SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2000; **5**: 231–242

Practice Section

Integrated Evaluation Procedure for Software/ Hardware System Development Processes based on the Software Capability Maturity Model (CMM)‡

Winfried Russwurm*,† and Ludger Meyer
Siemens AG, Dept. ZT SE 3, Otto-Hahn-Ring 6, 81739 Munich, Germany

The Software Capability Maturity Model (SW-CMM®) from SEI (Software Engineering Institute) is a world-wide industrial standard for evaluating software development processes. Based on this model, Siemens is using an assessment procedure to evaluate development processes as a prerequisite for process improvement programs. The benefits of these improvements are limited by deep interactions between the hardware and system development processes and the software development process. This problem could be solved by comparable maturity levels for software, hardware and system processes. Thus, the need for an analogous evaluation method for hardware and systems was identified. The SW-CMM is analyzed to ascertain the analogous applicability to hardware development processes. Based on the results, a procedure is defined leading to a joint Siemens Process Assessment. It is now capable of evaluating all development processes (software, hardware, systems). This procedure has been applied successfully numerous times within the Siemens Groups. The findings of the joint assessments address predominantly project management functions like quality assurance, configuration management and risk management. As a consequence, several tailor-made process improvement programs have already been launched. These cover process issues irrespective of hardware or software, especially for project management functions. More benefits are identified by including a wider variety of roles in process training. Copyright © 2000 John Wiley & Sons Ltd

KEY WORDS: Capability Maturity Model; software/hardware system development process; maturity level; joint process assessments for hardware and software in systems; improvement of system development processes

*Correspondence to: Winfried Russwurm, Siemans AG, Dept. ZT SE 3, Otto-Hahn-Ring 6, 81739 Munich, Germany
†E-mail: winfried.russwurm@mchp.siemens.de
‡ The article is based in part on a conference paper presented at the 4th Annual European Software Process Group Conference (ESEPG '99), Amsterdam, 7–10 June 1999.

1. INTRODUCTION

The Software Capability Maturity Model (SW-CMM®)[1] version 1.1 from SEI (Software Engineer-

[1]SW-CMM is a registered trademark of the Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

ing Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania) (Mark *et al.* 1993a,b) is acknowledged as a world-wide industrial standard for evaluating software development processes. The SW-CMM is represented by a staged model consisting of five maturity levels (maturity increasing from level 1 to 5). There are 18 key process areas, each is assigned to one maturity level (see Figure 1). The process capability of a software development process is characterized by the compliance with the goals of the key process areas.

The CMM is used as a basis for assessment procedures to determine the process capability measured in terms of a maturity level. A typical procedure consists of a questionnaire approach, a standardized assessment walkthrough and a defined structure of results. The Siemens AG, among others, has recognized the importance of processes in software development and defined such a procedure as early as 1991 and has used it as a standard procedure since 1993 in over 150 process assessments for projects in about 60 business units world-wide. The fundamental basis is the SW-CMM and the SEI Assessment questionnaire, but it is also influenced by other sources, for example by the BOOTSTRAP (Haase *et al.* 1992) algorithms (threshold levels, quartile maturity levels etc.).

2. MOTIVATION FOR INTEGRATED SOFTWARE/HARDWARE PROCESS ASSESSMENTS

The focal point of all assessments is to use the evaluation results as a solid base for commencing a process improvement program tailored to the requirements of the business units. They need mature software development processes as inherent prerequisites for efficient development activities and for creating products which are a market success. Although the proportion of software required for product development is steadily increasing, only few software projects prevail without joint hardware development. Examples are the various embedded systems used in Siemens' products like in telecommunication devices, transport and installation control systems, medical technology, automation, and even in appliances. Therefore, the development processes within the Siemens Groups must be increasingly adjusted to provide for coordinated software and hardware design.

The success attained in software assessments and subsequent improvement programs (Mehner *et al.* 1998) leads to significantly improved maturity levels for software processes, verified by reassessments. The limitation to the evaluation of software

