



**Presents**  
**An IT Metrics and Productivity Journal Special Edition**

**Focus on Dr. Hans Sassenburg, SEI Europe**  
**A CAI State of the Practice Interview**  
**November, 2006**

**Biography of Dr. Hans Sassenburg**

Hans Sassenburg received a Master of Science degree in electrical engineering from the Eindhoven University of Technology (Netherlands) in 1986 and a PhD degree in economics from the University of Groningen (Netherlands) in 2006. He worked as an independent consultant until 1996, when he co-founded a consulting and training firm. In 2001 he moved to Switzerland, where he founded the new consulting firm SE-CURE AG. In addition, he has been a visiting scientist at the Software Engineering Institute since January of 2005. In 2006 he co-founded the joint venture Software Improvement Group AG, offering a portfolio to create management transparency with respect to the quality of software applications. The Software Improvement Group supports its services and products with automated source code analysis. Our interview between Hans Sassenburg and Michael Milutis, Executive Director of the IT Metrics and Productivity Institute, took place in November of 2006.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

**CAI: Could you tell us a little bit about yourself, your early career, and how you got to be where you are today?**

**HANS SASSENBURG:** I received my master's degree in Electrical Engineering from the Eindhoven University of Technology in 1986. In 1983, however, I had already started working as a freelancer in the software industry. I continued as a freelancer till 1996, initially working as a programmer, later becoming a

software architect, project leader, and SEPG leader. In the early nineties I learned about the Software CMM, which I regarded as a major breakthrough in providing evidence that the software industry was (and still is) very immature. I also started to be active in the area of measurement and analysis using the standard works from Boehm ("Software Engineering Economics", 1981) and Putnam and Myers ("Measures for Excellence", 1992). I still think that both references are very useful, but unfortunately not widely enough read.

In 1996, I decided to co-found a training and consulting firm specializing in Software Process Improvement and Software Architecture. Through my assignment as an associate professor at the Eindhoven University in a post-masters program, I was able to recruit very professional and skilled employees. At the same time, I co-founded and chaired SPIDER, the Dutch SPIN. Both of these groups turned out to be very successful.

In 2000, the Software Improvement company, at that time employing 25 professionals, was sold. In the meantime, SPIDER had grown to a network with more than 750 members with quarterly events, various special interest groups, and annual conferences. In 2001, I founded a new consulting firm, but lacked the motivation to work on acquiring business. In my view, despite the enormous investments in process improvement, the business problems were increasing exponentially due to the growing size and complexity of projects and the increasing pressure to shorten time-to-market. I decided to write a book about my experiences as a kind of farewell. The book uses the building of a house in the Swiss Alps as a metaphor to demonstrate the immaturity in the software industry and our failure to learn from other disciplines. The book's success brought me to a turning point in my career. Instead of leaving the industry, I decided to try to make a contribution to the discipline by starting a PhD project. Over the next three years, I studied the phenomenon of software release decisions, a key issue for software manufacturers. During this research, I combined three different disciplines: economics, software management, and

group behavior (or social psychology). It was an exciting time, confirming my opinion that the software industry can learn a lot from other disciplines which offer a wealth of information and validated research results.

By early 2005 I had joined the SEI. This has allowed me to meet very skilled, experienced and motivated people who are continually researching new ideas and piloting them in a practical setting.

After my PhD was finished in early 2006, I continued to work for the SEI. I am currently still a member of the SEMA team, active in the area of measurement and analysis. My assignments include bringing ideas and courses to European universities and setting up joint research projects with European industry.

I am continually approached by investors who are interested in co-founding new companies. One of my projects this past year has resulted in the creation of a joint venture with a Dutch firm of a Software Improvement Group in Switzerland. It is a spin-off of the Dutch Centre for Mathematics and Informatics. Our specialization is in the automated analysis of source code and our portfolio consists of high-level consulting services and a licensed toolset. My current role in the firm is to build up the company and to help customize its portfolio for Swiss and German industry, which are very focused on the financial and automotive sectors, respectively.

