Software Process Definition for Multi-Organizational Development in the Aerospace Domain

Fabio Bella[1],
Dirk Hamann[1],
Masa Katahira[2],
Yuko Miyamoto[2],
Jürgen Münch[1],
Makoto Shizunaga[2]

[1] Fraunhofer Institute for Experimental Software Engineering (IESE)
Fraunhofer Platz 1,
67663 Kaiserslautern, Germany
{bella, hamann, muench}@iese. fraunhofer.de
+49 (0)631 6800-2133

[2] Japan Aerospace Exploration Agency (JAXA)
2-1-1 Sengen
Tsukuba-Ibaraki, 305-8505, Japan
{Shizunaga.Makoto, Miyamoto.Yuko}@jaxa.jp, katahira@computer.org

Abstract: Many projects in challenging domains such as the aerospace domain turn out to be of particular criticality due to the collaboration efforts required between different organizations and teams involved in very different disciplines. Enabling smooth collaboration of multi-organizational, multi-disciplinary teams is a key for the successful development of complex software-dependent systems. Defining suitable collaboration procedures for each new project ad-hoc can be a costly and risky undertaking. Organizations performing more than one single project together should consider the idea of integrating their processes in a more consistent way by addressing and integrating both organization-specific process models and process terminology. In this paper, we present initial results and experience from process integration conducted at the Japan Aerospace Exploration Agency (JAXA). The development processes of different departments are being integrated with the processes of the prime suppliers. In particular, we focus on eliciting the requirements for development process descriptions and infrastructure from different stakeholders' viewpoints as part of the scoping of an appropriate multi-organizational software process management infrastructure. The methodology is illustrated by a concrete case study performed at the Japan Aerospace Exploration Agency (JAXA). In addition, selected lessons learned and consequences for process management support in the aerospace domain are sketched.

Keywords: software process modeling, process integration, global development

1. INTRODUCTION

Large system engineering efforts in the aerospace domain are typically conducted in a multi-organizational structure with different types of organizations such as space agencies, contractors, subcontractors, and organizations in charge of independent quality assurance. Due to the diversity of domains, processes, process standards, and national cultures, integrating contributions from different organizations and disciplines (such as system engineering, software, hardware, and economics) in a system engineering project is a very challenging task. To achieve meaningful collaboration, many organizations operating in this domain manage such projects by adapting their own processes and work products at the beginning of each project. For organizations performing more than one project together, this is a costly and risky way of collaboration, since the respective development processes must be synchronized and the consistency of the resulting multi-organizational process must be evaluated each time ad-hoc on the basis of the project specification.

In this article, we present an approach for integrating multi-organizational, multidisciplinary development processes. To achieve smooth process integration between partners that conduct more than one single joint project together, the approach proposed addresses collaboration at the process definition level rather than at the project specification level. This means that the processes of different organizations that are relevant for a specific type of projects are aligned once and applied without or only with minor changes within the context of each project of that type. The approach is currently being applied to integrate development processes at the Japan Aerospace Exploration Agency (JAXA) and its contractors. First results and lessons learned are presented.

2. RELATED WORK

Due to the complexity of the problem and the different aspects to be addressed, many of the challenges that arise when setting up a process infrastructure for distributed multi-organizational environments are addressed in different fields of research. Knowledge management in software organizations, for example, is a well-established research field that supports the extraction of knowledge from both quantitative and qualitative data gathered from software development projects. Basili et al. [1] have formalized an approach to knowledge management and continuous improvement in software organizations based on the idea of experience factories. An application of the approach within the context of a European research project can be found in [4]: The project was aimed at providing a sound knowledge base and, in particular, an integrated reference process for developing services for the wireless Internet.

The field of software process modeling [9] provides methods and techniques for eliciting, compiling, and validating process knowledge. Descriptive process modeling [3] plays a major role for integration at the process model level. Different levels of process integration are distinguished in [17] where integration at the project level, process integration, and alignment of process meta-models are discussed. Process integration based on process commonality analysis is addressed in [20]. Approaches to integrating partial, role–specific process models (e.g., views) into one comprehensive descriptive process model are discussed in [25], [26]. In these approaches, variations in the models are seen as possible inconsistencies that trigger a deeper analysis. The final goal is to obtain a multi-view-consistent comprehensive process model.