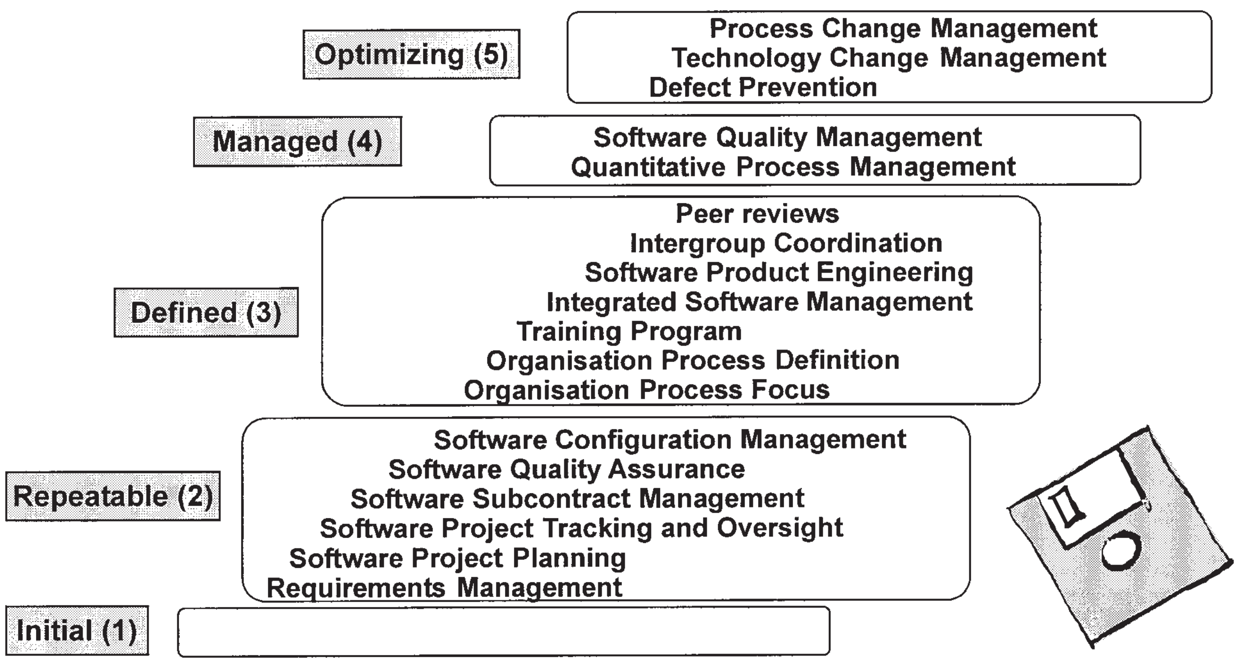

Figure 1. Maturity levels assigned to the different key process areas, according to the SW-CMM

processes was done because software is one of Siemens businesses key elements, but software is just one issue of a system supplier, and Siemens supplies predominantly systems. Therefore, it soon turned out that the software process is just an 'island' within a variety of other closely interacting processes. Assessors and improvers increasingly found limitations to the benefit of improving the software process. Once the software process had been improved, deficiencies of relating processes become apparent and the demand for improving these processes arises. Because of its close interaction with software in product design, the hardware development process is identified as a weak element in the chain, a chain that also sometimes lacks an overall system development process.

Structured processes lay the systematic groundwork for software development. The engineering of hardware has been a well-known tradition for decades, but has not yet caught up with the systematic software approach. Thus the need is identified to improve and evaluate the development process as a whole. For evaluation and improvement, regardless of whether the product consists of software, hardware or both, analogous methods have to be used. The requirements for a universal assessment procedure comprise capability for pure hardware processes and for system processes in combination of hardware and software. Another requirement is the compatibility with the former software process assessment procedure. The comparability to the maturity levels and findings obtained in earlier assessments is very important with regard to reassessments. Therefore, in conjunction with the requirement for a universal procedure, the new assessment procedure had to be tailorable according to the process considered. The new procedure had to be an analogous evaluation procedure using an analogous questionnaire, an analogous assessment walkthrough and an analogous structure of results and proposals and last but not least a comparable scale of maturity levels.

Apart from the SW-CMM, there exist more CMMs, e.g. for software acquisition, for security engineering and for systems engineering. The systems engineering CMM cannot fulfill the requirements, because it does not cover software and hardware development in detail and it is not compatible with the SW-CMM, particularly regarding the continuous representation of maturity levels compared to the staged representation used in the SW-CMM. The abundance of CMMs, which do not necessarily exhibit the same basic structure, led to the development and recent publication of the CMMI (CMM Integrated Framework), but still today there is no CMM for hardware available. Therefore a basis for hardware as an analogue to software has to be created. To achieve a dependable comparison of the evaluation procedures required by hardware and software, the SW-CMM is examined to ascertain if it could be applied analogously to hardware development processes. The analysis should also identify essential modifications and changes to ensure the portability. As a boundary condition, only a hardware development process should be considered, production processes are excluded.

3. RESULTS OF THE CMM ANALYSIS FOR APPLICABILITY TO HARDWARE PROCESSES

3.1. Analysis of the Key Process Areas of the SW-CMM® for their Pertinence in Hardware Development Processes

The architecture of the SW-CMM® shows its composition of six key process areas for maturity level 2 (Repeatable), seven for maturity level 3 (Defined), two for maturity level 4 (Managed) and three for maturity level 5 (Optimizing). The CMM requires defined goals for all of these 18 key process areas (see Figure 1); specific criteria are required for attaining these objectives. These criteria – commitments, abilities, activities, measurements and verifications – are termed 'common features'. All of these goals and common features are analyzed to ascertain if and to what extent they could be similarly employed in hardware development processes. The investigation analyzes whether:

- the common features could represent hardware development without making any changes or by simply making nomenclature alterations like merely changing the term 'software' to 'hardware';
- modifications are required in the explanations and examples associated with the common features;
- essential modifications are necessary to apply

the software common features to hardware, including the addition of more common features or their omission;
- strongly software-affiliated key process areas should be fundamentally revised;
- key process areas are completely not applicable to hardware;
- additional key process areas can be identified which are required just for hardware or for mixed hardware/software system development.

