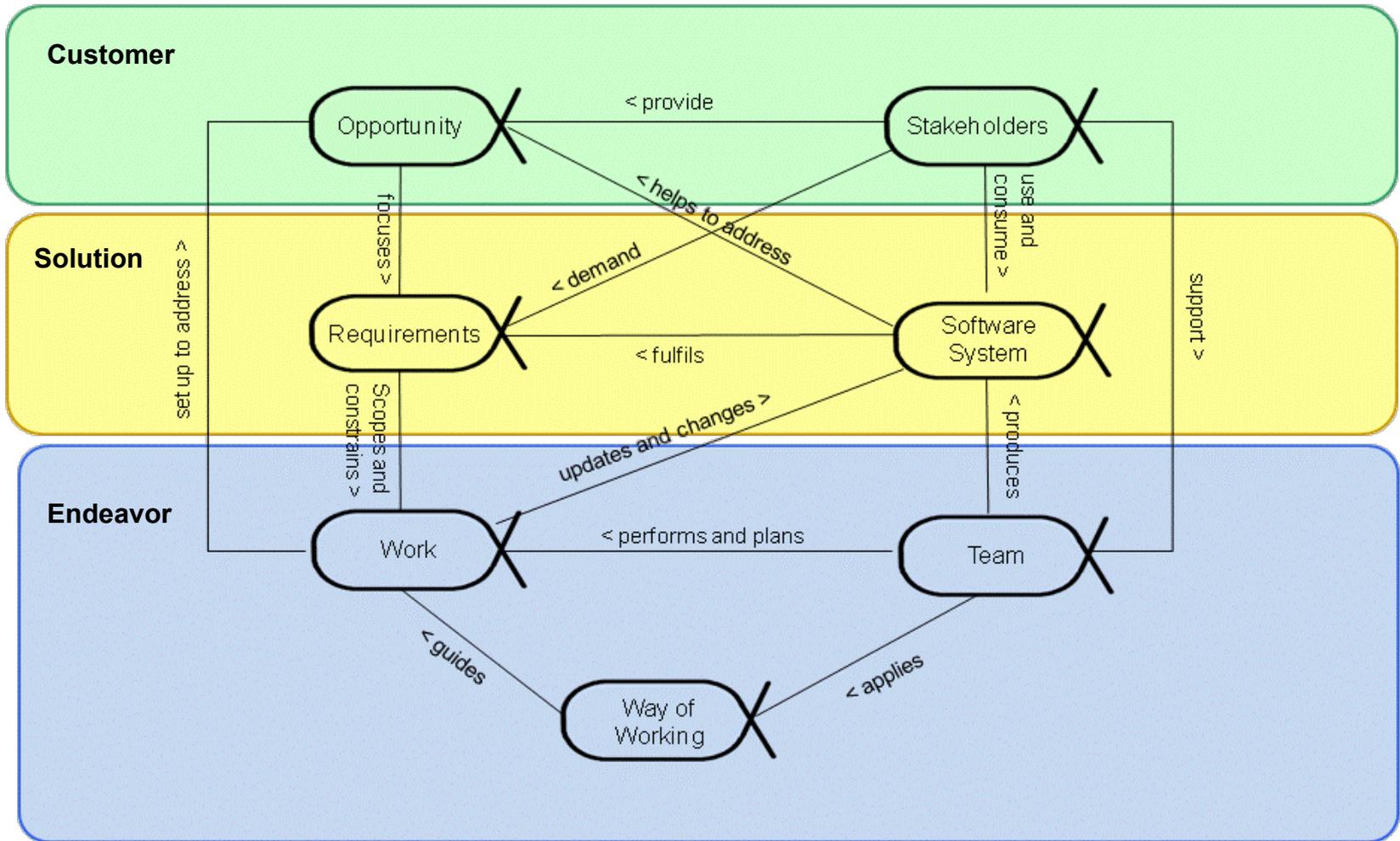


Kernel

The Kernel includes the essence of software engineering

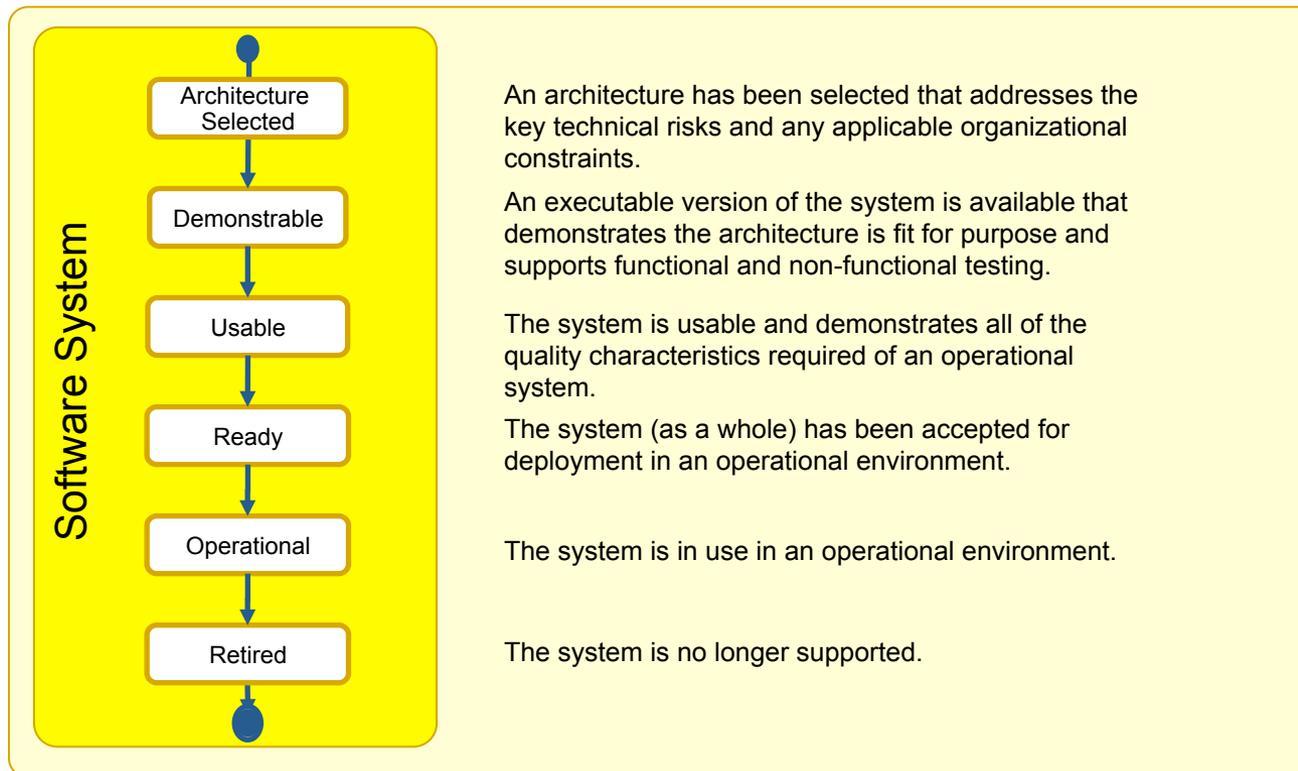
Michelangelo (attributed) “I am freeing the statue from the block”.
Paraphrasing him: “We are freeing the kernel from the methods”.

