Requirements for Software Process Modeling Tools

Rodrigo M. Pagliares

Computer Science Department – Federal University of Minas Gerais (UFMG)
Av. Antônio Carlos, 6627 – CEP 31.270-010 – Belo Horizonte – MG - Brazil

pagliares@gmail.com

Abstract. *To be done.....*

1. Introduction

Software Development is all about process. In the last years, several software processes were developed and used [Boehm 1996], [Krutchen 2003], [Jacobson+ 1999], [Humphrey 1995], [Schwaber 2004], [Beck 2004]. A portion of these tried to impose some form of rigor in the way which software is conceived, documented, developed and tested [Pfleeger and Atlee 2005]. The other ones were conceived, in the late 1990s, by some developers who had resisted to this rigor and formulated their own principles, trying to highlight the roles that flexibility could play in producing software quickly and capably. They codified their thinking in "the agile manifesto" [Agile 2009] that focuses on alternative ways of thinking about software development.

Several process and software process definitions can be found on literature. Accordingly to IEEE, a process is a sequence of steps performed for a given purpose [IEEE 1990]. More specifically, Ian Sommerville defines a software process as a set of activities whose goal is the development or evolution of software [Sommerville 2006]. Roger S. Pressman states that software process is a series of predictable steps – a roadmap to be followed helping the creation of timely and high quality results [Pressman 2001].

A software process can be modeled in natural language, by graphical descriptions or using software process meta-models [Nardini+ 2008], [Combemale+ 2006]. Henderson-Sellers [Henderson+ 2004] depicts various benefits of software process modeling:

- ensure a consistent, reproducible approach to all projects providing an uniform proposal to software development with guidance;
- control and support the whole development life with good project planning and management;
- allow consistency and traceability though the whole life cycle;
- emphasize analysis and understanding through customer involvement;
- reduce risk associated with shortcuts and mistakes;
- produce complete and consistent documentation from one project to the next;
- gain flexibility to process tailoring and managing and supporting different types of projects.

Since the 80s, there have been various process meta-models and finally OMG (Object Management Group) decided to create a standard in this area. The specification of SPEM (Software & Systems Process Engineering Meta-Model), version 1.0, was

published in 2002. In 2005, SPEM 1.1 was available with minor updates. SPEM 1.1 still had several shortcomings; for example, semantics were ambiguous and hard to understand and there was a lack of enactment support. SPEM 2.0 was released in 2007 fixing the flaws in previous versions, being compliant with UML 2 and providing guidance on migrating existing process models from SPEM 1.1 to SPEM 2.0 [OMG 2008].

SPEM is a standardized way of expressing any software development process. The specification was developed especially to address the unique and complex nature of software development. It is vendor, framework and methodology neutral and leverages the expressiveness and popularity of UML (Unified Modeling Language) [OMG 2009]. The SPEM 2.0 is a meta-model developed by OMG, described through a meta-modeling language (meta-meta-model) [OMG 2006], [OMG 2008].

With so many benefits and a standardized language for process modeling, ones would argue that the direct use of SPEM for modeling process would be a natural choice. However, as stated by Wilson de Pádua Paula Filho, the SPEM 2.0 metamodel is of huge complexity and hard to be used directly by process developers [Paula Filho 2009]. To ease this complexity, a process modeling tool becomes necessary. Until now, few of these were developed but, with the advent of SPEM 2.0, is my belief that a plethora of them will be available in a near future.

Several people would benefit from using or constructing these tools: process designers could more easily describe and improve the existing company´s software development process; software development team members would have, for instance, some level of consistency and shared language across the organization; software engineering instructors could use these tools to teach current best practices in their curriculums and also bring the state-of-the-art software best practices to the software industry.

As so many people would benefit from software process modeling tools, a catalog with requirements of such ones would be of great help, aiding team members, instructors, project managers and process designers to compare different implementations on the market and helping tools providers to direct future developments. Matthias Hoffmann, Nikolaus Kuh, Matthias Weber and Margot Bittner presented a catalog of requirements for requirements management tools in the area of automotive as well as aircraft and defense systems [Hoffmann+ 2004]. The objective of this work is to provide a similar catalog in structure but with focus on software modeling tools instead.