3.2. Easily Modifiable Key Process Areas

The analysis confirms that the software key process areas listed below can be applied to hardware without any changes or by simply interchanging the terminology (i.e. changing the word 'software' to 'hardware'):

- Requirement Management (RM)
- Organization Process Focus (PF)
- Organization Process Definition (PD)
- Training Program (TP)
- Intergroup Coordination (IC)
- Peer Reviews (PR)
- Quantitative Process Management (QP)
- Software Quality Management (QM)
- Defect Prevention (DP)
- Process Change Management (PC)

3.3. Key Process Areas Requiring Minor Modification

On the other hand, there are key process areas requiring more modification than merely changing terminology, because they cannot be explicitly applied to hardware. The changes comprise the revision of examples and explanations and the addition of new hardware-dependant common features.

- *Software Project Planning (PP) and Software Project Tracing (PT)*. The 'critical computer resources' within the SW-CMM® can be formulated in more general terms for hardware as well as for systems (i.e. critical system resources). Effort and size estimations must be revised for hardware products (i.e. number of various board assemblies, components required etc.). The common features should be extended to include the planning and tracking of production costs, lifecycle costs and profitability.
- *Software Subcontract Management (SM)*. Alterations should be made describing special hardware requirements for selecting subcontractors. Amendments should be made regarding prefabricated units and component procurement.
- *Software Quality Assurance (QA)*. Activities for planning and implementing special standard tests and approval tests (i.e. electromagnetic compatibility tests or mechanical resistance tests) have to be supplemented for hardware. Quality assurance plans must be completed to assure that resources and equipment for testing and verification are provided for in advance.
- *Software Configuration Management (CM)*. For software it is of prime importance to properly administrate documents, code, tools and test procedures for different configurations and their changes. CM for hardware must administrate documents, various devices and components, testing equipment and design tools. The common features should be extended to include Component Management (standard components, procedure for obsoletes, data sheets etc.).
- *Integrated Software Management (IM)*. Engineering and management activities are integrated into a coherent, generic standard development process to be implemented throughout the company. The individual projects tailor the company-wide standard processes according to rules, resulting in customized project processes. Special emphasis is placed on the methods and standards employed. For hardware, this key process area can be applied analogously by modifying and supplementing some examples.
- *Technology Change Management (TM)*. Technology Change Management involves identifying and evaluating new technology, and implementing it throughout the company when it proves appropriate. Additional technology areas have to be added to the SW-CMM which come about as a result of new production procedures or advancements, e.g. in chemistry, physics or material sciences. The organization should utilize the latest state-of-the-art methods. At the same time, it is necessary to pursue activities leading to advancements developed by the company itself. The company should not only use external know-how, but strive

for technological leadership resulting from in-house initiatives.

3.4. Key Process Areas Requiring Comprehensive Revisions

The key process area 'Software Product Engineering' (PE) describes the software development process based on the model 'Requirements – Design – Code – Testing – Integration – Integration Testing – System Testing'. The examples and descriptions in this key process area are dominated by the needs of a software development process.

For the transfer to hardware, fundamental revisions are required; for hardware, the development model 'Requirements – Design – Implementation – Testing – Integration – Integration Testing – System Testing – Field Trial' is more suitable. Design documentation for production purposes and the assembly of prototypes takes place in the 'Implementation' phase. The 'Field Trial' phase is not included in the SW-CMM® but it is common in software as well as in hardware. An interface to prototype fabrication and mass production should be planned to accommodate integrated methods and tools required by hardware. For the design of a system, the development models of software and hardware have to be matched and fitted under the overall system development process.

It is possible to develop software within a project using uniform design methods for all design units. In contrast, different methods are required for hardware, i.e. for creating board assembly units or for creating ASICs (application specific integrated circuits). Methods and tools can vary depending on the design sector. Thus, this key process area considers four different hardware design sectors:

1. circuit design;
2. mechanical design;
3. ASIC;
4. special units.

This particular aspect is displayed in Figure 2 within the system development together with the various development phases required in software, hardware and system development. Special units include, for example, sensors, actors, electroceramics or antennas. The development of these special units can be different to, for example, circuit design.

Comprehensive revisions and enhancements have to be made to the regulations and examples connected with this key process area for hardware processes. Interfaces within the hardware, as well as between hardware and software, must be augmented to consider system design factors.

