Software Process Simulation — At a Crossroads?

He Zhang[*1], David Raffo[2], Thomas Birkhölzer[3], Dan Houston[4], Raymond Madachy[5], Jürgen Münch[6], Stanley M. Sutton Jr.[7]

[1] *State Key Laboratory of Novel Software Technology, Software Institute, Nanjing University, Nanjing, Jiangsu, China*
[2] *School of Business Administration, Portland State University, Portland, OR, USA*
[3] *University of Applied Sciences, Konstanz, Germany*
[4] *Computer & Software Division, The Aerospace Corporation, El Segundo, CA, USA*
[5] *Department of Systems Engineering, Naval Postgraduate School, Monterey, CA, USA*
[6] *Department of Computer Science, University of Helsinki, Helsinki, Finland*
[7] *IBM T.J. Watson Research Center, Yorktown Heights, NY, USA*

SUMMARY

Software Process Simulation (SPS) has been evolving over the past two decades after being introduced to the software engineering community in the 1980s. At that time the SPS technology attracted a great deal of interest from both academics and practitioners in the software process community – even to the extent of being one of the recommended techniques for achieving the quantitative modeling KPA for CMMI Level 4. However, in recent years, the growth of SPS seems to have slowed along with the number of reported applications in industry. This article summarizes the special panel during ICSSP 2012 whose goal was to assess whether this technology remains applicable to today's software engineering projects and challenges and point out the most beneficial opportunities for future research and industry application. Copyright © 2014 Press

1. BACKGROUND AND MOTIVATIONS

The idea and the pioneering work of simulating software processes can be dated back to the early 1980s when Abdel-Hamid and Madnick's seminal work in System Dynamics modeling of a large scale software project [1, 2]. Raffo and Kellner's research followed and focused more deeply on the software process using the State-Based and Discrete Event Simulation paradigm [3, 4]. Scacchi's work among others modeled the software development process down to the level of a process script that could be enacted by individual process actors [5]. Other researchers followed in the 1990s and early to mid 2000s. Hybrid models were also developed which often combined the best of Discrete Event and System Dynamics paradigms [6]. Agent-Based models were also developed and applied to software projects [7]. Many useful models and results were published through the PROSIM[†] international conference series and the special issues in the *Journal of Systems and Software* and the journal of *Software Process: Improvement and Practice*.

From reviewing this body of work, it can be seen that SPS models were developed to address a variety of software lifecycle models including *waterfall*, *incremental*, *spiral development*, and *product line development* among others [8, 9]. Different project scopes were also modeled from

---

[*]Correspondence to: State Key Laboratory of Novel Software Technology, Software Institute, Nanjing University, 22 Hankou Rd. Nanjing, Jiangsu 210093, China
[†]International Workshop on Software Process Simulation Modeling

Prepared using **smrauth.cls** *[Version: 2012/07/12 v2.10]*

specific lifecycle phases, to complete projects, as well as distributed development projects and multiple concurrent projects. Many of these articles documented savings and benefits made possible with the use of SPS models and showed the applicability of SPS to support process planning, managing, controlling, and improvement [10, 11].

As a results of the early promise, research in SPS arena has been an important focus of the software process community, particularly since the PROSIM workshops from the late 1990s. The recent retrospective impact analysis of SPS [11] found a noticeable number of studies reported the success of the practical adoption of SPS in software industry. However, compared to other software process practices, for example the agile methods, inspection, or cost estimation, the acceptance and adoption of SPS in software practice have been more limited.