Process tailoring is closely related to the field of process modeling and addresses the adaptation of reference processes to the concrete situation of actual projects. Caivano et al. [7], Hesse et al. [15], and Münch [18] reformulate the problem of process tailoring in terms of reuse of process fragments and propose different approaches for composing project-specific development processes on the basis of validated reference process elements. Osterweil [19] highlights the similar nature of process assets and software artifacts. Hayashi et al. [14], Rombach [22], and Stanley et al. [24] start from this consideration to motivate and suggest the use of product line technologies [8] for process management and project-specific process tailoring. Bayer et al. [2] propose concrete adaptations of existing product line approaches for engineering process lines in the case of software products with a strong focus on business processes.

Global software development, on the other hand, is an interesting, emerging research field that addresses those organizational problems that usually arise when development is spread among different distributed teams. Berenbach [5] discusses the impact of organizational structure on development processes performed in distributed environments with particular attention to requirements engineering. Bikram et al. [6] propose a research agenda for the field and highlight the importance of both process and metric issues and the need for a deeper understanding of the dynamics of distributed software development. Grinter et al. [13] present different approaches for coping with multi-side coordination of work, one of which is based on the co-location of teams performing related process steps. Ramasubbu et al. [21] propose enhancements of the CMM reference process model to address distributed development more properly.

3. BACKGROUND

The Japan Aerospace Exploration Agency (JAXA) was founded on October 1, 2003 by merging the following three Japanese organizations into one independent administrative institution: a) the Institute of Space and Astronautical Science (ISAS), which was devoted to space and planetary research; b) the National Aerospace Laboratory of Japan (NAL), which focused on research and development of next-generation aviation; and c) the National Space Development Agency of Japan (NASDA), which was responsible for the development of large-size launch vehicles such as Japan's primary large-scale launch vehicle H-IIA, satellites, and the International Space Station (ISS). The consolidation of these three formerly independent organizations allows a continuous and systematic approach to space exploration, from basic research to development and practical application. Due to the consolidation, several issues arise, on the other hand, in terms of integration of the different development processes, the related proof of process conformity at project execution time, and the challenging coordination of distributed system engineering.

The focus of the case study described in this paper is on JAXA’s satellite development projects. JAXA’s satellites offer a wide variety of valuable services such as communications and weather observation, which are essential to modern life, as well as astronomical observation and space development.

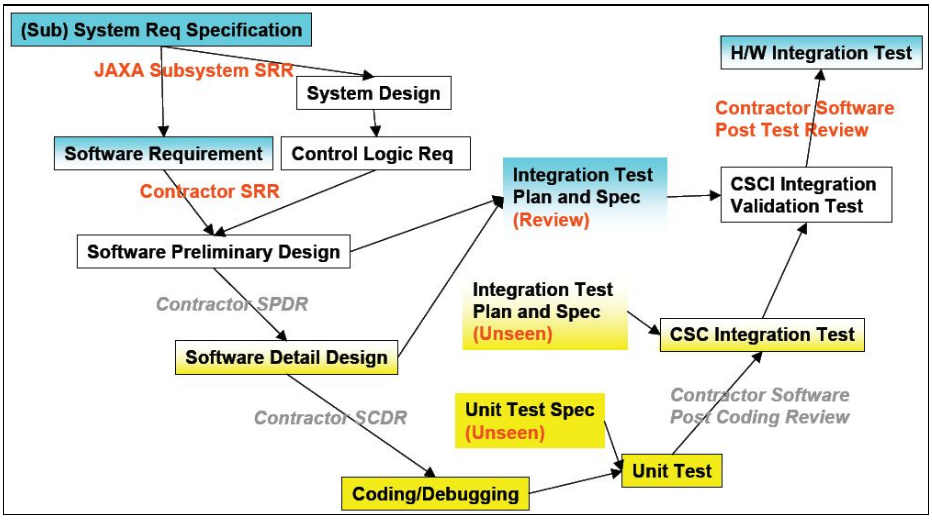

Figure 1. High-level description of JAXA’s processes

The typical satellite project structure can be described best when looking at Figure 1. JAXA itself is mainly responsible for the phases (Sub-)System Requirement Specification, SW Requirement, Integration Test Planning and Specification, and HW Integration Test. The prime contractors are mainly responsible for designing and further detailing the requirements (see white boxes in the figure). Subcontractors are then responsible for implementing and testing them. On the basis of this structure, the focus for the case study was set to the software requirements phase as the main interface in the early stage of a development project.