Inside the Kernel

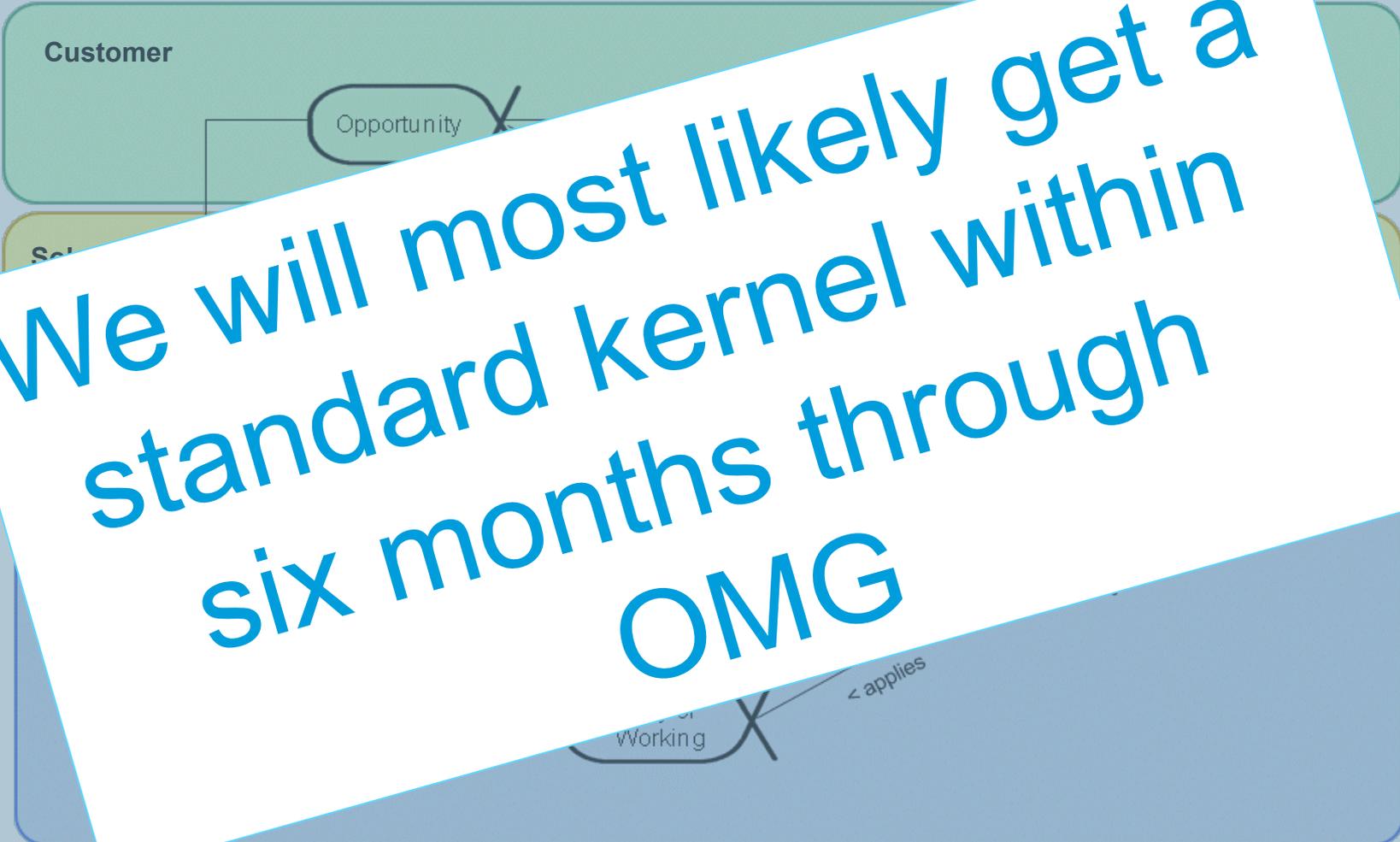


An Example of a Kernel Element

Software System: A system made up of software, hardware, and data that provides its primary value by the execution of the software. A software system can be part of a larger software, hardware, business or social solution.



We will most likely get a standard kernel within six months through OMG



Agenda

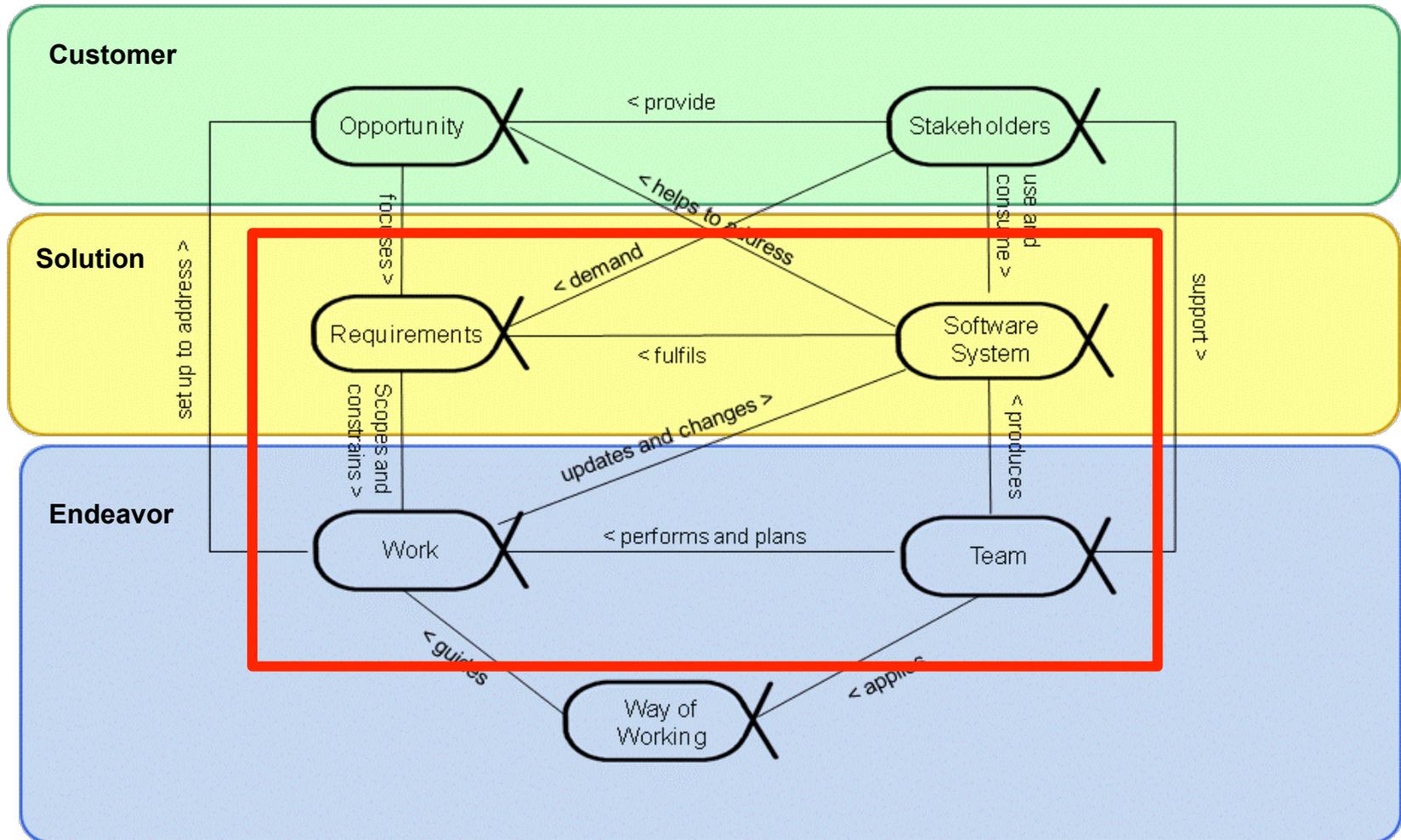
- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

