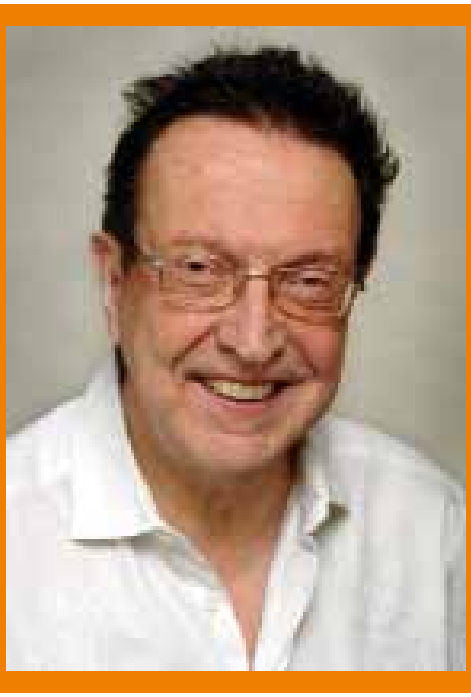
Ivar Jacobson
Chairman, Ivar Jacobson International (IJI)

Discover the Essence of Software Engineering



Object-oriented methodology is a well established technique in modern software development paradigm. However, looking back, with software engineering evolutions, we face some fundamental problems and concerns regarding development methods and process. A Global initiative Semat– Software Engineering Method and Theory tries to address the problems we face as a community. Semat is a community effort that attempts to discover the essence of software engineering.

The software industry, which is perhaps the largest sector in the world, has become extremely successful. However, software engineering as a discipline is still challenged by similar problems as it was at its beginning.

Object-oriented methodology is a very important and very influential software development paradigm. Thus, no doubt it has a place in our history.

In this article, however, I would like to take one step back, and look at software engineering as a whole – from its inception about fifty years ago until nowadays – to discuss how we as a community are doing, what the fundamental problems are that we are facing, what the shared essentials of the different development methods are, and how we could discover and capture these essentials to make software engineering a discipline that is built on solid theory, proven principles and best practices. These questions and others eventually led to the formation of the global initiative Semat – Software Engineering Method and Theory.

1. The problems we face as a software engineering community

More than four decades ago, a meeting held in Garmisch in Germany gathered software leaders of that time to define the term "software engineering" and to lay out its future. Despite the previous effort in formalizing software development, it was not until the Garmisch meeting that the problems of software development were formally discussed.

Over the last forty years, software has ubiquitously infiltrated into every aspects of our society and people's life. The software industry, which is perhaps the largest sector in the world, has become extremely successful. However, software engineering as a discipline is still challenged by similar problems as it was at its beginning. As a community we have really not come much closer to a good understanding of what software engineering is. We have developed and adopted a whole series of techniques, such as structured programming, structured analysis and design, object-oriented programming, object-oriented design, component development, UML,

Every adoption of a new method makes the old one obsolete. It is like throwing out the baby with bathing water. This is not the smart way of doing our work.

Unified Process, CMMI, XP, Scrum, and now Lean and Kanban, just to name a few.

There are no doubts, that as our discipline moves forward, more and more new methodologies continue to appear. The software industry to some extent is like the fashion industry in the way that we keep chasing fads or silver bullets, but keep ignoring the basic, fundamental things in our discipline. Over these years we have developed hundreds of thousands of methods, since basically every product development organization creates its own way of working. This is no surprise and this is the way it should be. The abundant number in itself is not the problem, but the fact that it is virtually impossible to compare the way different teams work (even within a single organization) is a serious problem. The knowledge gained from these methods is not preserved from system to system, from generation to generation. It is neither transferrable nor reusable among systems, among practitioners, and among organizations. Every adoption of a new method makes the old one obsolete. It is like throwing out the baby with bathing water. This is not the smart way of doing our work.

Furthermore, academia is in a constant catching up mode of chasing what is new. There is not a consistent and fundamental teaching platform across different instructors and different universities. Research topics and education curricula at universities are generally so remote from what the industry wants and needs. As a consequence, the research results mainly stay as research lab "orphan", are difficult to be adopted by industry; students who graduate from universities have a hard time to adapt themselves to the challenging industrial environment. The result is both researchers and educators are not happy because they see their hard work is buried at its birth, and not as fruitful as it should be.

2. Semat – discovering the essence of software engineering

Against the backdrop of the problems we face today, the Semat community, www.semat.org, was founded in 2009 by Bertrand Meyer, Richard Soley and I. Ahead of the very first Semat meeting held in March 2010 in Zurich, late Watts Humprey suggested: "This (SEMAT) meeting in Zurich (2010) is likely to be an historic occasion much like the 1968 NATO session in Garmish."