3.5. Additional Key Process Areas Required for Hardware

In the SW-CMM® patents are not currently taken into account. Patents and intellectual property rights are, however, of prime importance in hardware development and they are also becoming more significant in software development. Thus, the key process area Patent Coordination (abbreviated PA) was included and assigned to maturity level 2.

The common features for Patent Coordination are concerned primarily with systematic studies and evaluations of foreign patents, the methodical identification of own intellectual property rights and patents from the project results and in-house know-how, and correlating patents to form a patent strategy. The goals of the new key process area comprise, for example, that the patent coordination activities are planned and that any patents relevant to the project are controlled.

4. CHARACTERISTICS OF THE UNIVERSAL ASSESSMENTS

4.1. Definition and Properties of the Questionnaire

The analysis of the portability of the SW-CMM® to hardware development processes demonstrates that most of the aspects involved in systematic hardware development are provided for when an analogous conversion ensues. Based on the results, modifications and enhancements described above, a revised SW-CMM® can be used as a standard model for developing hardware. The groundwork is laid for an integrated evaluation procedure for software, hardware and, furthermore, for combined software/hardware systems.

The next step is to incorporate the hardware-dependent elements within the existing software questionnaire to create an all-purpose assessment questionnaire with provisions for an superordi-

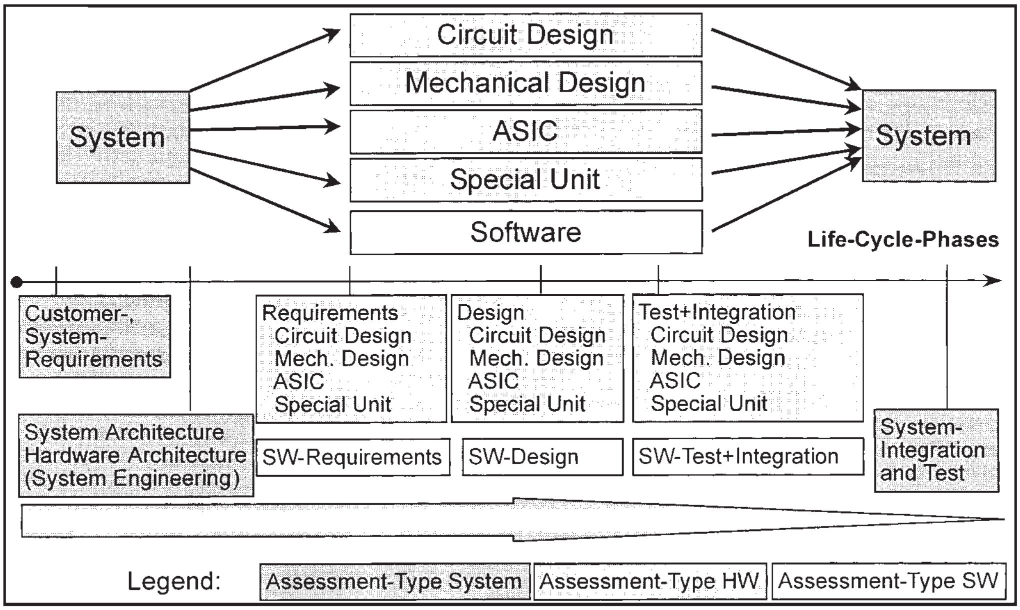

Figure 2. System development process tailorable to process assessments for software, system and four different hardware design sectors

nated system process. The structure of the questionnaire (see Figure 3) is maintained. The questions filed under 'Process' remain almost unchanged as a consequence of Section 3.2; e.g. process focus and process improvement are universal to all development processes and even to processes in general. The questions filed under 'Project' are supplemented according to the results of Sections 3.2, 3.3, 3.5 with emphasis on CM, SM and the new key process area for patent coordination. These are now applicable to any development process.

As the major consequence of Section 3.4, the questions regarding software engineering remain unchanged, while in parallel questions covering