The questions before the software process community which we plan to address in this paper are:

- *What are the benefits to applying SPS today? Is SPS still a useful and usable technology for the software industry?*

- *What areas are promising for new research in the SPS area? What issues is SPS uniquely capable of addressing?*

- *What obstacles exist to widespread adoption of SPS and how can these obstacles be addressed?*

2. THE SPECIAL PANEL

At ICSSP 2012, a special panel session was organized with the aim to address the above questions. A number of leading experts in the Software Process Simulation field from both academia and industry participated in the panel. They were:

- Thomas Birkhölzer, University of Applied Sciences, Konstanz, Germany

- Dan Houston, Computer & Software Division, The Aerospace Corporation, El Segundo, CA, USA

- Raymond Madachy, Department of Systems Engineering, Naval Postgraduate School, Monterey, CA, USA

- Jürgen Münch, Department of Computer Science, University of Helsinki, Helsinki, Finland

- David Raffo, School of Business Administration, Portland State University, Portland, OR, USA

- Stanley M. Sutton Jr., IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

He Zhang[‡] from National ICT Australia moderated the panel. The panelists presented their position papers on the topic of SPS from different perspectives. Their positions are summarized in this article.

3. THE BENEFITS TO PRACTITIONERS AND INDUSTRY

The expectations for SPS have been high for those who have become involved with it. In [12], Kellner, Madachy, and Raffo described the possible SPS benefits for *strategic management*, *planning*, *control and operational management*, *process improvement and technology adoption*, *understanding*, *training and learning*. However, such a vision has yet to be realized widely. Based

---

[‡]He is now with the Software Institute, Nanjing University, China

on recent evidence in the community, the panelists indicated SPS remains a much-needed field of endeavor but it needs to be matured. Although it can be very valuable in addressing many problems, it is overkill for some problems and inadequate for others. Our challenge is to demonstrate the utility of SPS to our colleagues and show our customers what kinds of questions are answered best through modeling and simulation.

In order to better understand the benefits of simulation, organizations also need to be aware of the cost of 'not simulating'. An analogy and comparison between simulation in software process and in other disciplines was made. In the software development field, in practice, SPS is often perceived as an additional cost to a software project. However, in other fields of engineering, simulation is seen much more as a *cost saver* than as a *cost driver*. On the contrary, it is often a mandatory prerequisite before testing and implementing any real world versions of systems. Therefore, process engineering work and research should consider the consequences of waiving simulation, e.g., neglecting a verification and validation process during a development project prior to its enactment.

Like any simulation, an SPS model is an abstraction (i.e. simplification). The level of expectation for exactness (with respect to details and parameter values) should be set in the context of purpose and claimed results. Obtaining *reasonable results* (even, if not exact or proven) using a simplified model and parameter values is better than not simulating at all. Trustworthy, quantitative predictions are the final step of simulation maturity, which might still be beyond the horizon in many areas of software engineering.

4. FUTURE DIRECTIONS FOR PROCESS SIMULATION RESEARCH

Broadening Process Perspectives

In order to increase the adoption and leverage the impact of process simulation, process modelers need to take broader and more integrated perspectives, and extend their efforts. Apart from the areas pretty well explored already, there are still many software specific issues, which require further modeling research. The *uncharted territory* of dynamic process modeling requires challenges and critiques not just from model developers but also from prospective simulation consumers and critics as well to point to missing or desirable aspects.

Software has been increasingly embedded in everyday life and is much more complex than ever before. A holistic approach to developing complex systems will help software engineers better understand the overall context of where their critical software resides and thus create more effective systems. In most circumstances the software process is one of several processes (including systems engineering, hardware development and business processes) to be integrated in an enterprise that involves not just engineers and technicians but also people in non-technical roles. To make greater impact, process modelers should involve other disciplines and considerations in broader, more holistic models. In the community, [13] presents some recent examples of models with a broadened scope that represent a shift from a focus on single project/system dynamics towards multiple, interacting projects for systems of systems and product lines.

In actual practice, software process is never as static as a blue print and constantly evolves and varies over time. In contrast, models of emerging and evolving software processes, such as global development processes, agile methods, and process changes led by new technologies, are still scarce in SPS community in spite of their popularity in practice. Simulation of these vibrant trends may make a better use of software process simulation in the future.

