Effective Process Integration for Space-related Software Development

Jürgen Münch

Fraunhofer Institute for Experimental Software Engineering (Fh IESE)
Processes and Measurement Department
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
Juergen.Muench@iese.fraunhofer.de

Abstract

Due to the increasing complexity of system engineering and market constraints, space organizations developing software and systems are forced to define, deploy, and evolve highly efficient and integrated processes that address aspects such as distributed development in multi-organizational, multi-disciplinary, and sometimes multi-national settings. Often, such organizations are confronted with a lack of standardized and harmonized interfaces that would allow smooth integration of different types of processes (such as development and validation processes as well as processes across organizational or cultural boundaries). In addition, such organizations typically have to adhere to several regulatory constraints and standards with their processes. Due to the lack of harmonization between domains, processes, process standards, and nations, the need to integrate contributions from different disciplines (system engineering, software, hardware, economics) in a system development project makes collaboration even more difficult. The effectiveness of such processes is therefore significantly reduced. Usually, organizations collaborate today by adapting their own processes and work products project by project. This is a costly way of collaboration. This article describes how to effectively integrate processes for multi-organizational projects. Based on a description of fundamental process modeling approaches, different integration levels for collaboration are described and a roadmap is shown on how to smoothly improve this integration. In addition, advanced integration support with respect to process variability is sketched.

1. Introduction

Software and system development efforts increasingly depend on multi-organizational and multi-discipline collaboration. Contributions from development sites, units, and teams with diverse backgrounds, such as different disciplines, languages, countries, cultures, organizational cultures, development approaches, and work processes need to be harmonized in an efficient way. In addition, legal and organizational boundaries from subcontracting relationships aggravate the collaboration. One major challenge is to enable effective and efficient collaboration of multi-disciplinary and multi-company teams to develop software-dependent systems (such as the space systems).

In their efforts, the different cooperating entities (such as contractors and subcontractors) need to adhere to different regulatory constraints and standards, for instance the level 2-standards ECSS E-40B or ECSS-Q80B for European Ground Segment Space Applications. In addition, companies usually have their own, company-specific processes and procedures that are partially derived from country-specific constraints (e.g., English companies might use a company-specific model that is compliant to PRINCE2).

Hence, multi-organizational development projects are confronted with problems and productivity losses because the development process standards and regulatory constraints are not aligned, the interfaces are not clear and explicit, and the underlying concepts of the standards are not compatible and clear. One important consequence is that in many cooperation projects, significant efforts are necessary at the beginning to define integrated procedures and processes for cooperation from scratch every time. This reduces the effectiveness gains that stem from distributing work to specialized development units.

For setting up a common process for collaboration, often, an ad hoc combination of the existing procedures, processes, and standards is chosen. This, however, has several disadvantages:

- Missing methodological support: A systematic method for the integration guaranteeing the correctness, consistency, accuracy, etc. of the resulting process is missing. This leads to inappropriate and suboptimal processes. In addition, it is not ensured that the best suited process artifacts will be selected.

- Missing adjustment: The customization to application domains, company cultures, legal aspects, etc. is usually not incorporated appropriately.

- Increased cost and effort: The ad hoc definition of process integration procedures results in increased time-to-market and high costs. This is also worsened by the fact that developers need to get familiar with the new processes every time and cannot profit from a learning curve.

Besides the process, several other technologies and mechanisms support the integration of multi-organizational and multi-disciplined work. Examples are computer-supported cooperative work tools (i.e., CSCW [Grundin et al., 1996]), communication infrastructures, and CASE environments that support distributed development. The process can be seen as a fundamental basis for coordinating collaborative work. All other integration approaches can be included in a process. One primary aim for organizations involved in space-related software development should therefore be to look for solutions to improve collaboration on the basis of efficiently integrated processes.

2. Process Integration Levels

In analogy to the integration of software components, where different integration types and mechanisms exist (e.g., data coupling, control flow coupling, common interface definitions), different levels of process integration exist. Besides the ad-hoc cooperation level that is characteristic for organizations operating without formalized procedures and defined interfaces, the following levels can be defined:

Process Integration Level 1: Process Instance Level. On this level, organizations typically operate multi-organization cooperation projects mainly in the context of project management activities. This includes, for instance, arrangements with respect to produced and consumed products between cooperation partners, control flows, milestones, quality gates, synchronization points, or activities on the “process instance” level. On this level, project participants typically do not have insights into the processes of their partners. Typical organizations on this level are SMEs (i.e., small and medium-sized enterprises) and contracting organizations (e.g., public authorities, automotive OEMs). Potential benefits of the cooperation mechanisms on this level are: 1) project risk reduction by mutual understanding of interfaces, 2) good suitability for organizations that start multi-organization projects, 3) support for better negotiation, 4) support for better project controlling, 5) harmonization on the level of project management processes, 6) reduced communication overhead, 7) better adherence to the project objectives.