A few words on how to use the kernel

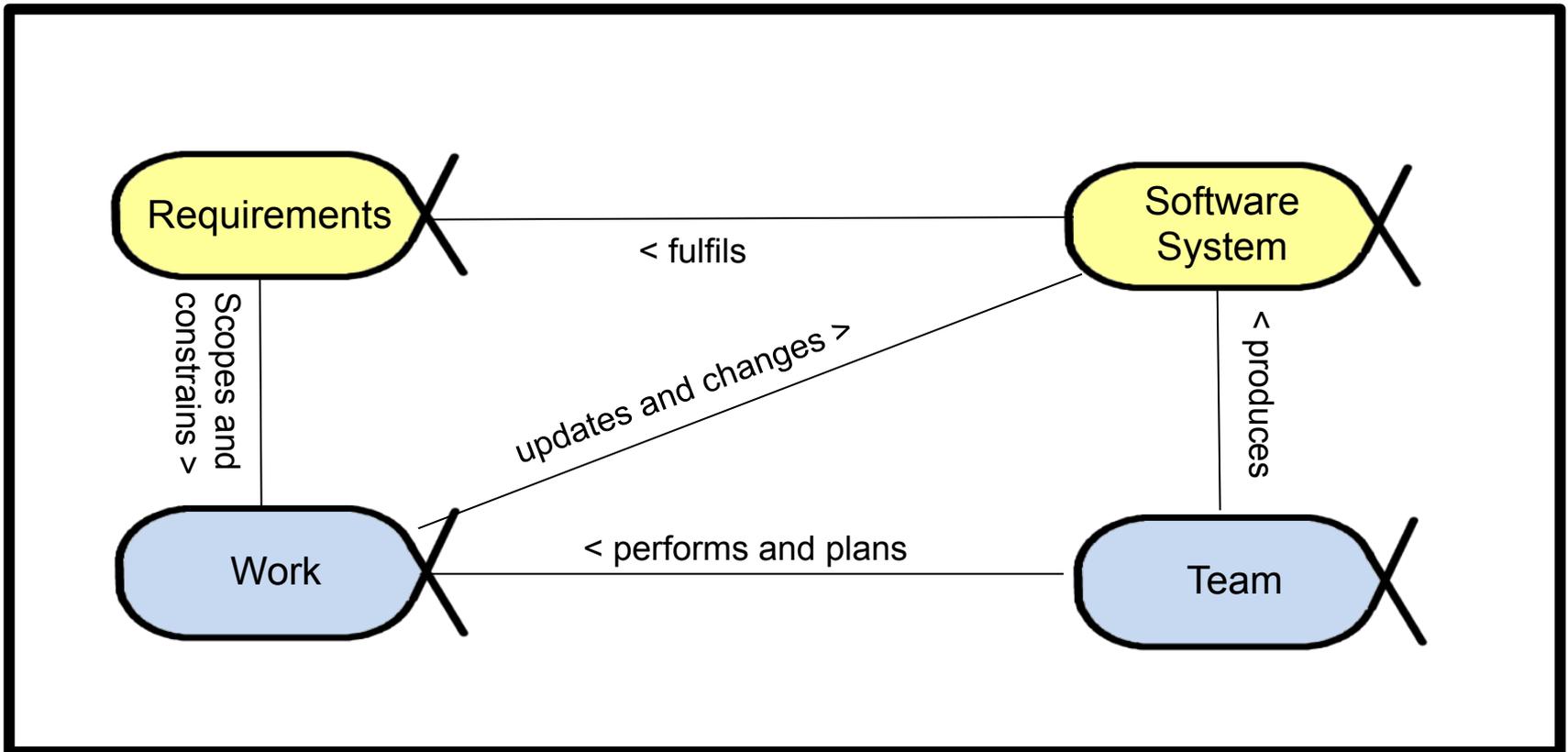
- Each kernel element represents one universally known concept in software engineering
 - Requirements, Software System, Work , Team, etc.
- Each kernel element has a series of states.
- Compactly described as a deck of cards
 - Can fit into your pocket.



An example: an extract from the kernel



An example: an extract from the kernel



Plan: Determine Current State

Requirements

<input type="checkbox"/> Requirements Conceived <ul style="list-style-type: none"> Need for system agreed by initial stakeholders Users and customers identified Expected benefit of system agreed <p>1 / 6</p>	<input type="checkbox"/> Requirements Bounded <ul style="list-style-type: none"> Theme, scope, success criteria of system is clear Mechanisms for managing requirements in place Constraints and assumptions considered <p>2 / 6</p>	<input type="checkbox"/> Requirements Coherent <ul style="list-style-type: none"> Described requirements provide coherent picture of the system Conflicting requirements separated Important usage scenarios explained Priority of requirements clear <p>3 / 6</p>	<input type="checkbox"/> Requirements Sufficient <ul style="list-style-type: none"> Requirements adequately describe solution and acceptable to stakeholders Rate of change to agreed requirements is low and under control <p>4 / 6</p>	<input type="checkbox"/> Requirements Satisfactory <ul style="list-style-type: none"> System implementing requirements is worth making operational Enough requirements are implemented <p>5 / 6</p>	<input type="checkbox"/> Requirements Fulfilled <ul style="list-style-type: none"> System implementing requirements is accepted as fully satisfying the need No outstanding requirement items prevent system from being accepted Stakeholders accept requirements as accurate <p>6 / 6</p>
--	---	--	--	---	---

Software System

<input type="checkbox"/> Software System Architecture Selected <ul style="list-style-type: none"> Architecture selected that address key technical risks Criteria for selecting architecture agreed Platforms, technologies, languages selected Buy, build, reuse decisions made <p>1 / 6</p>	<input type="checkbox"/> Software System Demonstrable <ul style="list-style-type: none"> Executable version of system demonstrates architecture is fit for purpose Supports functional and non-functional testing Critical interface and system configurations exercised <p>2 / 6</p>	<input type="checkbox"/> Software System Usable <ul style="list-style-type: none"> System is usable and has desired quality characteristics System can be operated by users Functionality and performance have been tested and accepted Defect levels acceptable Release content known <p>3 / 6</p>	<input type="checkbox"/> Software System Ready <ul style="list-style-type: none"> System (as a whole) has been accepted for deployment in operational environment Sponsors, users, stakeholders accept system as fit for purpose Installation and other documents available Operational support in place <p>4 / 6</p>	<input type="checkbox"/> Software System Operational <ul style="list-style-type: none"> System in use in operational environment System available to intended users At least one example of system is fully operational System supported to agreed service levels <p>5 / 6</p>	<input type="checkbox"/> Software System Retired <ul style="list-style-type: none"> System no longer supported Updates to system will no longer be produced System has been replaced or discontinued. <p>6 / 6</p>
---	--	--	---	--	---