The remainder of this paper is organized as follows: in section 2, we present software modeling tools models and features. Section 3 describes the requirements for such tools. Practical and operational aspects are presented in Section 4. Section 5 lists some tools, already on market, and how well they adhere to the requirements presented at Section 3. Finally, in Section 5, conclusions are made.

References

Agile (2009) “Manifesto for Agile Software Development”, http://agilemanifesto.org , Last access on march, 2009.

Bandinelli, S., Fuggetta, A., Lavazza, L., Loi, M. and Pietro Picco, G. (2005) "Modeling and Improving an Industrial Software Process" In: IEEE Transactions on Software Engineering, vol. 21, issue. 5, 1995, Pages: 440-454.

Beck, K. (2004), Extreme Programming Explained: Embrace the Change, 2nd Edition. Addison-Wesley.

Bendraou, R., Combemale, B., Cregut, X. and Gervais, M. P. (2007) "Definition of an Executable SPEM 2.0", In: Proceedings of the 14th Asia-Pacific Software Engineering Conference. IEEE Computer Society, Pages: 390-397.

Boehm, B. (1996) "Anchoring the Software Process", In: IEEE Software 13(4), Pages: 73-82

Borsoi, B. T. and Becerra, J. L. R. (2008) "The Use of Object Orientation to Define Process Models", In: Proceedings of the Sixth international Conference on Software Engineering Research, Management and Applications. IEEE Computer Society, Pages: 85-92.

CMMI (2006) "CMMI for Development, Version 1.2.", CMU/SEI-2006-TR-008. Software Engineering Institute.

Combemale, B., Crégut, X., Caplain, A. and Coulette, B. (2006) "Towards a rigorous process modeling with SPEM", In: 8th International Conference on Enterprise Information Systems: Databases and Information Systems. Pages: 530-533.

Debnath, N., Riesco, D., Cota, M.P., Garcia Perez-Schofield, J.B. and Uva, D.R.M. (2006) **"**Supporting the SPEM with a UML Extended Workflow Metamodel", In: IEEE International Conference on Computer Systems and Applications. Pages: 1151-1154.

ECMA (1993) "TR/55 Reference Model for Frameworks of Software Engineering Environments"

ECMA (1994) "TR/69 Reference Model for Project Support Environments"

EPF - Eclipse Process Framework (2009), http://www.eclipse.org/epf/ . Last access on march, 2009.

Gonzalez-Perez, C. (2007), "Supporting Situational Method Engineering with ISO/IEC 24744", In: IFIP WG8.1 Working Conference on Situational Method Engineering: Fundamentals and Experiences (ME07), Geneva, Switzerland. Pages: 7-18.

Gonzalez-Perez, C. and Henderson-Sellers, B. (2007) "Modelling software development methodologies: A conceptual foundation", In: Journal of Systems and Software. Volume 80, Issue 11. Pages: 1778-1796.

Henderson-Sellers, B., Serour, M., McBride, T., Gonzalez-Perez C. and Dagher L. (2004) "Process Construction and Customization", In: Journal of Universal Computer Science, vol. 10, no. 4, Pages: 326-358.

Henderson-Sellers, B., Gonzalez-Perez, C. and Ralyté, J. (2008), "Comparison of Method Chunks and Method Fragments for Situational Method Engineering", In: Proceedings of the 19th Australian Conference on Software Engineering. Pages: 479-488.

Hoffmann, M., Kuhn, N., Weber, M. and Bittner, M. (2004) "Requirements for Requirements Management Tools", In: Proceedings of the 12th IEEE International Requirements Engineering Conference. Pages: 301-308.

Humphrey, W. S. (1995), A discipline for Software Engineering. Addison-Wesley.

Humphrey, W. S. (1999), Introduction to the Team Software Process. Addison-Wesley

Hsueh, N., Shen, W., Yang, Z., and Yang, D. (2008) "Applying UML and software simulation for process definition, verification, and validation", In: Journal of Information and Software Technology. Volume 50, Issue 9-10. Pages: 897-911.

IEEE (1990) "IEEE Standard Glossary of Software Engineering Terminology", IEEE std.610.12-1990. IEEE Computer Society, 1990.