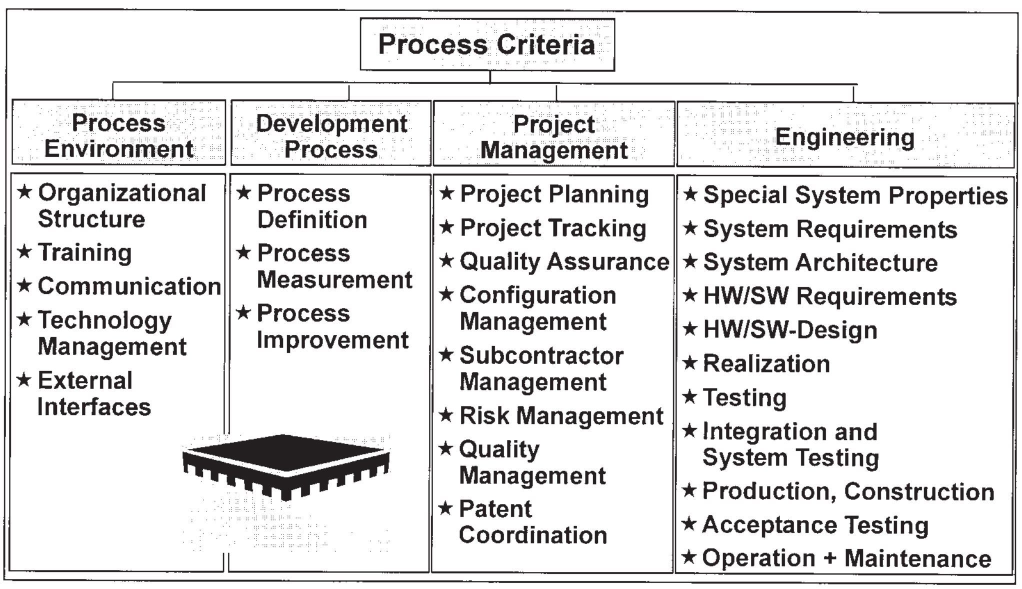

Figure 3. Process theme areas covered by the Siemens Process Assessment for software, hardware and systems

hardware engineering are defined. For design and implementation, the hardware section of the questionnaire is diversified to cover the four hardware design sectors (see Figures 2 and 3). To include the superordinated system process, system requirements, system architecture, system integration and test are integrated in the lifecycle phases. Firmware, often designed within the domain of hardware processes, is considered as software per definition. All questions are labeled to characterize their relevance to software, hardware or software/hardware-system processes. By tailoring, i.e. omitting questions not relevant for the process considered in practice, it is ensured that the questionnaire is capable of fulfilling the requirement for compatibility to the former software questionnaire. The new universal questionnaire also supports the principle of commonality; assessors do not have to handle different questionnaires and different assessment procedures. This helps to keep the conduction of assessments simple and efficient.

According to the results of Section 3, the five CMM maturity levels are also used as hardware and system evaluation measurement criteria in an analogous way. This means that, for example, relevant questions for hardware project planning are assigned to maturity level 2, as they are for software. They lead to standardized evaluations when comparable command of the process exists.

4.2. Assessment Walkthrough and Result Structure

In addition to the questionnaire, the assessment walkthrough and the result structure are also examined for their compliance to the new questionnaire. As a result it turns out that the assessment walkthrough does not have to undergo major modifications. The same applies for the result structure, which can be kept except for the extension regarding hardware and system engineering.

The assessment walkthrough starts with preparations including the tailoring of the questions asked according to the development process and projects considered (hardware, software or systems). A kick-off presentation with all participants is followed by a senior management interview and a site assessment interview, which covers mainly the process documentation. The approximate duration of each step can be seen from Figure 4. The project assessments cover mainly the practice process. They can be either conducted as for pure software process, or pure hardware process or system process. The results of each project assessed are discussed with the participants in individual feedback sessions. This is followed by producing a summary of the detailed results. The summary is discussed with the senior management in a separate feedback session and on the last day of the assessment the results are presented to all participants.

In the feedback sessions and final reports, a standardized result structure is used. The presentation of the complete set of findings etc. is organized according to the process areas shown in Figure 3. For each process area (or cluster), several maturity levels are obtained from the relevant questions. This comprises a maturity level for the documented process, a maturity level for the implementation of every project considered and a maturity level for the overall process. Figure 5 shows a characteristic result structure for the planning and controlling cluster. Detailed verbally expressed findings as well as recommendations and proposals for improvements supplement the result structure. According to the integration of, for example, hardware-specific questions in the questionnaire, the process areas now include hardware engineering clusters and system engineering clusters. If necessary, also a diversification for different hardware design sectors can be applied. Apart from the clear visualization of strengths and weaknesses according to process areas, it is now easy to assign improvement measures and proposals to the relevant processes (hardware, software, systems).

In practice, an assessment leads to a large number of proposals and measures. To filter out the most beneficial improvement areas, priorities are obtained for all measures and these will show up compacted in the portfolio diagram (see Figure 6). The representation shows the priority for the benefits determined by the assessors versus the importance according to the business goals determined by the senior management. All action clusters positioned within the outlined target area indicate a high likelihood for inclusion in a subsequent improvement program.