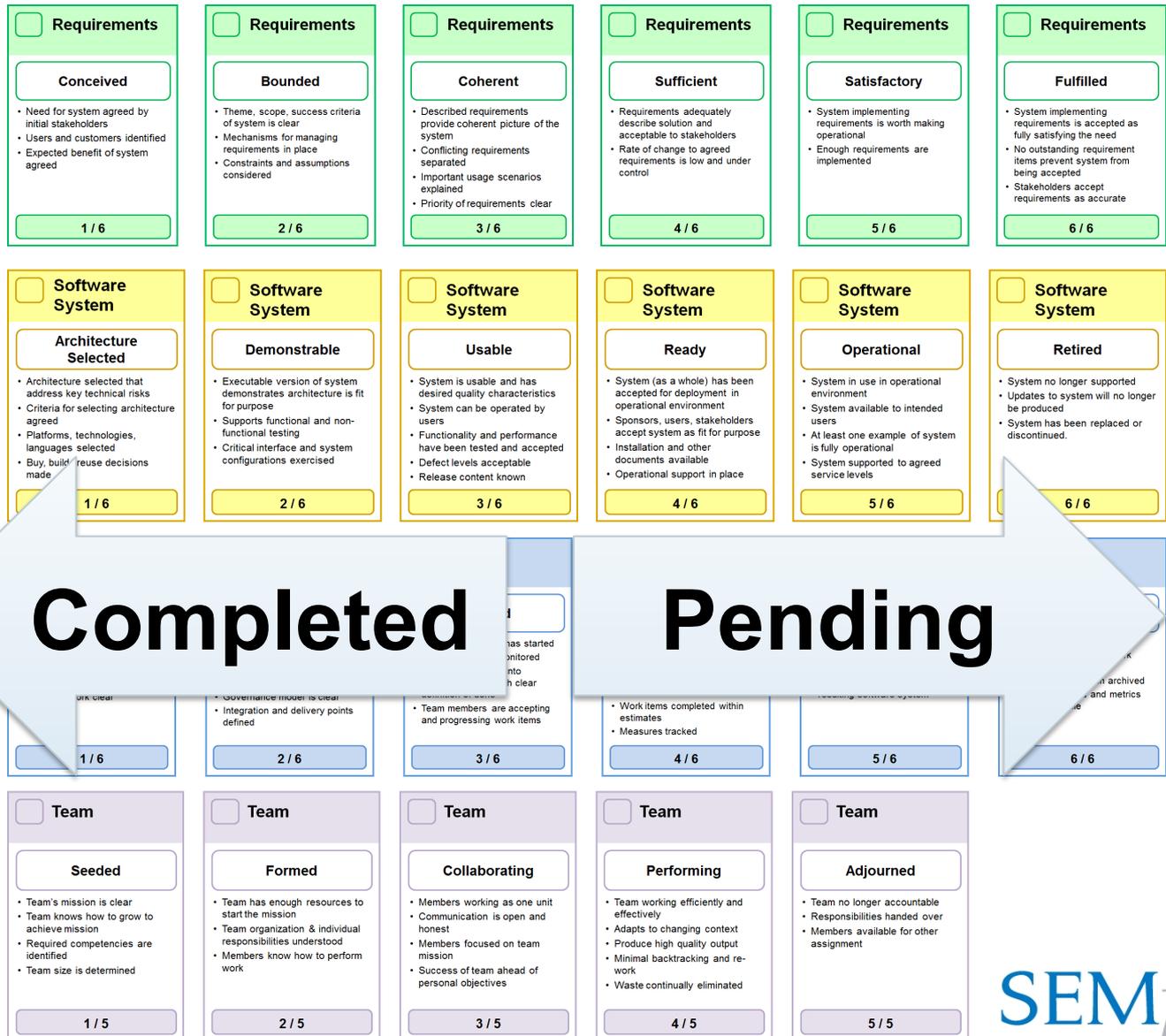
Work

<input type="checkbox"/> Work Initiated <ul style="list-style-type: none"> Work initiator and client known Work goal and constraints clear Sponsorship and funding model clear Priority of work clear <p>1 / 6</p>	<input type="checkbox"/> Work Prepared <ul style="list-style-type: none"> Cost & effort understood Funding in place Resource availability and risk exposure understood Governance model is clear Integration and delivery points defined <p>2 / 6</p>	<input type="checkbox"/> Work Started <ul style="list-style-type: none"> Development work has started Work progress is monitored Work broken down into actionable items with clear definition of done Team members are accepting and progressing work items <p>3 / 6</p>	<input type="checkbox"/> Work Under Control <ul style="list-style-type: none"> Work going well, risks being managed, productivity levels acceptable Unplanned work & re-work under control Work items completed within estimates Measures tracked <p>4 / 6</p>	<input type="checkbox"/> Work Concluded <ul style="list-style-type: none"> Work to produce results have been finished Work results are being achieved The client has accepted the resulting software system <p>5 / 6</p>	<input type="checkbox"/> Work Closed <ul style="list-style-type: none"> All remaining housekeeping tasks completed, and work officially closed Everything has been archived Lessons learned and metrics made available <p>6 / 6</p>
--	--	--	--	---	--

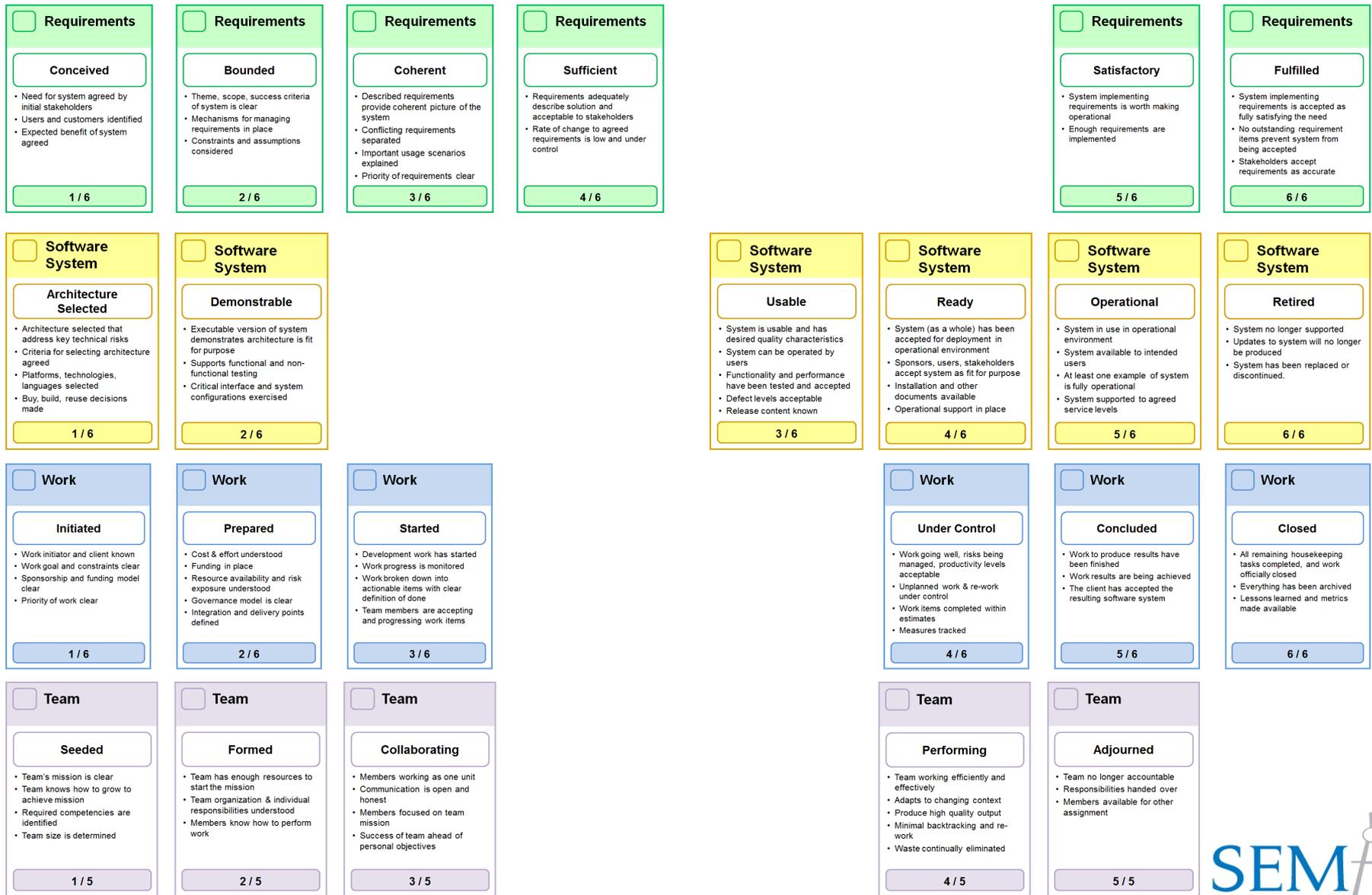
Team

<input type="checkbox"/> Team Seeded <ul style="list-style-type: none"> Team's mission is clear Team knows how to grow to achieve mission Required competencies are identified Team size is determined <p>1 / 5</p>	<input type="checkbox"/> Team Formed <ul style="list-style-type: none"> Team has enough resources to start the mission Team organization & individual responsibilities understood Members know how to perform work <p>2 / 5</p>	<input type="checkbox"/> Team Collaborating <ul style="list-style-type: none"> Members working as one unit Communication is open and honest Members focused on team mission Success of team ahead of personal objectives <p>3 / 5</p>	<input type="checkbox"/> Team Performing <ul style="list-style-type: none"> Team working efficiently and effectively Adapts to changing context Produce high quality output Minimal backtracking and re-work Waste continually eliminated <p>4 / 5</p>	<input type="checkbox"/> Team Adjourned <ul style="list-style-type: none"> Team no longer accountable Responsibilities handed over Members available for other assignment <p>5 / 5</p>
---	--	---	---	---