IEEE (1995) "IEEE Std 1348-1995, "Recommended Practice for the Adoption of CASE Tools", *(ISO/IEC 14471),* IEEE Press, 1995.

IEEE (2004) "IEEE Std 1517-1999, IEEE Standard for Information Technology-Software Life Cycle Processes-Reuse Processes", IEEE Press, 2004.

ISO/IEC - International Organization for Standardization/International Electrotechnical Commission (2007) "ISO/IEC 24744 - Software Engineering Metamodel for Development Methodologies".

ISO/IEC (2001) "ISO/IEC 9126-1:2001: Software engineering - Product quality -Part 1: Quality model".

Jacobson, I., Rumbaugh, J. and Booch, G. (1999), The Unified Software Development Process, Addison-Wesley.

Järvi A., Mäkilä T., Hakonen H. (2006) "Changing Role of SPI – Opportunities and Challenges of Process Modeling", In: Proceedings of the 13th European Conference on Software Process improvement (EuroSPI 2006).

Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J. and Linkman, S. (2009) "Systematic literature reviews in software engineering - A systematic literature review", In: Journal of Information and Software Technology. Vol. 51, Issue 1, Pages: 7-15.

Krutchen, P. (2003), The Rational Unified Process – An Introduction, 2nd Edition. Addison-Wesley.

Nardini, E., Molesini, A., Omicini, A. and Denti, E. (2008) "SPEM on test: the SODA case study", In : Proceedings of the 2008 ACM Symposium on Applied Computing. Pages: 700-706

OMG (2006) "Meta Object Facility (MOF) Core Specification", v2.0. Formal/2006-01-01, http://www.omg.org/docs/formal/06-01-01.pdf .

OMG (2008) "Software & Systems Process Engineering Meta-Model Specification", v2.0. Formal/2008-04-01, http://www.omg.org/docs/formal/08-04-01.pdf.

OMG (2009) "Unified Modeling Language: Superstructure", Version 2.2. Formal /2009-02-02, http://www.omg.org/docs/formal/09-02-02.pdf

Park, S., Choi, K., Yoon, K. and Bae, D. (2007) "Deriving software process simulation model from spem-based software process model", In: Proceedings of the 14th Asia-Pacific Software Engineering Conference (APSEC 2007*)*, Pages: 382–389.

Park, S., Kim, H., Kang, D. and Bae, Doo-Hwan (2008) "Developing a software process simulation model using SPEM and analytical models", In**:** International Journal of Simulation and Process Modeling. Vol. 4, No.3/4. Pages: 223 - 236.

Paula Filho, W. P. (2009), Engenharia de Software: Fundamentos, Métodos e Padrões, LTC, 3th edition. Rio de Janeiro, Brazil. Chapter 5, Page:108 (in Portuguese).

Pfleeger, S.L. and Atlee, J. M. (2005), Software Engineering: Theory and Practice, Prentice Hall, 3th edition. Page: 59.

Pressman, R. S. (2001), Software Engineering: A practitioner´s Approach, McGraw-Hill, 5th edition. Page: 19.

Schwaber, K. (2004), Agile Project Management with Scrum, Microsoft Press.

Sommerville, I. (2006), Software Engineering, Addison-Wesley, 8th edition. Page: 6.

Wu, M., Li, G. and Yan, J. Y. H. (2006) "A Metamodel Approach to Software Process Modeling Based on UML Extension", In: IEEE International Conference on System, Man and Cybernetics. Volume: 6, Pages: 4508-4512

Yu, E. S. K. and Mylopolous, J. (1994) "Understanding 'Why' in Software Process Modeling, Analysis, and Design" In: 16th International Conference on Software Engineering.

Zhang, H., Kitchenham, B. and Pfhal, D. (2008) "Software Process Simulation Modeling: Facts, Trends and Directions", In: Proceedings of the 2008 15th Asia-Pacific Software Engineering Conference. Pages: 59-66.

Zhang, R. and Zhang, B. (2008) "Integrated Evaluation Model for Software Process Modeling Methods", In: Proceedings of the international Conference on internet Computing in Science and Engineering. IEEE Computer Society, Pages: 355-358.