Strengthening Pragmatic Support

The development effort required for the most complex software process models may reach up to a few person years. Comprehensive and complex models (of software, processes, or systems) can only be developed and maintained when based on a solid and rich foundation of mature, established, and proven components. The same is true for process simulation models. The build-up of such a library of simulation components requires a significant amount of time, research, and development beside prior or independent concrete applications. Such an endeavor can only be realized by a collaborative

effort. The software process community should appreciate and encourage research and groundwork efforts toward a common knowledge base and library building of simulation components.

Similar to business and scientific research in general, analytics are as important as (if not more important than) modeling and simulation in software process research in particular. Modeling, simulation, and analytics can and should be closely intertwined in practice — "*three sides of the same coin*".

Moreover, any complex model will require interplay, among multiple simulation paradigms. Advanced tools should be developed and chosen, which at least enable and support hybrid, modular, and integrated models and component libraries.

5. APPLYING PROCESS SIMULATION IN SOFTWARE PRACTICE

Working Closely with Practitioners

The previous success of SPS applications in practice highly relied on the collaboration between academia and industry [11]. This kind of collaboration appears to have been and continues to be more critical than in other areas of software engineering. One possible reason is that both simulation model building and the application lessons have not yet been made easily accessible to industry. To address this issue, perhaps the most effective and efficient approach is for researchers to take critical roles in enabling SPS technology transfer. This will create a culture and constitution that encourages and facilitates close collaborations between researchers and practitioners and will go a long way to speeding SPS adoption.

Our challenge also comes from the demonstration of the utility and benefits of SPS. Wide communication of SPS benefits will be essential for successful SPS deployment. The SPS research community must become leaders in educating developers and development managers in SPS and in embedding SPS knowledge and skills in existing processes and tools. Also we must learn how to model and provide answers quickly and structure a modeling engagement in response to a specific question.

The acceptance and willingness of creating or using software process simulation models in organizations requires that practitioners or decision makers see a sufficient value. However, it is not sufficiently clear who is willing to invest in software process simulation and why. With the goal of better understanding the value of SPS, "*Validated Learning*" [14] is suggested to conduct experiments with potential '*customers*' to test value hypotheses, i.e. "*tests whether a product or service really delivers value to customers once they are using it*" [14]. A simulator would not be developed unless a potential customer sees a value in such results. It is also important to be able to demonstrate the value of simulators with respect to higher level goals of an organization up to business goals. One approach to aligning software process simulation with higher level goals of an organization in a seamless and quantifiable way is to define simulation goals using GQM+Strategies [15].

To make a difference in practice, practitioners can be a source of problems, motivations, data, resources, and validation. Working with practitioners can be very rewarding in and of itself. However, researchers should be prepared to have many challenges that can stand in the way of realizing the potential value of process simulation in practice, such as: practitioner availability, data issues, solution adoption and evaluation and so forth. These sorts of challenges can be extremely frustrating, but to make the biggest impact, researchers and project leads must work around or through them.

Are You Ready for Process Simulation?

Having a descriptive process model that reflects the *real* practices in organizations is in most situations an essential prerequisite for creating process simulation models. [16] demonstrates a recent example how a complex software process can be simulated based on its descriptive model. In terms of the authors' experiences, however, there is often a gap between the *intended* (*planned*), the *real* (*enacted*), and the *perceived* software process in organizations. Getting descriptive models

that reflect the real company practices requires that firms have an organized approach to managing and controlling the definition, adherence, and evolution of software-related processes.

Another prerequisite is that companies are aware of their information needs. Although such visibility seems to be highly important, many companies do not know their information needs on the different levels of the organization sufficiently. Without an analysis of the information needs, the definition of dependent variables of a simulation model cannot be done in a value-creating way. In terms of the authors' experiences, most of the cost and duration of process simulation are given to elicitation of process information and data gathering and interpretation.

Successful industrial adoption of SPS requires four knowledge areas and skill sets: *modeling and simulation*, *analytics and statistics*, *managing engagements*, and *software development and maintenance* (i.e. domain knowledge and experience). This combination of knowledge and skills is difficult to find in industry. Many people are developed in one or two of these areas, but relatively few have developed three or all four of them. In the short term, academicians can address this hurdle by guiding students and collaborators from industry toward cross-disciplinary work and interdepartmental studies.