4. APPROACH FOLLOWED TO INTEGRATE PROCESSES AT JAXA

The approach followed consists of two main phases: process analysis and process (re)definition. The first phase focuses on analyzing process information from JAXA by systematically comparing it with selected international standards such as ISO/IEC 15504 [16] and the ECSS Series of the European Space Agency (ESA) (in particular [10], [11], [12]). Based on the results of the investigative study, preliminary recommendations for further enhancements and modifications of the existing processes and development practices were derived. The second phase aims at reorganizing JAXA’s development process and the related documentation and setting up an infrastructure for managing the overall development process.

The approach applied in the second phase is based on systematic evaluation and configuration of a process management infrastructure, process scoping, descriptive process modeling, process redefinition, and a continuous evaluation loop. The approach consists of six main steps to be performed iteratively:

1. *Determine Goals and Purpose of the Process Model* - Process management within a company starts by stating the objectives and scope of the process model. It is important to know what the process model will be used for (e.g., measurement, understanding), who will use the process model, and what the people involved in the process expect from it.
2. *Select Process Modeling Concepts* - The second step consists of selecting or developing a process modeling schema. This schema should identify and structure the information to be captured by the process model and

the relationships between these information entities. The concepts to be described by the process model depend on the objectives stated in the first step.

3. *Select/Tailor Notation and Tool Support* - The need for formal description of a software process has led to a variety of process modeling formalisms, so-called "process modeling languages". According to the domain they were originally intended for, software process modeling formalisms are tailored to solve specific problems and are based on different representations, with some of them using a graphical notation, some only textual representations. Depending on the process modeling concepts selected in the previous step, an appropriate modeling language has to be chosen or created. Supporting tools are selected or developed according to the chosen modeling language. The use of tools can be crucial for effective process modeling.
4. *Elicit Process Knowledge* - The goal of elicitation is to acquire all the information needed to describe the software process. The sources of information include interviews with the people involved, observation, and analysis of organizational documents and products.
5. *Create Process Model* - The process information gathered in the previous step serves as input when the actual model is created. The process participants are asked to review the process model. It is very likely that several cycles of creation, respectively modification and review, will take place until all process participants agree upon the process model.
6. *Analyze Process Model* - Process models are checked for completeness, correctness, and structural consistency (i.e., static analysis) and behavioral aspects, such as possible risks, critical paths, or cost overruns. The descriptive model can also be used to check for conformance during process performance. Of course, this step only makes sense if a descriptive model is used prescriptively or as a benchmark process.

The approach, based on previous studies by Becker et al. [3], is tailored for gathering and consolidating process-related information from different organizations. Compared to process definition for single organizations, major enhancements of the approach were needed to identify the main stakeholders and their needs with respect to process documentation and infrastructure, to stick to a process terminology that could be understood and used within every organization involved, and, finally, to establish suitable procedures for process management between the organizations involved (i.e., different JAXA departments, prime contractors, and organizations in charge of independent quality assurance). In particular, with respect to the first step *determine goals and purpose of the process model*, 33 typical scenarios that involve process models were discussed and prioritized with JAXA members and contractors. The scenarios were kept simple to allow their discussion with stakeholders with different background and from different organizations.

Table 1. Example of scenario for activities concerned with software process engineering

Role	Process Engineer (e.g., JAXA Safety and Mission Assurance Department)
Organizational context	Organization - JAXA
Activity	Process modeling: create semi-formal models of processes applied within JAXA and for the communication between Jaxa and contractors (i.e., JAXA standard process)
Purpose	Provide a consistent documentation of JAXA processes (e.g., for training and process analysis)
Point in time	Next step in JAXA roadmap to be performed as a one-shot action

Table 1 shows an example of a scenario. JAXA and contractors' members were asked to evaluate the relevance of each scenario from their perspective, to identify the documents to be produced / exchanged and the technical infrastructure required. With respect to the second step *select process modeling concepts*, the emerging standard Software Process Engineering Metamodel (SPEM, [23]) was introduced and tailored together with JAXA members and contractors to agree upon a common process terminology. Finally, the creation of the required process documentation was initiated by training JAXA personnel and analyzing both initial process models and the activities performed to model development processes and to ensure the quality of the resulting descriptions.

5. RESULTS

The process integration efforts are still ongoing: The first phase aimed at analyzing the process documentation available at JAXA has already been performed. The approach to process (re)definition was also introduced and fine-tuned together with JAXA and its prime suppliers and applied to the area of satellite development. The areas of ground system development and launch vehicle development will be addressed as next steps.

