The Smarter Way: SEMAT

Ivar Jacobson
As an Industry

Everyone of us knows how to develop our software, but as an industry we don’t know it

We don’t have a widely accepted foundation

Who said that? 😊
Software Development is driven by fashions and fads

- Fifteen years ago it was all about OO
- Ten years ago it was about components, UML, Unified Process
- Five years ago it was about RUP and CMMI
- Two years ago it was about XP
- Today it is about Scrum
- Next year it is about Lean
- Kanban?

All good, but none is all you need

The software industry keeps looking for a silver bullet
What is Semat?

An attempt to re-found software engineering on sound principles, best practices and theory

Started by “the troika” (Ivar Jacobson, Bertrand Meyer, Richard Soley)

Timeline:
- Initial papers (by Ivar and Bertrand): mid-2009
- Call for Action (now Richard had joined us): Sept. 2009
- Joined by signatories and corporate signatories
- Over a thousand “supporters”
- Vision Statement, foundational meeting: March 2010
- First milestone: March 2011

SOFTWARE ENGINEERING METHOD AND THEORY
Welcome to become a supporter www.semat.org
A CALL FOR ACTION STATEMENT

• 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.

This is not smart!
CALL FOR ACTION STATEMENT cont’d

• 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 smart!
Signatories as of June 30, 2010

- Pekka Abrahamsson,
- Scott Ambler,
- Victor Basili,
- Jean Bézivin,
- Dines Bjørner,
- Barry Boehm,
- Alan W. Brown,
- Larry Constantine,
- Steve Cook,
- Bill Curtis,
- Donald Firesmith,
- Erich Gamma,
- Carlo Ghezzi,
- Tom Gilb,
- Ellen Gottesdiener,
- Sam Guckenheimer,
- Robert Grass,
- David Harel

- Brian Henderson-Sellers,
- Watts Humphrey,
- Martin Griss,
- Capers Jones,
- Ivar Jacobson,
- Philippe Kruchten,
- Robert Martin,
- Stephen Mellor,
- Bertrand Meyer,
- James Odell,
- Meilir Page-Jones,
- Dieter Rombach,
- Ken Schwaber,
- Alec Sharp,
- Richard Soley,
- Andrey Terekhov
- Ed Yourdon
Signatories as of June 30, 2010

- Pekka Abrahamsson,
- Scott Ambler,
- Victor Basili,
- Jean Bézivin,
- 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 Grass,
- David Harel

- Brian Henderson-Sellers,
- Watts Humphrey,
- Martin Griss,
- Capers Jones,
- Ivar Jacobson,
- Philippe Kruchten,
- Robert Martin,
- Stephen Mellor,
- Bertrand Meyer,
- James Odell,
- Meilir Page-Jones,
- Dieter Rombach,
- Ken Schwaber,
- Alec Sharp,
- Richard Soley.
- Andrey Terekhov
- Ed Yourdon

For current list, please see www.semat.org

Agile, Iterative, RUP, Computer science, Metrics, CMMI, Etc.
Corporate Signatories as of May 9, 2010

- ABB
- Ericsson
- Fujitsu UK
- IBM
- Microsoft, Spain
- SAAB
- Samsung SDS
- Software Engineering Center - Korea
- Telecom Italia
- City of Toronto, Ontario
- Wellpoint

For current list, please see www.semat.org
Semat is separated into six tracks:

1. Definitions
2. Theory
3. Universals
4. Language
5. Assessment
6. Requirements

Results within 12 months
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
- There must be a kernel
- The Semat kernel: track 3 and 4
- If successful what impact can we expect?
- Wrap up
What went well and what went wrong

The *perceived* “rise and fall” of RUP

Let’s be clear, the “rise and fall” are all about perception. RUP is still very much alive.