Process Integration Level 2: Process Model Level. On this level, organizations typically operate multi-organization cooperation projects by harmonizing their engineering processes (such as requirements engineering, testing). This level reveals the relevant internal working processes to the project partner for better overall project optimization, e.g., in iterative or interweaved processes for software-based systems development. Achieving this level requires that sufficient cooperation mechanisms on the process instance level are already implemented. Typical organizations on this level are medium and large organizations that repeatedly cooperate in multi-company consortia; especially organizations that already have (or are forced to have) a defined process (e.g., organizations that have to comply to CMMI level 3 and higher, space standards, FDA regulations, process governance). Potential benefits of the cooperation mechanisms are: 1) harmonization on the level of project engineering/technical processes (e.g., design), 2) better controlling of subcontractors based on process performance indicators, 3) possibility to demonstrate compliance to process standards, 4) more effective coupling of engineering/technical processes, 5) more reliable tailoring of processes with respect to process standards, 6) possible process improvement of the overall development process. In addition, the benefits of process instance level cooperation mechanisms apply here.

Process Integration Level 3: Meta Model Level. On this level, organizations typically describe and manage their processes for cooperation by harmonizing their process architectures, process notations, and the process management approach (see also [Verlage and Münch, 1997]). This means that the meta model [Goldmann et al., 1999] for the process description (such as SPEM) as well as meta processes (such as process evolution) are harmonized. This level requires that cooperation mechanisms on the project and process levels are already implemented. Typical organizations on this level are large-scale industry organizations or long-term established cooperation consortia. Benefits of the cooperation mechanisms are: 1) effective management of process engineering across organizations, 2) more effective tailoring mechanisms, 3) more effective documentation mechanisms, 4) facilitation of more effective compliance to standards, 5) higher process automation by using joint process engines or workflow management tools. In addition, the benefits of process instance and process model level cooperation mechanisms apply here.

3. Process Integration Roadmap

Based on experience from the development of several multi-organizational process definition programs the following steps can be recommended for improving process integration:

1. *Process investigation or process assessment:* Process assessments can be used in order to identify process areas that have weaknesses or are not defined. Typically, best practice collections are used as benchmarks. For instance, ISO/IEC 15504 or its specialization for the space domain could be used.
   Output: Assessment results including a list of recommendations for process improvement.

2. *Process management scoping:* The area of process management activities should be defined and clear goals for these activities should be defined. As an example, the area could be the interface processes between contractors and subcontractors and the goal could be to reduce rework effort caused by communication problems at these interfaces.

Depending on the current degree of process integration, the following steps can be performed subsequently:

3. *Integration on the process instance level:* A process for planning and negotiation of cooperation in multi-disciplinary and multi-company teams (based on harmonizing key work products at the interface between stakeholders, shared terminology, and synchronization approaches) should be established.

4. *Process management organization:* An organizational unit (such as a software engineering process group) needs to be institutionalized, which is responsible for all process management activities (such as process evolution, provision of tailoring guidelines, process documentation, organization of process trainings). Process deployment can fail especially if an organization does not put enough emphasis on the promotion of process models and the infrastructure needed for process deployment (see also [Sheard, 2003]). Early results from a multi-case study conducted at Nokia clearly show the importance of a stable and well implemented infrastructure for process deployment [Raffo et al., 1999].

5. *Integration on the process model level:* An approach for mapping process models in multi-disciplinary and multi-company projects for repeatable cooperation in subsequent projects with much less lead time and effort losses for initial process harmonization should be established. This includes activities such as descriptive process modeling [Becker et al., 1997], process model analysis, and process coupling.

6. *Integration on the meta model level:* A meta model that allows organizations to effectively customize and integrate processes by harmonizing the process notations and process management procedures should be established.

7. *Continuous improvement:* The processes should be analyzed during execution and potential improvement options should be identified. Well-established approaches supporting continuous improvement are

Goal-Oriented Measurement (with GQM [Basili and Weiss, 1984]) and the Quality Improvement Paradigm (QIP) [Basili et al., 1994]. A feedback loop needs to use the findings for process optimization.

Depending on the maturity of an organization, step 3 is often already in place. However, the integration is usually not explicitly described and several activities need to be done on this level. Typically, organizations already have processes in place that cannot easily be changed. Therefore, we recommend conducting step 4 before step 5. However, sometimes step 4 and step 5 can be combined if all processes are going to be defined from scratch or redesigned.

4. Outlook

System and software engineering processes in the space domain are complex, creative and critical for developing high dependability software. In particular, they need to be defined in a way that they can be tailored to varying contexts (such as technologies, organizational characteristics, project constraints) without losing process quality and compliance to standards. This requires an enhanced tailoring concept and, as a consequence, a deep understanding of the variabilities of a process [Ocampo et al., 2005]. In the product domain, the software product line engineering approach allows for effective reuse of software artifacts. Software processes should be managed in a similarly systematic way, i.e., proactive development of integrated reusable processes and an organizational framework that institutionalizes the process management activities need to be established. Applying such concepts also to multi-organizational projects is a challenging but necessary task for the future.