Summarizing, a complete analogous assessment procedure for hardware processes and for mixed software-hardware systems was developed. It

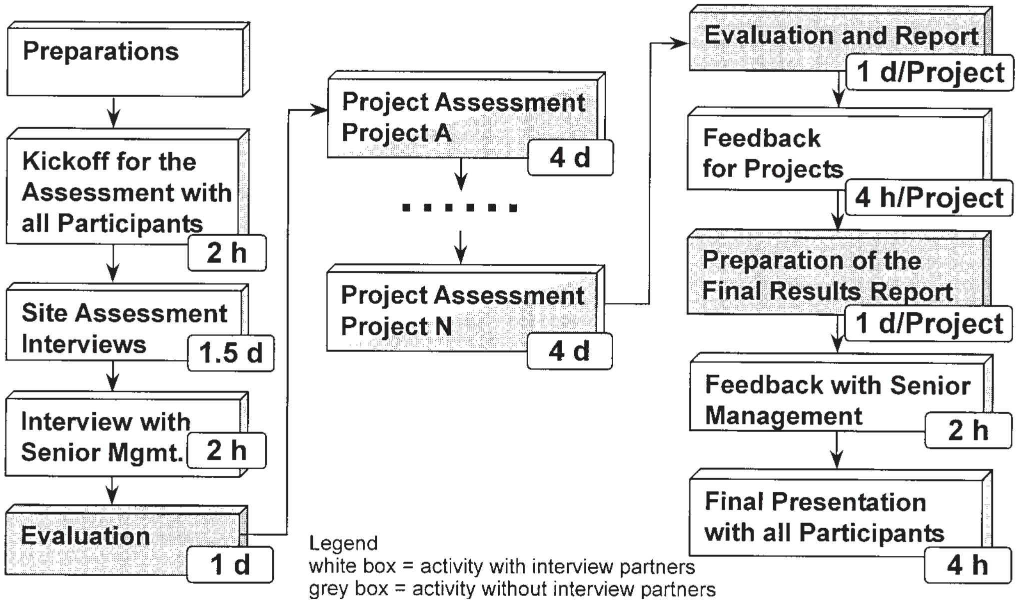

Figure 4. A schematic walkthrough of the Siemens Process Assessment for software, hardware and systems

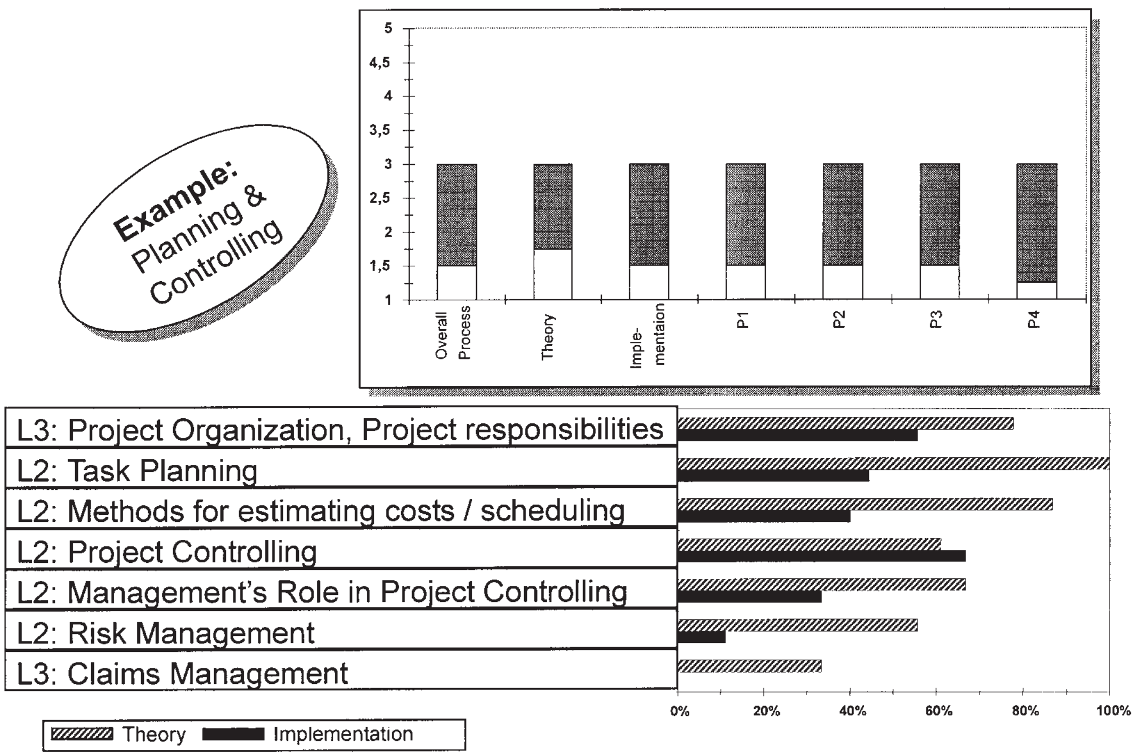

Figure 5. Characteristic result structure for the planning and controlling cluster. Detailed verbally expressed findings and recommendations for improvements are supplemented