Plan: Determine Current State



Plan: Determine Next State



Plan: Determine How to Achieve Next State

Requirements

Satisfactory

- System implementing requirements is worth making operational
- Enough requirements are implemented

5 / 6

Software System

Usable

- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Work

Under Control

- Work going well, risks being managed, productivity levels acceptable
- Unplanned work & re-work under control
- Work items completed within estimates
- Measures tracked

4 / 6

Team

Performing

- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

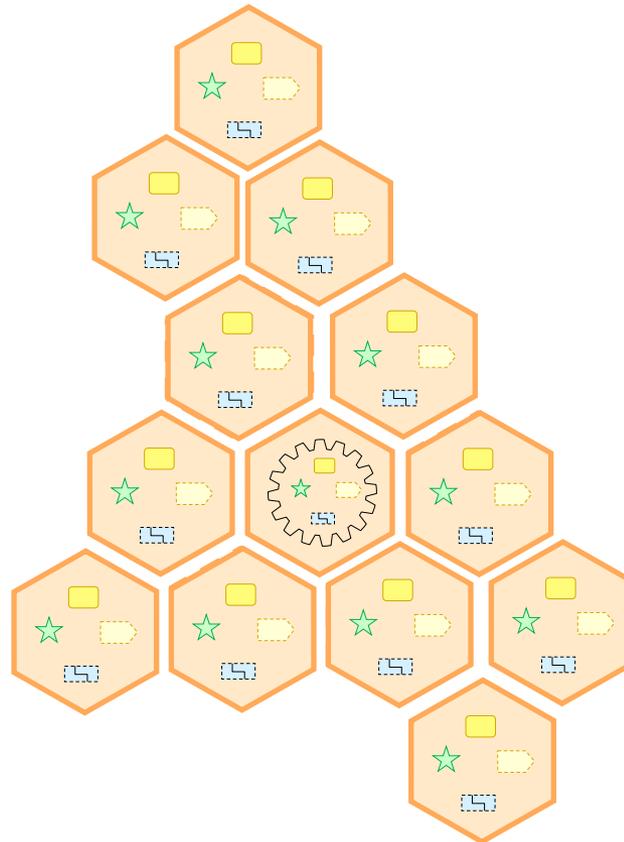
4 / 5

Agenda

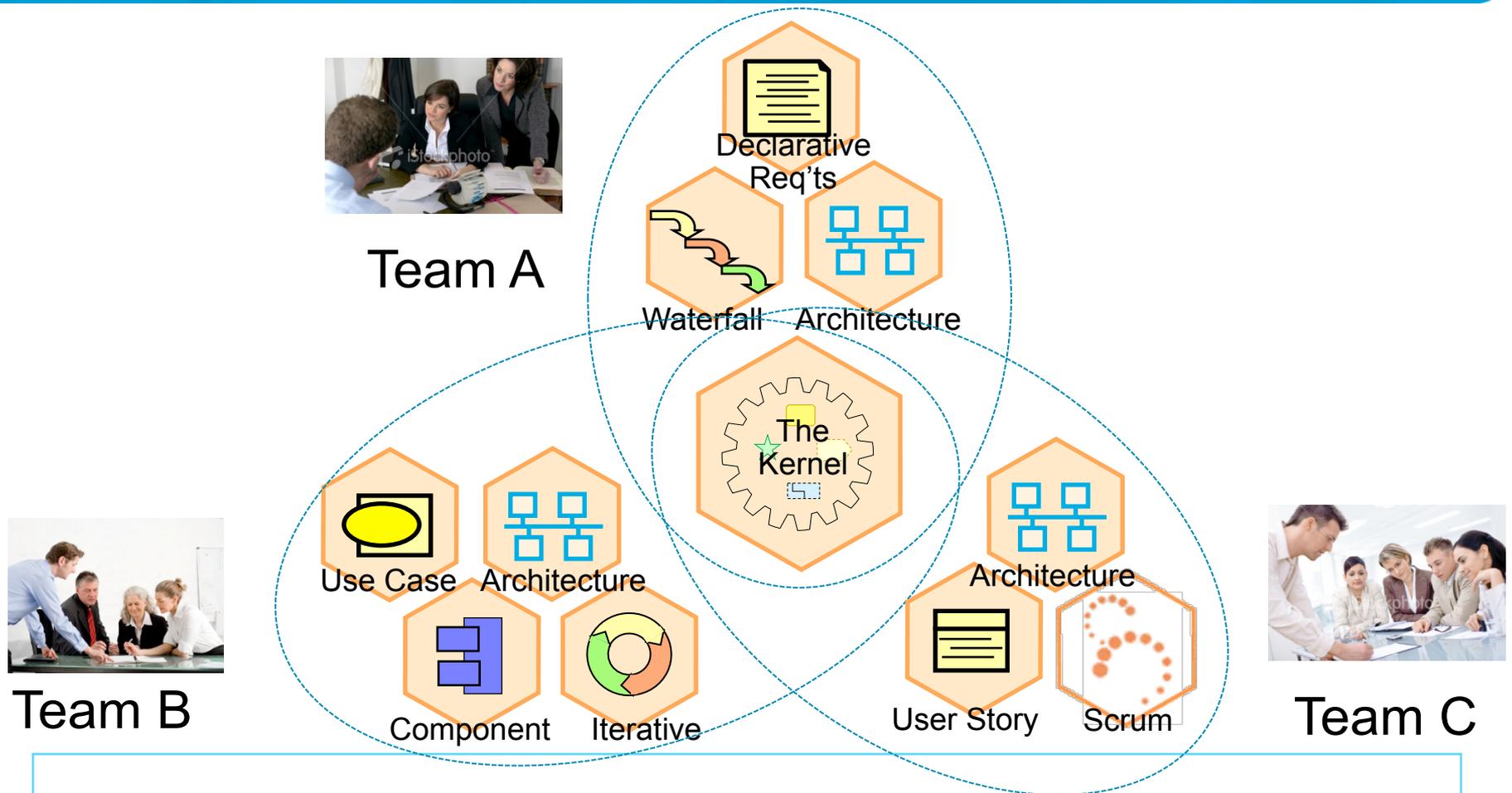
- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

Extending the Kernel with Practices

- The Kernel is Universal.
- By adding Practices on top of the Kernel you create a specific Method.
- Practices such as Use Cases, Test-Driven Development, Etc