**CAI: In your PhD, you approached the topic of software release decisions from an economic, a software management and a social psychological perspective? Why?**

**HANS SASSENBURG:** This PhD project was conducted in the Faculty of Economics of the University of Groningen (Netherlands). Consequently, it was clear that I had to include an economic perspective. Besides, what is a release decision anyway but a market entry decision where the objective is to

maximize profits and where the inputs into your release decision are expected cash inflows and outflows?

The market entry decision poses some specific problems. When should a product be released? What is the market window? What are the expectations of customers and end-users? How do you verify the correct implementation of functional and non-functional requirements? How much testing is needed?

The social psychological perspective also demands our attention. The decision to release a product will normally involve different stakeholders, who will not necessarily have the same goals. This means elements of individual and group conflict are likely to surface when discussing a release decision.

In my research project each of these different perspectives were reviewed and explored in depth, both from a theoretical and from an empirical point of view, through the study of practical examples. Ultimately, I wound up with a method that would enable software manufacturers to evaluate and compare different release alternatives. The results can be used to improve strategic software release decisions, especially those in which large financial loss is possible.

However, the root problem in the software industry is the lack of complete and reliable information as input. Consequently, I also proposed several measures to find a higher optimum ratio between the value of decision making information and the corresponding costs of such information.

### **CAI: What were your final conclusions?**

**HANS SASSENBURG:** A series of case studies confirmed that software manufacturers struggle with determining values for quality attributes like reliability and maintainability of their products, especially during testing prior to a release decision, during the design phase and in post implementation. As a result, the estimates for post-release operational costs for short-term

corrective activities and long-term product enhancements, prior to release decisions, is a very difficult task. Such difficulties lead to a greater use of non-analytical methods for deciding when a software product is “good enough” for release. Non analytical methods on their own, however, are not sufficient or appropriate for software release decisions, especially in cases where large prospective financial loss outcomes to a software manufacturer and its customers are possible.

**CAI: The Standish Group came out with a study several years ago that showed that 70% of all software projects were coming in over budget, over schedule, or not at all. Why do you think these numbers are so bad?**

**HANS SASSENBURG:** First of all, I do not know enough about the data set used for this analysis to formulate an opinion about the conclusions. Was it a representative sample over different market segments? Was the quality of the data verified in some way? Were all contributors reporting these data in a consistent way? These are the kinds of questions that we would need to ask before making any final judgments about the report.

Secondly, I do not believe that differences between planned and actual budgets/schedules are necessarily the right indicators for determining how well we are performing. Some organizations state aggressive project goals in order to ensure that the project team will not get lazy while in other cases the requirements will not be entirely known at project start and will grow and change as a project progresses.

What we can say with certainty, however, is that most organizations fail to learn from the past. Most organizations also ignore the existence of powerful tools that are available for making realistic estimates and, in turn, higher project success rates. Post-mortem analysis of finished projects is one example

of a technique that is not made enough use of in the software industry. There is simply no time available. We are usually already well underway with the next project, typically with the same irrational level of optimism regarding estimates, even though no changes to the way we work have been made.

There are many rules of thumb and sophisticated tools that are available for deriving more solid estimates for budget and schedule. One caveat, though, is that tools like COCOMO II and SLIM will not give you predictions at project start. A project must progress and uncertainty levels must decrease before reliable estimates can be obtained with these tools.

Let me offer a recommendation for reducing budget and decreasing lead time that will cost nothing. In the past, I worked for an organization where we decided to implement a logging mechanism in the code to monitor how often certain functionality codes were being used. We found that over 50% of the implemented modules were never executed. By ignoring or removing this code, maintenance costs were highly reduced. Less code takes less time and fewer hours and one gets a system that is easier and cheaper to maintain. It is as simple as that.

**CAI: What are some of the most well known and effective software process improvement methods in use today? How would you differentiate them from each other? How can organizations determine which approach is best for them?**

**HANS SASSENBURG:** Organizations worldwide are turning to an ever-increasing set of international standards and models in their effort to improve their processes in general. Nevertheless, defining and implementing process changes can be both expensive and time consuming. As organizations in multi-model improvement environments attempt to coordinate the standards they

have chosen (for business reasons) and those they have had mandated to them (by regulators), they will encounter a common set of barriers, chief among these being the apparent dissimilarity of various models across projects and groups and the desire of each of these groups to champion their own preferred standard or model.