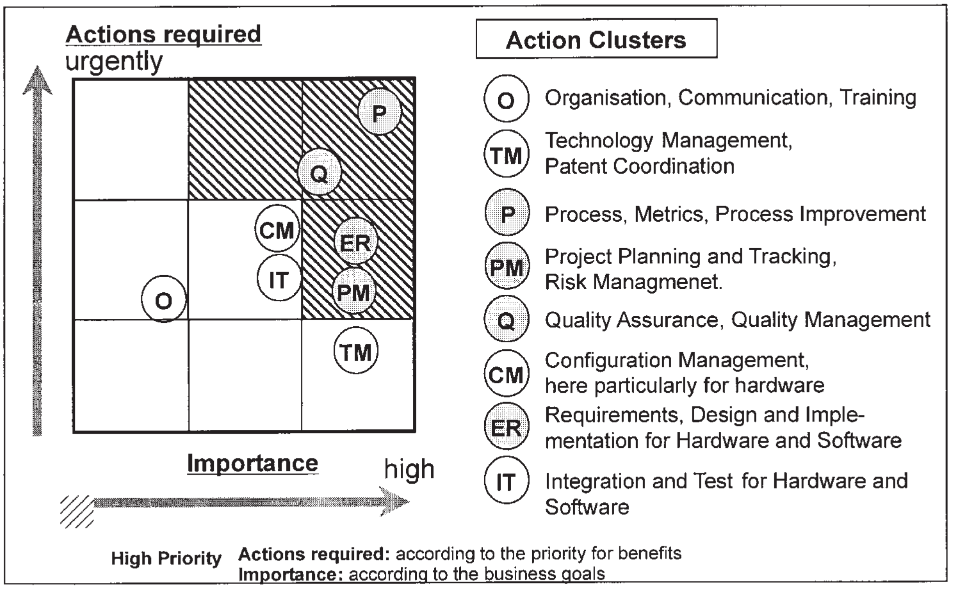

Figure 6. Priority for the benefits versus importance for business goals. Action clusters positioned within the outlined target area indicate high priority for subsequent improvement

forms the basis for comprehensive assessments to be conducted in the same way as that employed for software, to which the new procedure stays compatible (Russwurm and Meyer 1999).

5. EXPERIENCES FROM PERFORMED ASSESSMENTS AND DIRECTIONS OF IMPROVEMENT

5.1. General Experiences

A pilot version of the integrated assessment procedure was applied to processes within the company. Analysis information gained from the projects confirms that the procedure is well-received. It includes all of the important aspects required by hardware development processes and by mixed hardware/software systems. It is satisfactory for implementing improvement programs. The assessment procedure is now released for broad utilization within the Siemens Groups.

The demand for these joint assessments is steadily increasing. Meanwhile the proportion of requests for joint assessments is equivalent to the pure software assessments. Several dedicated improvement programs have already been launched as a result of the integrated assessments and yield significant contributions to process maturity for the overall development process.

5.2. Directions of Improvements – Project Management

The improvement programs based on joint assessments cover a range of engineering and project management practices. For the project management functions, the improvement measures concentrate on common methods and practices across the complete development project or system, and avoiding insular methods for the software subproject. This comprises particularly risk management, project planning and tracking, quality assurance and configuration management, as well as the implementation of a systematic patent coordination. In addition, common tools will help to sustain the management functions.

Improvements for quality assurance cover, for example, a common quality assurance plan for the system and reviews for all work products using harmonized review methods. The situation of own and foreign patents and the creation of intellectual property has always been of strong relevance to hardware design. Patents are getting more and more important for software, because the situation is changing towards an enhanced software patentability. Therefore, a systematic approach in patent coordination for hardware and software regarding the identification of foreign patents and their analysis, as well as the identification of own intellectual, is included in improvement programs. Another beneficial area of improvement is the installation of superordinated risk management across the project, though risk management is typically known as a software project management function. By the division of projects into subprojects, hardware and software may be developed in different departments and different environments. Therefore, room for improvement was found for change management and configuration management regarding an overall process for all subprojects and the communication of changes to all project members.

5.3. Directions of Improvements – Engineering

There is an excellent hardware design tradition for physical, viewable and touchable objects maintained for decades. The pressure of product and fabrication costs leads to the permanent innovation of hardware technologies, especially in the field of electronics and semiconductors. Areas of improvements are the enhanced definition of consistent and integrated design processes across all four design sectors (see above) with clear interfaces to software and system activities. In addition, reuse respective asset commonality has to be upgraded to a managed black-box reuse across software and all hardware design sectors. Increasingly complex systems demand comprehensive requirements management activities across the whole system and a strong separation between requirement analysis and design activities at all levels of system structuring.