Acknowledgement

In the context of cooperation projects between the Fraunhofer Institute for Experimental Software Engineering (IESE) and the Japan Aerospace Exploration Agency (JAXA), a comprehensive process investigation was performed and currently process management scoping and documentation activities are being performed. The results of these projects influenced the the roadmap description in this article. The author would like to thank all persons who contributed to these projects. The author would also like to thank Sonnhild Namingha from the Fraunhofer Institute for Experimental Software Engineering (IESE) for reviewing the first version of the article.

Biography

Jürgen Münch is Department Head for Processes and Measurement at the Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern. From November 2001 to December 2003, Dr. Münch was an executive board member of the temporary research institute SFB 501 "Development of Large Systems with Generic Methods" funded by the German Research Foundation (DFG). Dr. Münch received his PhD degree (Dr. rer. nat.) in Computer Science from the University of Kaiserslautern, Germany. Dr. Münch’s research interests in software engineering include: (1) modeling and measurement of software processes and resulting products, (2) software quality assurance and control, (3) technology evaluation through experimental means and simulation, (4) generic methods for the development of large systems, (5) technology transfer methods. He has been teaching and training in both university and industry environments, and also has significant R&D project management experience. Jürgen Münch is a member of IEEE, the IEEE Computer Society, and the German Computer Society (GI), a member of the program committee of various software engineering conferences, and has published more than 50 international publications.

The department headed by Dr. Münch develops methods, techniques, and tools to provide companies with validated practices for

- getting and maintaining control over the definition and evolution of their processes
- measuring and managing the quality of their processes and products
- analyzing their products and processes for decision support in software development.

Therefore, the research focuses on

- rationalized process management and improvement (process documentation and evolution, processes and techniques for contract development and COTS integration, continuous process improvement)

- measurement-based quality management (goal-oriented measurement, data-based quality modeling, cost estimation and risk management)

- goal-oriented process and product assessments (adaptable process assessments, cost-effective code assessments)

Technology-wise, this department is well-known regarding (descriptive) process modeling and goal-oriented measurement. Its major strengths are the focus on descriptive process modeling as a starting point for process improvement, the definition of measurement programs supporting the individual needs of companies, and assessment methods (FAME) customized to the needs of specific companies.

Fraunhofer IESE’s Activities in Japan

The Fraunhofer Institute for Experimental Software Engineering IESE offers methodological instruments to design software development processes in a plannable manner and to be able to bring software-based products to market more efficiently by consistently applying engineering-style principles. The work foci of software engineering, project- and quality management as well as experience management and human resource development concentrate on typical problem areas such as quality defects, time and budget overruns, or lack of sustainability of the established methods in practice. The Japanese Information-technology Promotion Agency (IPA) has signed a long-term cooperation agreement for collaborative research with Fraunhofer IESE. The main purpose is to support the Japanese Software Engineering Center (SEC). The long-term relationships of Fraunhofer IESE with Japanese universities such as the renowned universities in Osaka and Nara in the area of software engineering led to intensive collaboration in terms of contract research with companies such as Ricoh Co., Ltd. and Fujitsu and organizations such as the Japan Aerospace Exploration Agency (JAXA).

References

Basili, V.R., Weiss, D.: A Methodology for Collecting Valid Software Engineering Data. In IEEE Transactions on Software Engineering, Vol. SE-10, No. 6, pp. 728-738 (1984)

Basili, V.R., Caldiera, G., Rombach, D.: Experience Factory. Encyclopaedia of Software Engineering, Vol. 1, pp. 469-276, 1994.

Becker, U., Hamann, D., Verlage, M.: Descriptive Modeling of Software Processes. In: Proceedings of the Third Conference on Software Process Im-provement (SPI '97), Barcelona, Spain (1997)

Goldmann, S., Münch, J., Harald H.: A Meta-Model for Distributed Software Development, Proceedings of the Eighth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE’99), pp. , Stanford, California, USA, June 16-18, 1999

Grundin, J., Poltrock, S., Patterson, J.: CSCW Overview. Special Presentation at ACM Conference on Computer-Supported Cooperative Work (CSCW'96), Boston, USA (1996)

Raffo, D., Kaltio, T., Partridge, D., Phalp, K., Ramil, J.F.: Empirical Studies Applied to Software Process Models. In: International Journal on Empirical Software Engineering, vol. 4, No. 4 (1999)

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

Sheard, S.: Process Implementation. In: Proceedings of the 13th Annual Symposium of the International Council on Systems Engineering, Arlington, VA, June-July 2003

Verlage, M., Münch, J.: Formalizing Software Engineering Standards, Proceedings of the Third International Symposium and Forum on Software Engineering Standards (ISESS’97), pp. 196-206, California, USA, June 1-6, 1997