“Good”

- Many proven practices
  - Use-cases (incl test)
  - Iterations
  - Components
  - Architecture
  - Etc.
- Supported UML
  - UML replaced all the hundred modeling languages at the time

“Bad”

- A soup of practices
- Too big
  - People don’t read process books
- Hard to extend with agile, CMMI, etc.
- Adoption extremely hard
  - Process savvy
  - Revolutionary
- Gap between what people said they did and what they really did – The Process Gap
We looked for fundamental changes.

“Bad”
- A soup of practices
- Too big
  - People don’t read process books
- Hard to extend with agile, CMMI, etc.
- Adoption extremely hard
  - Process savvy
  - Revolutionary, not evolutionary
- Gap between what people said they did and what they really did – The Process Gap

Fixing what was “Bad”
- Make practices first class citizens, and process a composition of practices
- Focus on the essentials instead of trying to be complete
- Extensions through practices
- A new user experience with focus on developers, not on process engineers.
- Enact the process

We redesigned RUP as EssUP
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
- There must be a kernel
- The Semat kernel: track 3 and 4
- If successful what impact can we expect?
- Wrap up
In the future, an ever present but invisible process

Process becomes second nature

We need a new paradigm

From the successes in modern software development

Practice is a First Class Citizen
the unit of adoption, planning and execution of process

The team’s way-of-working is just a composition of Practices

Examples:
Unified Process
CMMI, Spice
XP, Scrum

The Software Engineering Camp
Process Maturity Camp
Agile Methods Camp
We needed a shared definition of “practice”

A practice is a separate concern of a development method
- consisting of a set of activities with a clear beginning and end,
- performed by a set of individuals with specific competencies,
- when applied resulting in a set of new or modified artifacts of measureable value to the stakeholders of the software product being developed.

Examples:
1. ‘Iterative development from start of project until deployment’. Alternatively, ‘Waterfall development’ with the same scope.
2. ‘Use case driven development from requirements to test’. Alternatively, ‘Feature-driven development’ with the same scope.

Key ideas:
1. Practices are the elements that should be made lean (as a consequence a practice should include both do-activities and verify-activities).
2. Practices are the elements that need to be measured.
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
- There must be a kernel
- The Semat kernel: track 3 and 4
- If successful what impact can we expect?
- Wrap up
**Focus on the Essentials**

**What is Essential?**
- It is the key things to do and the key things to produce
- It is about what is important about these things
- It is less than a few percent of what experts know about these things
  - Law of nature: People don’t read process books
- It is the placeholders for conversations
  - Law of nature: People figure out the rest themselves
  - Training helps
- It is the base for extensions

Starting with the essentials makes a practice adoptable.
Why Cards?

- Cards are tactile
- Cards are simple and visual
- Cards use conversational and personalized style
- Cards are not prescriptive so they get the learner to think more deeply
- Cards get… and keep… the readers attention
- Cards promote agility
- They can be written on to make minor adjustments to the practice on the fly

A practice is a set of cards

A team works on a set of instance cards
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
- Result
  - There must be a kernel
  - The Semat kernel: track 3 and 4
  - If successful what impact can we expect?
- Wrap up
Practices are enacted

Set Up Your Goals

Get Help To Reach Your Goals

Things to do

Things to produce
Agenda

• On what went well and what went wrong
• Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
• There must be a kernel
• The Semat kernel: track 3 and 4
• If successful what impact can we expect?
• Wrap up
Thus we fixed what didn’t work

Fixing what was “Bad”
- Make practices first class citizens
- Focus on the essentials
- Extensions through practices
- A new user experience with focus on developers
- Enact the process to close the gap

Great, but now more became evident!
Agenda

• On what went well and what went wrong
• Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
• There must be a kernel
• The Semat kernel: track 3 and 4
• If successful what impact can we expect?
• Wrap up
Hypothesis harvested from the fixing-the-problem work

- All methods comprise of a set of things that are always there - documented or not.
- We called this set the Kernel.
- Every method can then be described as a set of composed practices using the kernel.