As a specific hardware design issue, EMC (electromagnetic compatibility) is no longer considered as a side-effect that can be managed somehow. Owing to faster clock frequencies, quality aspects and legal requirements, EMC increasingly is a project risk and is included in risk management. Another way to counter this problem is to consider EMC relevant parts like shielding as dedicated design objects. Assigning planned design activities to EMC will also account for the increasing consumption of design resources by EMC. An additional support could be the integration of dedicated EMC activities into the development process.

5.4 Directions of Improvements – Process Management

As a logical consequence, a system engineering process for common and integrated design of hardware and software over all lifecycle phases and across all project management functions is indispensable. Such a superordinated process will eventually also match different paradigms now sometimes used for the design of system components. As an example, the paradigm for simultaneous engineering is demanding a fast transition of hardware design units to production, while software developed using an evolutionary paradigm might create frequent, severe and costly hardware changes.

Another issue is the overcoming of the artificial separation between hardware and software. It is a waste of time to discuss whether firmware is software or a part of hardware or something in between, or if firmware for DSPs (digital signal processor) is closer to hardware, and so on. The correct view is that software, as well as circuits and all other design components of a system, has to be developed in a mature environment according to an integrated process.

Nevertheless, the definition of processes and methods requires just a minor effort. The major part of process improvement activities have to be planned for the translation into practice. Both a communication plan and a training plan are required. Up to now, process training was predominantly limited to software engineers and project managers. A training plan for an integrated process will include many more roles like system engineers, hardware engineers, production engineers, logistics personnel, accounting managers, personnel for the sales and administration of business. These have to experience role-based training as well as cross-functional training. A communication plan will include in-project coaching, frequent publications in company papers, flyers, giveaways like stickers etc. and, last but not least, involvement of all key people from the very beginning.

6. CONCLUSIONS

After some years of intensive software process evaluation and improvement according to the goals of the CMM, the need for an expanded view of development processes towards hardware and systems has become more and more apparent. Our study shows, that the structure and philosophy of the SW-CMM can be applied to hardware processes. Every key process area can be used for hardware after distinctive changes for hardware design processes; the extent of the changes depends on the key process area. A new key process area for patent coordination should be added. The changes are the basic prerequisite for a universal system assessment procedure, which can also be employed for evaluating pure hardware or software processes. The expanded questionnaire and the assessment walkthrough are still compatible with the former one for software only. This extension for hardware makes it possible to evaluate all of the development processes which exist within the Siemens Groups in a universal way. Several system assessments have been performed. Further work on the evaluation and improvement of integrated processes will be done as part of studies regarding the CMM Integration Framework of the SEI now in progress.

The most prominent result from the assessments performed is that for designing a system composed of software and one or more hardware design sectors an overall design process is indispensable. It covers all design elements and activities including comprehensive activities and methods for all management functions and process functions. Such a process is a prerequisite for further process improvements and the benefits generated thereby. Neither the software process nor the hardware development process should be considered as an ‘island’.

The scope of this article is the integration of all development activities for a system under the roof of an integrated design process, but this process as a superset of the software process is itself just one of the business processes within a company. The matching of other business processes, like product definition, acquisition and after sales service, with the design process and clearly defined interfaces between all business processes will mark another field of possible improvements and further benefits.

REFERENCES

Paulk MC, Curtis B, Chrissis MB, Weber CV. 1993a. Capability Maturity Model for Software, Version 1.1. Software Engineering Institute, CMU/SEI-93-TR-24, DTIC Number ADA263403. Available in Internet: http://www.sei.cmu.edu/activities/cmm/obtain.cmm.html

Paulk MC, Weber CV, Garcia SM, Chrissis MB, Bush MW. 1993b. Key Practices of the Capability Maturity Model, Version 1.1. Software Engineering Institute, CMU/SEI-93-TR-25, DTIC Number ADA263432. Available in Internet: http://www.sei.cmu.edu/activities/cmm/obtain.cmm.html

Haase VT, Messnarz R, Cachia R. 1992. CHAP Software process improvement by measurement BOOTSTRAP/ESPRIT Project 5441. In *Shifting Paradigms in Software Engineering, Proceedings of the 7th Joint Conference of the Austrian Computer Society (OCG) and the John von Neumann Society for Computing Sciences (NJSZT) Klagenfurt, Austria*, Mittermeir R (ed.). Springer Verlag: Wien; 32–41.

Mehner T, Messer T, Paul P, Paulisch F, Schless P, Völker A. 1998. Siemens process assessment and improvement approaches. Experiences and benefits. Proceedings of the Twenty-Second Annual International Computer Software & Applications Conference (Compsac 98), Vienna, Austria, 19–21 August IEEE Computer Society, Los Alamitos, California.

Russwurm W, Meyer L. 1999. Integrated evaluation procedure for software/hardware system development processes based on the Software Capability Maturity Model (CMM). Proceedings of the 4th Annual European Software Engineering Process Group Conference, 7–10 June, Amsterdam, Netherlands.