6. FINAL REMARK

In the past two decades, Software Process Simulation has demonstrated its value to software practice. Nonetheless, this area is still maturing in software engineering field. The widespread adoption of SPS in software industry requires further value-oriented research as well as recognition of its benefits in practice, in which the collaboration between academicians and practitioners take a critical role.

The special panel is one important step towards the restimulation of the SPS related research and practice and the reunion of the community. Based on the panel discussion, a research agenda on SPS will be suggested and compiled by the panelists. We appreciate the importance of a joint effort on maturing process simulation in software engineering and systems engineering. Let us join the forces!

ACKNOWLEDGMENT

REFERENCES

1. Abdel-Hamid TK, Madnick SE. A model of software project management dynamics. *6th Computer Software and Application Conference (COMPSAC'82)*, IEEE, 1982.
2. Abdel-Hamid TK, Madnick SE. *Software Project Dynamics: An Integrated Approach*. Prentice Hall: Englewood Cliffs, N.J., 1991.
3. Raffo DM, Kellner MI. Using quantitative process modeling to forecast the impact of potential software process improvements. *10th International Forum on COCOMO and Software Cost Modeling*, Pittsburgh, Pennsylvania, 1995.
4. Raffo DM. Evaluating the impact of process improvements quantitatively using process modeling. *Proceedings of Conference of the Centre for Advanced Studies on Collaborative Research (CASCON'93), Volume 1–Software Engineering*, IBM: Toronto, Ontario, Canada, 1993; 290–313.
5. Mi P, Scacchi W. Modeling articulation work in software engineering processes. *Proceedings of the 1st International Conference on Software Process (ICSP'91)*, IEEE Computer Society: Washington, DC, USA, 1991; 188–201.
6. Martin RH, Raffo DM. Application of a hybrid process simulation model to a software development project. *Journal of Systems and Software* 2001; **59**(3):237–246.
7. Smith N, Capiluppi A, Ramil JF. Agent-based simulation of open source evolution. *Software Process: Improvement and Practice* 2006; **11**(4):423–434.

8. Zhang H, Kitchenham B, Pfahl D. Reflections on 10 years of software process simulation modelling: A systematic review. *International Conference on Software Process (ICSP'08)*, vol. LNCS 5007, Springer: Leipzig, Germany, 2008; 345–365.

9. Zhang H, Kitchenham B, Pfahl D. Software process simulation modeling: An extended systematic review. *International Conference on Software Process (ICSP'10)*, vol. LNCS 6195, Springer: Paderborn, Germany, 2010; 309–320.

10. Raffo D, Wakeland W. High value added ways to apply process simulation within organizations. *Technical Report CMU/SEI-2008-TR-002*, Software Engineering Institute, Carnegie Mellon University January 2008.

11. Zhang H, Jeffery R, Houston D, Huang L, Zhu L. Impact of process simulation on software practice: An initial report. *33rd International Conference on Software Engineering (ICSE'11)*, ACM: Honolulu, HI, USA, 2011; 1046–1056.

12. Kellner MI, Madachy RJ, Raffo DM. Software process simulation modeling: Why? what? how? *Journal of Systems and Software* 1999; **46**(2/3):91–105.

13. Madachy R. *Software Process Dynamics*. IEEE Press, 2008.

14. Ries E. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, 2011.

15. Basili VR, Heidrich J, Lindvall M, Münch J, Regardie M, Rombach D, Seaman C, Trendowicz A. Linking software development and business strategy through measurement. *Computer* 2010; **43**(4):57–65.

16. Zhang H, Klein G, Staples M, Andronick J, Zhu L, Kolanski R. Simulation modeling of a large-scale formal verification process. *International Conference on Software and Systems Process (ICSSP'12)*, IEEE: Zurich, Switzerland, 2012; 3–12.