There is a kernel!
Many different methods can be built out of this same kernel.
To verify the hypothesis we started all over

- We called our initiative EssWork (moving beyond EssUP)
- The Kernel we harvested is very small, extracted from a large number of methods
- It contains empty slots for things that every process have
  - Slots for
    - Competencies, such as analyst, developer, tester
    - Things to work with, such as backlog, implementation, executable system
    - Things to do, such as implement the system, test the system
- The Kernel is practice and of course method agnostic.
The EssWork Kernel

- contains empty slots for things that every process have

**Things to Work with**

- **Opportunity**
- **Requirements**
- **System**
- **Team**
- **Project**
- **Way of Working**

**Patterns To Apply**

- [ ]
- [ ]
- [ ]
- [ ]

**Things to Do**

- **Understand the Need**
- **Ensure Stakeholder Satisfaction**
- **Accept the System**
- **Specify the System**
- **Shape the System**
- **Implement Software**
- **Test the System**
- **Release the System**
- **Establish Project**
- **Steer Project**
- **Support Team**
- **Conclude Project**

**Competencies**

- **Customer Representative**
- **Analyst**
- **Developer**
- **Tester**
- **Leadership**
Using the kernel

Kernel
The kernel defines an “empty process”

Practice
Each practice contains practice-specs to add to the kernel.

Practices “slot” into the common kernel.

Way of Working
Change starts by harvesting your best practices from your own method.
Improve your method by adding other, proven practices

OK, there is a kernel!
Maybe there are many?
But none is widely-accepted!
That needs to be changed!
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
- There must be a kernel
- The Semat kernel: track 3 and 4
- If successful what impact can we expect?
- Wrap up
CALL FOR ACTION 2nd part

• 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

The Kernel ≈ The Kernel Language + The Universals
The Envisioned Kernel

Level 3

Methods

Composed of

Level 2

Practices

Defined in terms of

Patterns

Level 1

Universals

The kernel

Kernel language
Agenda

- On what went well and what went wrong
- Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
- There must be a kernel
- The Semat kernel: track 3 and 4
- If successful what impact can we expect?
- Wrap up
A recipe for success

Our work needs to be
• driven from the demands of the industry/developer community, and
• enabled and formulated by the research community, and
• popularized by the methodologists.

We need a theoretical basis that is widely shared and supported, one that crosses the boundaries between the different software development camps.
### Industry
Big companies have many processes.
Challenges:
- Reuse practices
- Reuse training
- “Reuse” of people
- Evolutionary improvement is hard

### Developers
Want to become experts. Challenges:
- Their skills are not easily transferable to 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 doesn’t create generalists

### Methodologists
Every method is a soup of practices. Challenges:
- Have to reinvent the wheel

---

**SEMAT will have significant impact on the software community.**
Agenda

• On what went well and what went wrong
• Addressing what went wrong
  1. Practices
  2. A new user experience
  3. Practices are not dead, they are enacted
  4. Result
• There must be a kernel
• The Semat kernel: track 3 and 4
• If successful what impact can we expect?
• Wrap up
Path to SEMAT (personal)

- Following experience-based evolution:

  - 2003: Unified Process
  - 2006: Essential UP
    - Practices!
      - All methods are compositions of practices
  - 2009: The Essentials
    - Kernel!
      - Underlying all methods/practices is a kernel
  - Semat
    - Widely accepted a must!
      - The kernel/the practices need to be widely accepted

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

• **A Call for Action:**
  – The Software world is immature, and why.
  – We, signatories, corporate signatories, supporters will refound software engineering and how.
    – Signed by 35 well-known individuals and 11 corporations
    – Supported by 1200 practitioners around the world

• **Key idea:**
  1. All methods are just compositions of **practices**
  2. There is a **kernel** consisting of:
    – Things we always have, do and produce when we develop software – find them. We call them the **universals**.
    – A process **kernel language** used to describe practices and universals.
  3. Practices will be shared over all platforms and all methods.

Watts Humphrey: “This meeting in Zurich is likely to be an historic occasion much like the 1968 NATO session in Garmish.”
Questions
Thank You

ivar@ivarjacobson.com