At the SEI, we believe that to succeed, organizations need to think about process improvement differently. A first step is to recognize that, despite the different structures and terminologies, despite different levels of abstraction in the standards and models, there are many common element types. The challenge is to identify the common elements.

For example, there are always good practice elements. These are model elements that define what an organization needs to improve in any particular area. The specific practices of the CMMI are an example of good practice elements. Other models and standards include TSP, PSP, ITIL, Cobit, etc.

You will also find a rich choice of improvement methods. These are methods that help the organization master the change management process, communicate the vision of the future state, plan and execute the resulting improvement initiatives, and manage them long term. DMAIC and IDEAL are good examples of improvement methods.

Finally, we can distinguish institutionalization elements. The Generic Goals and Practices of the CMMI are examples of promoting institutionalization.

**CAI: What are the major challenges that most organizations encounter when they first get started with software metrics?**

**HANS SASSENBURG:** By now one can point to many instances where measurement has been used effectively to inform management and technical

decision makers in the development and maintenance of software. Nevertheless, measurement is still not very well integrated into software or systems engineering education or practice. Furthermore, measurement still remains challenging for far too many organizations.

At the SEI, we stress that the ultimate benefit of software measurement comes from the decisions and actions that are taken in response to analysis of the data, not from the collection of the data itself. As Dave Zubrow, team leader of the SEMA group at the SEI, puts it: "Software measurement activities are like potential and kinetic energy; gathering the data creates a potential, but it takes analysis and action to make it kinetic." The SEI's Goal-Driven Software Measurement approach identifies ten steps to establish a measurement program so that, in the end, the program is aligned with the organization's business processes. In this manner, the risk of gathering data that doesn't get used is minimized.

**CAI: So much is being made of offshore these days. What is the importance of metrics and process in an offshore relationship? What should organizations be doing to get this right?**

**HANS SASSENBURG:** The primary impetus for offshore outsourcing is cheap labor and comparatively higher quality. The negative side effects revolve around the lack of direct client interaction and the lack of premium customer support.

In the end, though, I personally think that the case for outsourcing is very questionable for many organizations. The old justification, "They program while you sleep," just doesn't always hold. Too often, a problem will arise and the offshore people won't respond until the next day. Their response will be that they didn't know how the customer wanted them to proceed. Consequently, a

whole day will be lost. Problems like this are ultimately rooted in our inability to properly baseline and analyze our requirements up front. If we could improve our capabilities in this area, the case for outsourcing would be more attractive.

I would recommend that organizations first make a proper business case for their outsourcing efforts. There should be an overall business case at the senior management level that describes the expected long-term benefit and cost areas. From these benefit and cost areas, metrics can then be derived. It is important to monitor standard metrics on offshore projects; specifically, metrics for schedule, budget, functionality, and quality, not to mention the additional measurements that are related to your overall business objectives. The Goal Driven Software Measurement approach, with its 10 steps for establishing a measurement program that is aligned to an organization's business processes, can be very useful here.

**CAI: Aside from your own books, are there any other specific books on software metrics that you might recommend?**

**HANS SASSENBURG:** I mentioned already the standard works "Software Engineering Economics" (1981) from Boehm and "Measures for Excellence" (1992) from Putnam and Myers. Other works I recommended are "Software Cost Estimation with COCOMO II" (2001), from Boehm et al., "Software Metrics: A Rigorous Approach" (1997) from Fenton and Pfleeger, and any of the many books published by Capers Jones. In addition, I would recommend books on micro economics ("Fundamentals of Managerial Economics", Hirschey, M. 2003) and decision-making ("The Managerial Decision-Making Process, Harrison, E.F., 1987).

A great online resource that I can heartily recommend is the Software Engineering Information Repository, otherwise known as the SEIR

([HTTPS://SEIR.SEI.CMU.EDU/SEIR](https://seir.sei.cmu.edu/seir)). The purpose of SEIR is to provide a forum for the exchange of process improvement and metrics related information. There is much valuable information out here. The SEIR also offers visitors the possibility of making contributions to their online community of researchers.

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity Journal Editor at [michael\\_milutis@compaid.com](mailto:michael_milutis@compaid.com).