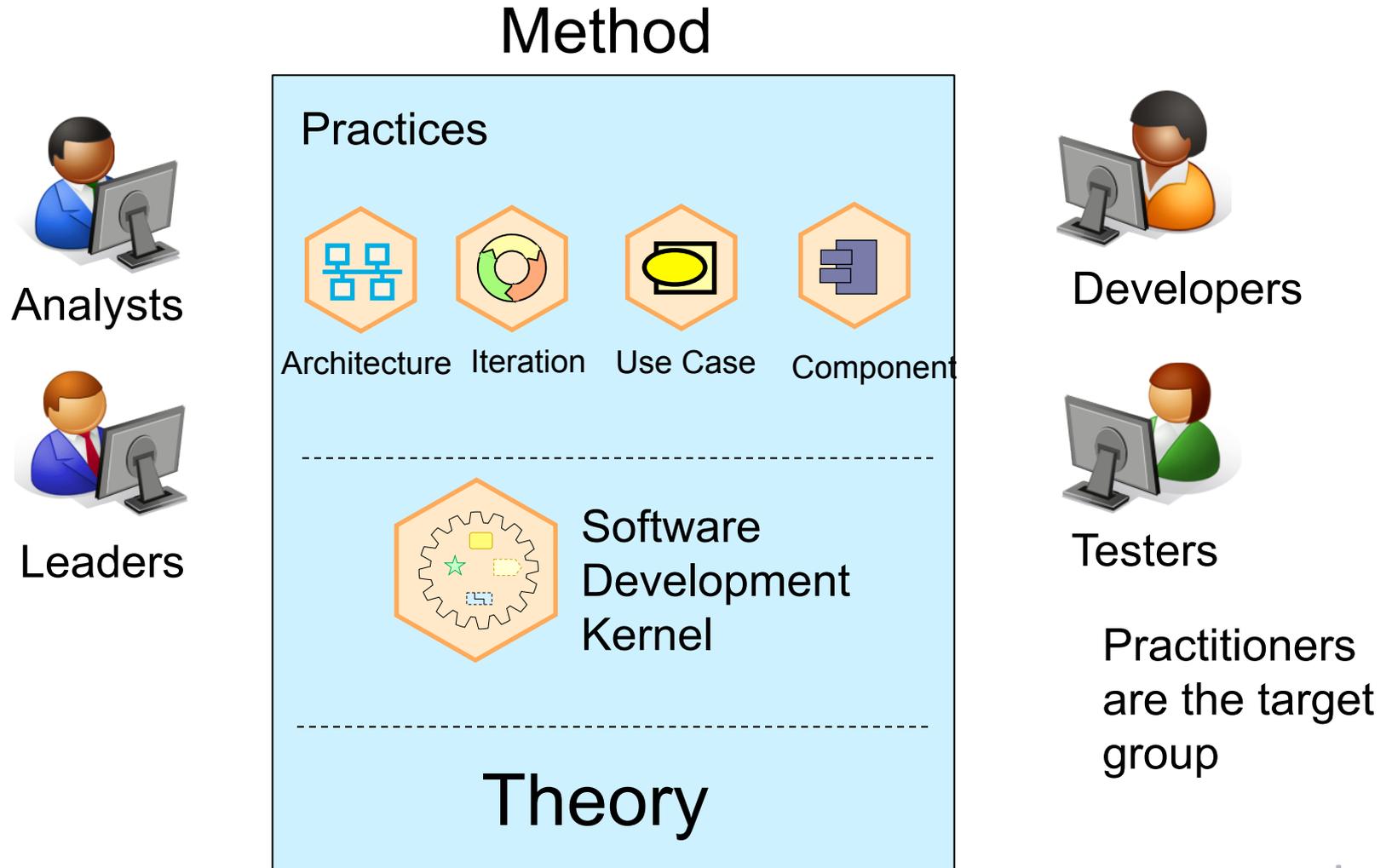


The Kernel provides a shared frame of reference



Many methods all built on the same kernel.

Summing up



Summary - What is new?

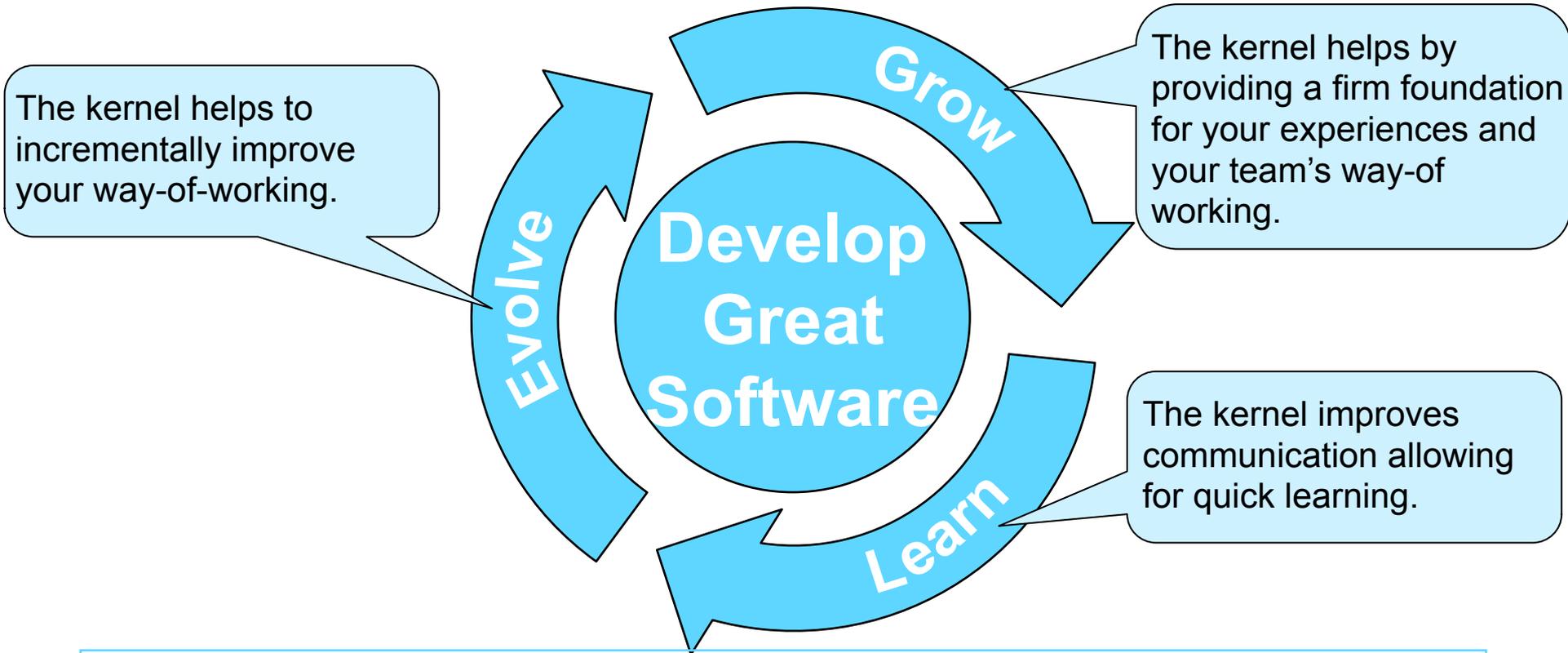
- Focus are the practitioners, not the process engineers
- Focus is Method Use and Adaption, not Method Description
- Semat is inclusive and not exclusive – it includes all relevant methods and practices (“good or bad”)
- There is a small Kernel of essential
- For small teams and large organizations
- Practices from the bottom, and not process from the top
- Separation of concerns is a fundamental principle
- Light and agile in working with methods
- The old metaphor: ‘Process is program – Team is computer’ is gone
- The process is what the team does. Adaption happens dynamically as a retrospective of what the team did through a feedback loop.

Watts Humphrey (CMMI): “This (SEMAT) meeting in Zurich (2010) is likely to be an historic occasion much like the 1968 NATO session in Garmish.”

Agenda

- The State of the Industry
- Specific Problems
- A Call for Action
 - Standing on a Solid Theory
 - Including a Kernel of Widely-Agreed Elements
- Using the Kernel
- Extending the Kernel
- What's in it For You? Final Words

What is in it for you as a Software Professional?



Your objective is to develop great software quickly.
The kernel helps you to learn, evolve and grow

What's in it for you

Industry

Big companies have many processes.

Challenges:

- Reuse practices
- Reuse training
- “Reuse” of people
- Evolutionary improvement is hard

Software Professional

Want to become experts. Challenges:

- Their skills are not easily reusable for a new product.
- Their career path follows a zig-zag track from hype to hype.

Academics

Asked to educate and research. Challenges:

- The Gap between research and industry
- No widely accepted theory
- Teaching instances of methods don't create generalists

SEMAT targets the whole software community.