Semat addresses the many challenges we face today in the software engineering field. In essence, the major challenge is to understand how to build great software, and why we need a theory for software engineering [1][2][3]. We suggested that we needed 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". To be effective the kernel must be kept concrete, focused and small [4].

Semat's primary goal "is to create a kernel and a language that are scalable, extensible, and easy to use, and that allow people to describe the essentials of their existing and future methods and practices so that they can be composed, compared, evaluated, tailored, used, adapted, simulated and measured by practitioners as well as taught and researched by academics and researchers [5]".

Semat addresses the "human" side of software development as well as the technical side because after all, it is people who develop software, not methods and tools.

Some Semat key concepts include:

1. Every method is just a composition of practices, either human- or technology-related.
2. Practices are reusable over many application areas and technologies.
3. Underneath all practices is a small kernel of universals, things that we always have, do or produce when building software.
4. Practices and universals are described by a lightweight and intuitive language.

Ahead of the very first Semat meeting held in March 2010 in Zurich, late Watts Humprey suggested: "This (SEMAT) meeting in Zurich (2010) is likely to be an historic occasion much like the 1968 NATO session in Garmish."

Research topics and education curricula at universities are generally so remote from what the industry wants and needs. As a consequence, the research results mainly stay as research lab "orphan", are difficult to be adopted by industry; students who graduate from universities have a hard time to adapt themselves to the challenging industrial environment.

The kernel we are looking for is the common ground or the essence of software engineering. With such a common ground, including some 10-20 key elements, we will have a light but powerful vocabulary to describe any method. This will make it easier to compare and evaluate methods.

For more detailed information, please go to www.semat.org and read our publications to have a better understanding.

We have high expectations that Semat will change the software engineering arena. The basis for the change is the understanding that underneath all methods is a kernel.

3. Semat is a community effort

Semat results are a collective effort from the community for the community. Recently, in order to provide the necessary governance of the work on developing the kernel, the responsibility for this work has

been moved to the Object Management Group (OMG, http://www.omg.org/). This move to OMG ensures the openness and fairness of the selection process and that the results benefit the entire community.

A Request for Proposal (RFP) has been developed by a couple of people from Semat and a couple of OMG members. The RFP is called "A Foundation for the Agile Creation and Enactment of Software Engineering Methods". This RFP was presented in an OMG meeting in March 2011, Washington D.C. The next time when an updated RFP proposal will be presented will be in June 2011 in Salt Lake City.

Semat addresses the "human" side of software development as well as the technical side because after all, it is people who develop software, not methods and tools.

A group of about 20 people within Semat has been working on a candidate kernel since March 2010. This group includes members coming from around the world representing practitioners, executives, professors and researchers. Thus far, there are four universals being proposed – team, work, software system, and requirements. There will be more developed as the work goes on.

Moving forward we need competent people to actively participate in the different task groups. We need people with the following expertise: user experience design to give the language a graphical, intuitive syntax; formal language designers to make sure the concrete syntax is mapped to meaningful semantics; identifying and defining kernel elements (modeling expertise); metrics and measurement experts to help measure the impact of Semat on the external world and to help measure each of its practices; open source tool support for language and kernel; requirement specification of what Semat should do, and more.

Semat needs a broader community involvement to make its results more relevant to practitioners, academic and industry. Please go to Semat blog site: http://sematblog.wordpress.com/, to give comments and feedback. Your comments and feedback are crucial to keep Semat in the right direction.

Welcome to become part of the Semat community.

We suggested that we needed 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".

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://www.semat.org/pub/Main/WebHome/SEMAT-vision.pdf

[5] Ivar Jacobson, Shihong Huang, Mira Kajko-Mattsson, Paul McMahon, Ed Seymour, "Semat – Three Year Vision" in the Proceedings of the Spring/Summer Young Researchers' Colloquium on Software Engineering by Institute for Systems Programming, Russia Academy of Sciences (SYRCoSE 2011: May 12-13, 2011, Yekateringburg, Russia). ■

About the Author

Dr. Ivar Jacobson is a father of components and component architecture, use cases, the Unified Modelling Language and the Rational Unified Process. He has contributed to modern business modelling and aspect-oriented software development.

However, all this is history. Lately he has been working on how to deal with methods and tools in a smart, superlight and agile way. He has developed a practice concept that is now being adopted by both developers and tool vendors. Now he is one of the leaders of a worldwide network Semat, which has agreed to revolutionize software development.

Dr. Jacobson is an international honorary advisor at Peking University, Beijing, and he holds an honorary doctor at San Martin de Porres University, Peru.

He is the principal author of six influential and best-selling books.

Ivar Jacobson is the chairman of Ivar Jacobson International which has subsidiaries in the US, UK, China, Singapore, Sweden and Canada.