The main focus of the first integration activities was placed, on the one hand, on the required process infrastructure and, on the other hand, on the structure of the overall development process, with particular attention to those interim development results that represent process interfaces between JAXA and its suppliers.

The analysis of scenarios performed in step 1 led to the identification of three different process models required: one standard process model should include the high-level descriptions of the main process building blocks, i.e. those process elements that should be used to create project-specific process descriptions; project-specific process models will be the result of tailoring the standard process and will be created and applied within the context of single projects; finally, a more abstract reference process model should be used as process dimension within the scope of a domain-specific process assessment framework. The three different types of models are seen as different views on the same process, i.e., the same information is presented with different degrees of abstraction.

The different degrees of abstraction are also reflected by the results of the analysis performed in step 2 aimed at identifying the required process terminology. Different subsets of SPEM should be adopted for the different process views: project-specific process models will require, for instance, all SPEM concepts, whereas the concepts Lifecycle, Phase, and Iteration were not found to be needed for defining the standard process and the process dimension of the assessment framework.

On the basis of the analysis of the process terminology, tools were investigated in step 3 with respect to their suitability for supporting the process modeling task. The tool Spearmint® was chosen for the initial modeling due to its ease of use, the experience of the process engineers involved in the project, and the support provided for the set of concepts required for the standard process model.

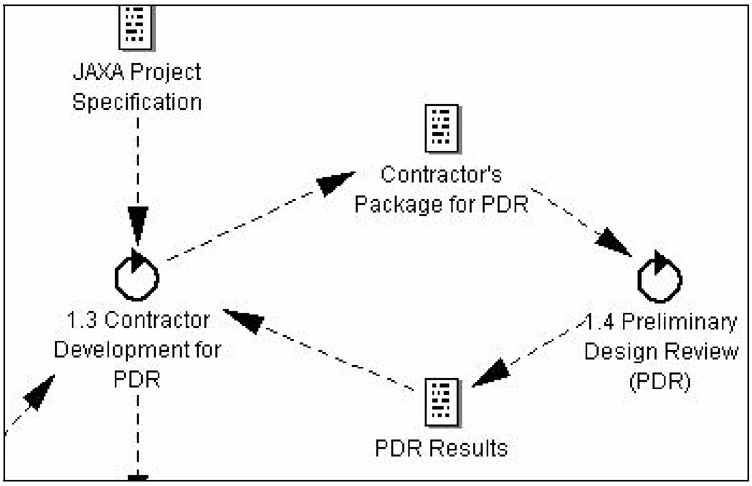

Figure 2. Excerpt of workflow included in the reference process

Further results include an initial exemplary process model produced in step 4, which was agreed upon by JAXA and suppliers' members. The initial process model focuses on the main documents / interim products exchanged among different organizations and, in particular, on the area of requirements engineering, and represents the basis for the standard process, which is still to be defined. Figure 2 shows an excerpt of the activities and products already modeled and their mutual relationships in the form of product flows.

6. DISCUSSION AND LESSONS LEARNED

Development processes are well understood at JAXA and the overall picture of the interactions between departments, on the one hand, and with the suppliers, on the other hand, is clear. For an integration of all development processes, the available process knowledge must be made explicit and available in the form of structured process descriptions that address at least all relevant development activities, products, roles, and their mutual relationships. Also, in order to be of value, the process descriptions must reflect the real processes.

To identify lessons learned to be taken into consideration for improving the approach in the next steps, retrospectives were held with the participation of JAXA members. The work with scenarios, the importance of SPEM, and the feedback mechanism currently implemented between the process engineers at the contractors' side and JAXA process engineers turned out to be important issues for the discussion.

Concerning the scenarios applied in step 1, their usage is regarded as appropriate. The approach allows aligning the aims of different classes of stakeholders such as contractors, developers, managers, and quality assurers, since the roles are explicitly addressed. This is particularly important in a multi-organizational context in which similar roles belonging to different organizations may have different aims. Some stakeholders have criticized the language used to describe the scenarios and advised that in their current form they can be difficult to understand without additional verbal explanations for people other than software process engineers. To address a broader audience of stakeholders, the descriptions should be reworked and made more understandable. Also, some scenarios should be merged, since they are closely related and less scenarios can leave more time for discussion.

Concerning SPEM, it is regarded as a good starting point for creating a process terminology that can be shared among different organizations. Some partners argued that the generality and the broad spectrum of the process language proposed is very suitable for aligning the different terminologies adopted within individual organizations. On the other hand, the complexity of SPEM should not be underestimated. In the future, much more effort should be spent on step 3 and more, deeper information should be provided to process engineers at the beginning of the step. A one- or even two-day training on SPEM instead of a brief introduction of about two hours is regarded as appropriate. Amount and quality of the process models formalized during the first iteration seem to be appropriate.