Agenda

- The State of the Industry
- Specific Problems
- A Case for Action
 - Based on a solid theory
 - Include a kernel of widely-agreed elements
- How to Use the Kernel
- What's in it For You?
- Wrap Up / Final Words

What you can do now

Become a supporter. Go to
www.semat.org
Join the China Chapter of Semat



Software Engineering Method and Theory
This is the Official SEMAT Website

[Home](#)

[Supporters \(signup\)](#)

[Call for Action](#)

[Organization](#)

[Signatories](#)

[Publications](#)

[Q&A](#)

[Semat in the News](#)

[Contact us](#)

OMG Proposal Submission

February 22, 2012 in [News](#)

We are happy to let you know that on February 20, 2012 the working group of Semat has submitted the proposal, entitled "Essence – Kernel and Language for Software Engineering", to respond to the OMG RFP: Foundation for the Agile Creation and Enactment of Software Engineering Methods (FACESEM) RFP (OMG Document ad/2011-06-26). The proposal provides comprehensive definitions and descriptions of the kernel and the language of software engineering methods, which meet (address) the mandatory requirements set forth in the FACESEM RFP.

This proposal can be found here: ["Essence – Kernel and Language for Software Engineering"](#).

The working group has been working intensively for the past a couple of months in preparing for this submission. The next step is to give a presentation during an OMG meeting in March in Washington D.C. There are 3 initial submissions which will all present during the OMG meeting.

[Search](#)

Navigation

[News](#)

[Old News](#)

[Acknowledgments](#)

[SEMAT Blog](#)

[SEMAT Supporter Signup](#)

Recent Posts

[OMG Proposal Submission](#)



What you can do now

Become a supporter. Go to

www.semat.org

Join the China Chapter of Semat

Read the OMG submission

Essence – Kernel and Language for Software Engineering Methods

Initial Submission – Version 1.0

In response to: Foundation for the Agile Creation and Enactment of Software Engineering Methods (FACESEM) RFP (OMG Document ad/2011-06-26)

This proposal can be found here: "Essence – Kernel and Language for Software Engineering".

The working group has been working intensively for the past a couple of months in preparing for this submission. The next step is to give a presentation during an OMG meeting in March in Washington D.C. There are 3 initial submissions which will all present during the OMG meeting.



Software Engineering Method and
This is the Official SEMAT Website

Home

Supporters (signup)

Call for Action

Organization

Signatories

Publications

Q&A

Semat in the News

Contact us

OMG Prop

February 22, 2012 in

We are happy to let you know that the "Essence – Kernel and Language for Software Engineering Methods" and Enactment of Software Engineering Methods (FACESEM) RFP (OMG Document ad/2011-06-26) comprehensive definition of Software Engineering Methods (SEM) (address) the mandate

 Search

Navigation

- News
- Old News
- Acknowledgments
- SEMAT Blog
- SEMAT Supporter Signup

Recent Posts

OMG Proposal Submission



What you can do now



SEMAT Software Engineering Methodology
This is the C

Home [Supporters \(signup\)](#) [Call for Action](#)

OMG Prop: Essence – Kernel Software Engineering

February 22, 2012 in [News](#)

We are happy to let you know that the initial submission of the “Essence – Kernel and Enactment of Software Engineering” proposal has been accepted for consideration. The address of the mandate is: [http://www.omg.org/standards/mandates/mandate.html](#)

This proposal can be found here: “Essence – Kernel Software Engineering”

The working group has been working intensively for the past few weeks. The next step is to give a presentation during an OMG meeting. We will all present during the OMG meeting.

Initial Submission – Kernel Software Engineering

In response to: Foundation for the Establishment of Software Engineering (OMG Document ad/2011-06-26)

The Essence of Software Engineering

Applying the Semat Kernel

David Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, Svante Lidman.



Read my new book with Pan Wei and three other top guys

mat

Search

Navigation

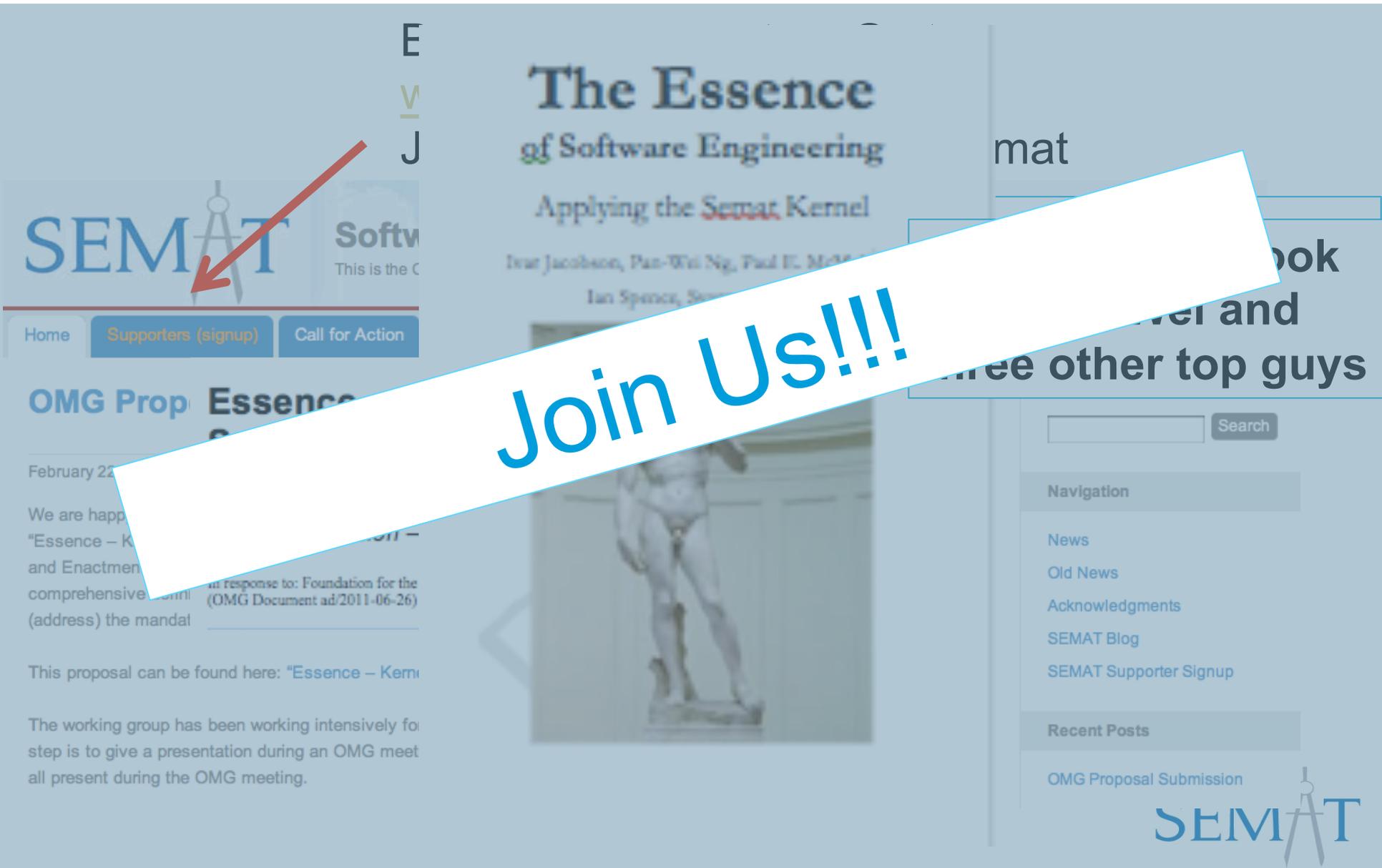
- [News](#)
- [Old News](#)
- [Acknowledgments](#)
- [SEMAT Blog](#)
- [SEMAT Supporter Signup](#)

Recent Posts

- [OMG Proposal Submission](#)



What you can do now



Home [Supporters \(signup\)](#) [Call for Action](#)

OMG Prop Essence

February 22
We are happy
"Essence – Kernel
and Enactment
comprehensive
(address) the mandat

This proposal can be found here: "Essence – Kernel

The working group has been working intensively for
step is to give a presentation during an OMG meet
all present during the OMG meeting.

