Refounding Software Engineering: The Semat Initiative (Invited Presentation)

Mira Kajko-Mattsson
ICT/KTH Royal Institute of Technology
Stockholm, Sweden
mekm2@kth.se

Ivar Jacobson and
Ian Spence
Ivar Jacobson Int'l., UK
(ivar, ispence)@ivarjacobson.com

Paul McMahon
PEM Systems
Binghamton, NY, USA
pemcmahon@acm.org

Brian Elvesæter and
Arne J. Berre
SINTEF. Oslo, Norway
(brian.elvesater, arne.j.berre)@sintef.no

Michael Striewe and
Michael Goedicke
University of Duisburg-Essen
Essen, Germany
(michael.striewe, michael.goedicke)@ paluno.uni-due.de

Shihong Huang
Dept. of Computer & Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL USA
shihong@fau.edu

Bruce MacIsaac
RMC Method Content
IBM
San Jose, CA USA
bmacisaa@us.ibm.com

Ed Seymour
Business & Applications Services
Fujitsu
22 Baker Street
London, W1U 3BW
ed.seymour@uk.fujitsu.com

***Abstract*—The new software engineering initiative, Semat, is in the process of developing a kernel for software engineering that stands on a solid theoretical basis. So far, it has suggested a set of kernel elements for software engineering and basic language constructs for defining the elements and their usage. This paper describes a session during which Semat results and status will be presented. The presentation will be followed by a discussion panel.**

***Keywords*-software engineering method; kernel; theoretical basis; Semat; OMG standard; practitioner**

I. Introduction

Even if software engineering has existed for more than 50 years, the software engineering community has not yet come to a common agreement on a set of concepts, rules and principles that are widely accepted as a basis for explaining and predicting various software engineering phenomena. Although much work has been done, such as for instance [8], to define a consistent conceptual and theoretical basis for software engineering has been and still is a challenging task. Software engineering is enormously broad and its endeavors are strongly context-dependent. By supporting development and maintenance of software systems in varying domains, it includes many methods and method variants with differences little understood and often artificially magnified. Software engineering is also strongly dominated by a huge split between the industrial practice and academic research.

Possessing many years of history, the software engineering community is now mature enough to define a kernel for software engineering that stands on a solid theoretical basis. All the code that is being written, the application types being built, the solutions being designed, the methods being employed, or the organizations being involved are now possible to explain with concepts and principles that are common to all software engineering endeavors. For this reason, it is high time the software engineering community gathered its forces and defined a conceptual and theoretical basis for software engineering.

In 2009, a new initiative - **Software Engineering Method And Theory (Semat) -** was launched with the goal of refounding software engineering as a rigorous discipline [3, 4, 7]. Semat is an open community group consisting of volunteers who share a common motivation and dedication towards refounding software engineering. As a first step, the group has created a kernel for software engineering methods standing on a multitude of disciplines such as computer science, sociology, psychology, empirical science [4].

Semat strives to make the software community successful in building software better and faster and make users and customers happier. Although Semat addresses all roles in software engineering, its main target group are practitioners. Semat's utmost objective is to make the practitioners successful as individuals and as partners. This is because *the practitioners and the industry drive what needs to be taught and researched. The academics teach and formalize what needs to be better understood. The practitioners are the doers in the industry and the industry leads the practitioners in creating business value.*

In this paper, we describe a session during which Semat and its preliminary results will be presented and discussed. The remainder of this paper is as follows. Section II briefly presents Semat and its status. Section III provides the session agenda. Finally, Section IV suggests future steps.

II. Semat and its Status

Semat was launched by Ivar Jacobson, Bertrand Meyer and Richard Soley. Those three wrote a *Call for Action* pinpointing lack of theoretical basis for software engineering and raising concerns and issues that challenged the field [4]. The call was quickly welcomed by the software community worldwide. So far, it has been signed by 36 thought leaders in the software engineering field, 17 large corporations and universities such as IBM, Microsoft, Ericsson, Chalmers, KTH, Huawei, SEI, and by close to 1700 supporters around

978-1-4673-1067-3/12/$31.00 © 2012 IEEE

ICSE 2012, Zurich, Switzerland
Invited Summary

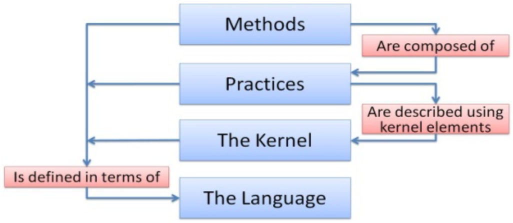

Figure 1. Semat Method Architecture

the world. Two chapters of Semat are set up in China and Latin America and more are on the way.