Also, the distinction between different process views, such as one JAXA standard process, many project-specific processes to be obtained as a result of tailoring the standard process, and a reference process to be adopted as process dimension of a domain-specific process assessment framework, is regarded as very important and useful.

Concerning process elicitation, as performed in step 4, and the feedback gathered from the contractors, interviews are seen as a suitable approach. However, contractors often need several days, sometimes even weeks, to answer specific questions offline. The response time is regarded as too long and the feedback cycles should be shortened. This could be achieved, for example, by involving contractors' process engineers in the process modeling activities performed at JAXA.

7. SUMMARY AND OUTLOOK

The integration of contributions from different organizations and disciplines in a complex system engineering project is a very challenging task due to the diversity of domains, processes, process standards, and national cultures. Partners that conduct more than one single joint project together would benefit from addressing the integration at the process definition level rather than at the project specification level.

In this article, an approach for integrating multi-organizational, multi-disciplinary development processes is presented. The approach extends existing process modeling approaches in order to mitigate problems deriving from different process terminologies applied in different organizations, misaligned expectations with respect to the use of process assets, and unclear requirements for the process infrastructure.

First experience collected shows that, as expected, typical problems that arise when eliciting and formalizing processes in individual organizations also arise in a multi-organizational context. They can be addressed through existing descriptive process modeling methods. Many of the additional problems that arise in such a context can be addressed through explicit agreement by all stakeholders upon purpose, scope, and terminology of the process model. To reach a meaningful agreement, a common understanding of the integration aims is required and can be achieved by means of intensive training sessions.

The many organizations and disciplines involved in projects in the aerospace domain lead to a great variety in terms of peculiar project characteristics with dramatic consequences in terms of process variability. The research community has already recognized the problem and suggests the use of product line technology for handling it; however, approaches to process line engineering for the specific aims of software and system engineering processes have not been proposed yet.

The case study described in this paper focuses on JAXA's satellite development projects. The areas of ground system development and launch vehicle development will be addressed next, which leads to increased complexity of the integration task. As a consequence, the investigation of suitable approaches for process line engineering becomes an unavoidable next step towards an infrastructure that best matches the requirements of software process management in this multi-organizational context.

Acknowledgements
We would like to thank the members of JAXA and their prime contractors involved in the satellite development projects investigated within the scope of the case study. The authors would also like to thank Sonnhild Namingha for reviewing the final version of the article.

References

[1] Basili, Victor R.; Caldiera, Gianluigi; Rombach, H. Dieter: Experience Factory. In: Marciniak, John J. (Ed.): Encyclopedia of Software Engineering. Volume 1. A-O.New York : John Wiley & Sons, 2002, 511-519 : Ill., Lit.

[2] Bayer, Joachim; Kose, Mathias; Ocampo, Alexis: Improving the Development of e-Business Systems by Introducing Process-Based Software Product Lines. In: Münch, Jürgen (Ed.) u.a.:7th International Conference on Product Focused Software Process Improvement. Profes'2006 - Proceedings. Berlin : Springer-Verlag, 2006, 348-361 : Ill., Lit. (Lecture Notes in Computer Science 4034).

[3] Becker-Kornstaedt, Ulrike; Belau, Wolfgang: Descriptive Process Modeling in an Industrial Environment. Experiences and Guidelines. In Software Process Technology. 7th European Workshop, EWSPT'2000 - Proceedings (2000), 176-189, Ill., Lit.

[4] Bella, Fabio; Münch, Jürgen; Ocampo, Alexis: Baselining Wireless Internet Service Development - An Experience Report. In: Brito e Abreu, Fernando (Ed.) u.a.: 5th Conference for Quality in Information and Communications Technology, QUATIC 2004 - Proceedings. Porto : Instituto Portugês da Qualidade, 2004, 161-169 : Ill., Lit.