The Essence of Software Engineering

Applying the ~~SEMAT~~ Kernel

Ivar Jacobson, Pan-Wei Ng, Paul E. McMillan,
Ian Spence, Steve



mat

ook
ver and
three other top guys

Navigation

- [News](#)
- [Old News](#)
- [Acknowledgments](#)
- [SEMAT Blog](#)
- [SEMAT Supporter Signup](#)

Recent Posts

[OMG Proposal Submission](#)



Join Us!!!

References

1. Ivar Jacobson and Bertrand Meyer: “Methods need theory” Dr. Dobb's Journal, August 06, 2009. Online at <http://www.drdobbs.com/architecture-and-design/219100242>
2. Ivar Jacobson and Ian Spence: “Why we need a theory for software engineering” Dr. Dobb's Journal, October 02, 2009. Online at <http://www.drdobbs.com/architecture-and-design/220300840>
3. Ivar Jacobson, Bertrand Meyer, and Richard Soley: “Call for Action: The Semat Initiative” Dr. Dobb's Journal December 10, 2009. Online at <http://www.drdobbs.com/architecture-and-design/222001342>
4. Ivar Jacobson, Bertrand Meyer, and Richard Soley: “The Semat Vision Statement” online at <http://blog.paluno.uni-due.de/semat.org/wp-content/uploads/2012/03/SEMAT-vision.pdf>
5. Ivar Jacobson, Shihong Huang, Mira Kajko-Mattsson, Paul McMahon, Ed Seymour. “Semat - Three Year Vision” Programming and Computer Software 38(1): 1-12 (2012), Springer 2012. DOI: 10.1134/S0361768812010021.
6. “Essence – Kernel and Language for Software Engineering”(OMG Document number: ad/2011-02-04) A proposal submitted to OMG RFP (OMG Document ad/2011-06-26)

References

1. Ivar Jacobson and Bertrand Meyer: "Methods need theory" Dr. Dobbs Journal, August 06, 2009. Online at <http://www.drdobbs.com/architecture-and-design/219100242>
2. Ivar Jacobson and Ian Spence: "Why we need a theory for software engineering" Dr. Dobbs Journal, October 02, 2009. Online at <http://www.drdobbs.com/architecture-and-design/219100242>
3. Ivar Jacobson, Bertrand Meyer, and Richard Solev: "The Theory of Software Engineering" Dr. Dobbs Journal December 10, 2009. Online at <http://www.drdobbs.com/architecture-and-design/219100242>
4. Ivar Jacobson, Bertrand Meyer: "The Theory of Software Engineering" blog.paluno.uni-due.de
5. Ivar Jacobson: "The Theory of Software Engineering" www.semanticscholar.org/author/Ivar-Jacobson/123456789 - Three
2012. DOI: 10.1134/
6. <http://www.omg.org/spec/ACT/1.0/> (OMG Document number: ad/2011-06-26)
(OMG Document ad/2011-06-26)

You can find these and more on www.semat.org

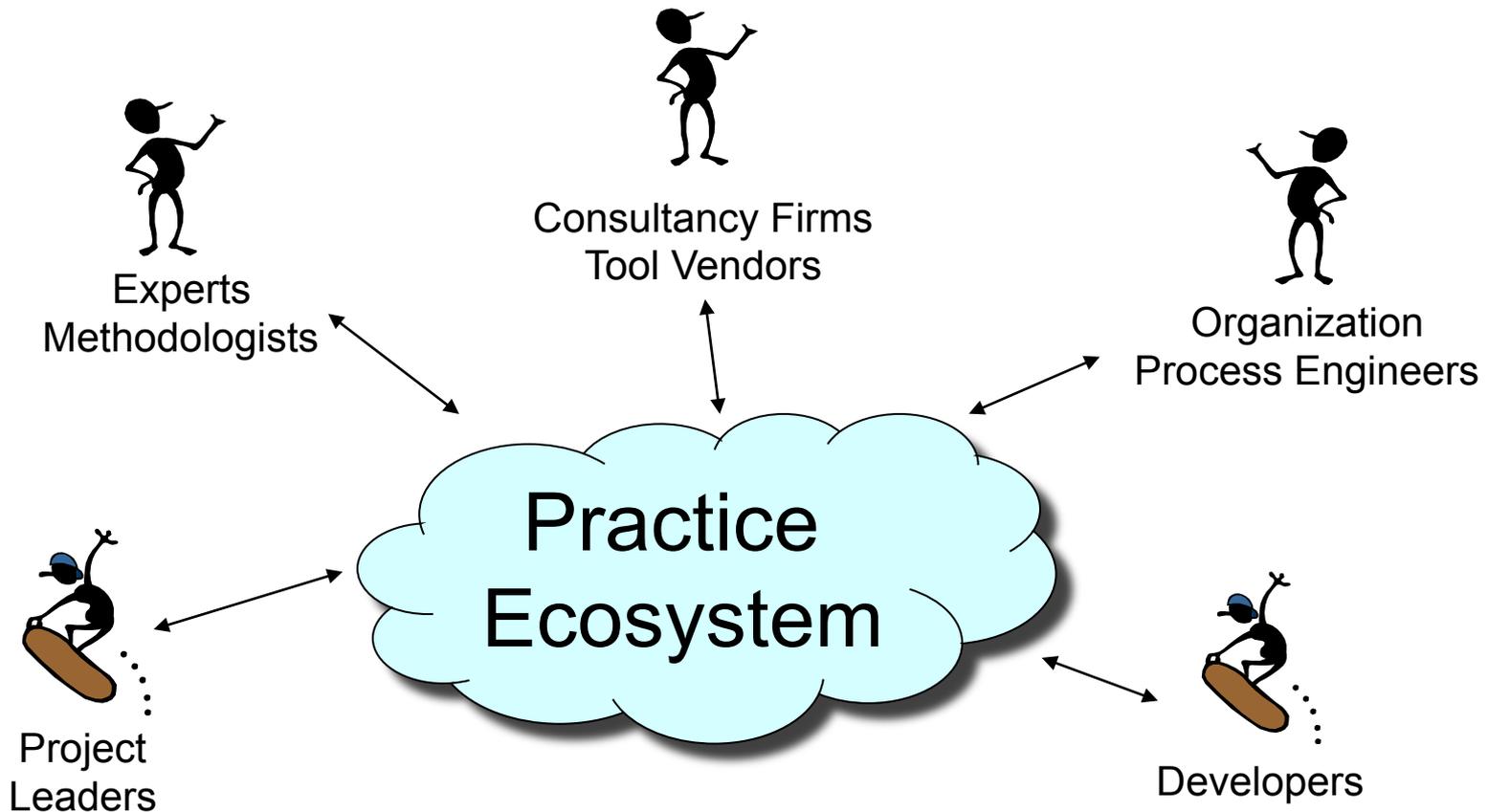
Questions



- Email:
 - ivar@ivarjacobson.com
- OMG website:
 - www.omg.org
- SEMAT website:
 - <http://www.semat.org>

There will be hundreds of practices

- Most will be contributed by the community



- I would be very glad to discuss with you what knowledge base (theory) could help us explain and predict the pertinent phenomena of software engineering. I propose that we are looking for a way to answer the main questions of the field, including the following:
 - Is method A better than method B for my organization? Why?
 - Is it true that adding manpower to a late software project makes it later (Brook's Law)? Why?
 - Is it true that a small sharp team, with as few minds as possible, is best? Why?
 - Is it true that the GOTO statement is harmful (as posited by Dijkstra)? Why?