The *Call for Action* included a *Grand Vision* describing a solution standing on a sound conceptual and theoretical basis. The solution revolves around a *kernel* including essential elements and characteristics that are common to all software engineering methods [4]. The kernel elements provide the essence of software engineering. Their choice is guided by the notion that, "*You have achieved perfection not when there is nothing left to add, but when there is nothing left to take away*" [1].

Establishing the kernel is crucial. The kernel is designed to (1) provide software practitioners with a tool needed to better understand, compose and compare individual practices and methods, (2) aid companies in realizing a consistent, identifiable framework for governance, whilst allowing their developers the freedom to use their preferred practices, (3) constitute a learning roadmap to help form new curricula and personal development goals, and, finally, (4) support research by providing context and agreed subjects of value. Moreover, the kernel is expected to reduce the fashions and fads prevalent in the software industry today, and usher in a more pragmatic and objective era. As illustrated in Fig. 1, Semat suggests that a method is a composition of practices that are described in terms of the essential kernel elements. In addition, Semat suggests a domain-specific language to be used for defining the kernel elements and their usage.

The work of Semat has been going on for almost two and a half years. Presently, Semat has three ongoing tracks. These are (1) *kernel track* developing the essential elements in the kernel, (2) *language track* creating the language, and (3) *practice track* evaluating the usability of the proposed kernel and the domain-specific language in the context of existing engineering practices. Semat has also a fourth theory track that is dormant right now. However, once Semat has agreed on the kernel and the language, the theory track will play a significant role in the work going forward.

So far, the kernel track has defined essential elements such as *Work, Way of Working, Team, Stakeholders, Requirements, Software System,* and *Opportunity*. The language track has defined the main constructs such as *Alphas* representing the essential elements, *Activities*, *Activity Spaces* to be filled in with a set of activities to be conducted in a specific software engineering endeavor, *Work Products* and *Practices*. Finally, the practice track has applied the kernel and language when describing agile practices such as Scrum with the purpose of evaluating their applicability, precision, suitability, coverage and ease of use.

III. Semat Session

The "*Refounding Software Engineering*" session will include the following topics:

- *Introduction to Semat* presenting the Semat initiative, its vision, target group and value proposition.
- *Semat kernel* providing insight into the kernel and its elements and explaining how the kernel can help developers in tackling their daily challenges.
- *Semat language* introducing the main language constructs and motivating why Semat needs its own language.
- *Evaluation* illustrating the applicability, precision, suitability, coverage and ease of use of the Semat approach by applying it on Scrum practices.
- *Differentiators* comparing Semat to other initiatives and motivating how Semat can help software practitioners.
- *Three-year vision* presenting the three year vision of Semat and its fulfillment [5].

The Semat presentation is going to be followed by a panel discussion. The panel will enable the audience to interact with the Semat members and resolve potential misunderstandings and misconceptions.

IV. Future

At the moment of writing this paper, Semat has submitted a proposal for a new standard to OMG [2, 3, 6]. The proposal focuses on how to provide optimal support for software practitioners. It suggests how the practitioners can create and work with software methods in an agile manner irrespective of the methods they use. Semat has also released its results for the worldwide evaluation [7]. Everybody is very much welcome to provide feedback for improving and extending the Semat results.

References

[1] A. de Saint-Exupéry, Terre des Hommes, Reynal & Hitchcock, 1939.

[2] Essence – Kernel and Language for Software Engineering" A proposal submitted to OMG RFP, OMG Document, 2011.

[3] Foundation for the Agile Creation and Enactment of Software Engineering Methods (FACESEM) RFP (OMG Document ad/2011-06-26.

[4] I. Jacobson, B. Meyer, and R. Soley. The SEMAT Initiative: *A Call for Action*. Dr. Dobb's Journal, vol. 10 Dec. 2009.

[5] I. Jacobson, S. Huang, M. Kajko-Mattsson, P. McMahon, E. Seymour, "Semat – Three Year Vision", Programming and Computer Software, vol. 38, Jan. 2012, pp. 1-12, doi: 10.1134/S0361768812010021.

[6] OMG RFP "A Foundation for the Agile Creation and Enactment of Software Engineering Methods", online at http://www.semat.org/pub/Main/WebHome/Foundation_for_SE_Methods_RFP_ad-11-06-26.pdf, retrieved on March 20, 2012.

[7] Software Engineering Method and Theory (Semat), www.semat.org, retrieved on March 20, 2012.

[8] C. Ghezzi, M. Jazayeri, D. Mandrioli, Fundamentals of Software Engineering, Prentice Hall, 2nd Edition, 2002.