[5] Berenbach, Brian: Impact of Organizational Structure on Distributed Requirements Engineering Processes: Lessons Learned. In Proceedings of the 2006 international workshop on Global software development for the practitioner 2006, GSD 2006. Shanghai, China, ACM Press, New York, NY, USA

[6] Bikram, Sengupta; Satish, Chandra; Vibha, Sinha: A Research Agenda for Distributed Software Development. In Proceeding of the 28th international conference on Software engineering 2006, Shanghai, China ICSE 2006. Shanghai, China, ACM Press, New York, NY, USA

[7] Caivano D., Visaggio C.A.: Process Diversity and how practitioners can manage it. Upgrade, the European Journal for the Informatics Professional, Vol. V, No.5, October 2004, CEPIS, pp.59-66.

[8] Coplien, J., Hoffman, D., Weiss, D.: Commonality and Variability in Software Engineering. IEEE Software, vol. 15, No. 16, pp. 37-45, Dec 1999.

[9] Curtis, B.; Kellner, M. I.; Over, J: Process modeling. Communication of the. ACM Volume 35, Number 9, 1992, 75-90.

[10] ECSS Q-80-B "Space product assurance - Software product assurance", 10 October 2003.

[11] ECSS-E-40 Part 1B Space Engineering: Software – Part 1: Principles and Requirements, 28 November 2003.

[12] ECSS-E-40 Part 2B Space Engineering: Software Engineering - Part 2: Document Requirements Definitions, Draft 8.

[13] Grinter, R., Herbsleb, J. and Perry, D. The geography of coordination: Dealing with distance in R&D work. International ACM SIGGROUP Conference on Supporting Group Work, 1999, pages 306-315

[14] Hayashi, Yoshikazu; Katahira Masa: Software Process Line – Application of Software Product Line to Software Process. In High Reliable Software Week, 2nd Advanced Information Technology Workshop (AITW), 5th Workshop of Critical Software (WOCS), Tokyo, 2005, 153-155, Lit.

[15] Hesse W., Noack J.: A Multi-variant Approach to Software Process Modelling. In Proceedings of the 11th International Conference on Advanced Information Systems Engineering, CAiSE'99, Heidelberg, Germany, June 14-18, Springer, 1999, (Lecture Notes in Computer Science 1626)

[16] ISO/IEC 15504, Information Technology - Software Process Assessment - Parts 1-5. International Standard, International Organization for Standardization (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland, 2006.

[17] Münch, Jürgen: Effective Process Integration for Space-related Software Development. In High Reliable Software Week, 2nd Advanced Information Technology Workshop (AITW), 5th Workshop of Critical Software (WOCS), Tokyo, 2005, 147-152, Lit.

[18] Münch, Jürgen: Transformation-based Creation of Custom-tailored Software Process Models. In: Institution of Electrical Engineers (IEE):ProSim'04. The 5th International Workshop on Software Process Simulation and Modeling. Herts, 2004, 50-56 : Ill., Lit.

[19] Osterweil, L. J.: Software processes are software too, revisited: an invited talk on the most influential paper of ICSE 9. In Proceedings of the 19th international Conference on Software Engineering (Boston, Massachusetts, United States, May 17 - 23, 1997). ICSE '97. ACM Press, New York, NY, 1997, 540-548.

[20] Ocampo, A., Bella, F., Münch, J.: Software Process Commonality Analysis, International Journal on Software Process: Improvement and Practice, 2005

[21] Ramasubbu, N., Krishnan, M.S., and Kompalli, P. Leveraging Global Resources: A Process Maturity Framework for Managing Distributed Development. IEEE Software, Volume 22, Issue 3, pages 80-86, May 2005

[22] Rombach, H. Dieter: Integrated Software Process and Product Lines. In: Li, Mingshu (Ed.) u.a.: Unifying the Software Process Spectrum. International Software Process Workshop, SPW 2005 - Revised Selected Papers.Berlin : Springer-Verlag, 2005, 83-90 : Lit. (Lecture Notes in Computer Science 3840).

[23] Software Process Engineering Metamodel Specification, January 2005, Version 1.1, formal/05-01-06, an adopted specification of OMG Group Inc. Available at http://www.omg.org/technology/documents/formal/spem.htm

[24] Stanley M. Sutton Jr and Leon J. Osterweil. Product families and process families. In 10th International Software Process Workshop (ISPW'96, Proceedings), pages 109--111, June 1996.

[25] Turgeon, J., Madhavhji, H., N.: A Systematic, View-Based Approach to Eliciting Process Models. In proceedings European Workshop on Software Process Technology, pp. 276-282, 1996.

[26] Verlage, M.: An Approach for Capturing Large Software Development Processes by Integration of Views Modeled Independently. Proceedings of the Tenth International Conference on Software Engineering and Knowledge Engineering SEKE, 1998.