Goal-oriented Improvement of Requirements Processes

Barbara Paech and Erik Kamsties,
Fraunhofer Institute for Experimental Software Engineering,
D- 67661 Kaiserslautern, Germany
{paech, kamsties}@iese.fhg.de

Abstract

Software engineering improvement methodologies offer an organizational framework for goal setting, planning, execution, and monitoring of improvement programs as well as technologies for assessing, modeling, and measuring of processes. These methodologies help in the identification of key practices to be improved, but they do not provide guidance on how to improve a specific practice like the specification of software requirements. In contrast, requirements engineering research and practice offers a wide variety of methods and techniques for specific software requirements practices, but little guidance on how and when to establish them within an organization.

In this paper, we introduce the requirements-specific assessment and improvement method RE-KIT-FRAIME. This method bridges the gap between improvement and requirements practices in two ways. It ties the building-up of requirements practices to an overall improvement method and it gives guidance on how to choose and design a particular practice.

1 Introduction

Software process improvement methodologies aim at improving software quality through the improvement of the software engineering process. During the last decade, a variety of such methodologies has emerged: Top-down approaches like CMM [1] and SPICE [2] structure software development into different processes, each process with its own set of key practices. Through an assessment of its own processes against the standard processes and practices, a company can identify practices which need to be established, and maturity levels give guidance on which practice to establish first. Bottom-up approaches like QIP [3] and GQM [4] support the development of company-specific strategic process practices, accompanied by a framework for monitoring and experience packaging. The recently developed PROFES method helps to ensure alignment of the chosen strategic process capabilities to the product quality goals [5].

Both kind of approaches help to get the “big” picture of a company’s or a project’s software process, but give little guidance on how to design a specific

Published at INSPIRE 2000, London, September 7-8, 2000

practice within an improvement effort. Typically, the practices are still quite high-level (see e.g. Table 1) and their implementation requires a number of choices which are not supported by the improvement methodologies. In the realm of requirements engineering, typical choices are

- what to include in a requirements document,
- how often to talk to the which user or customer,
- how to represent specific requirements, and
- how to elicit needed information.

Requirements engineering research and practice have developed a wide variety of methods and techniques to solve the above questions (see e.g. [6,7,8,9,10,11,12] for recent conferences, journal and books). Typically, the solution provided by the method or technique is described in detail, but requirements process engineers are left alone with the questions of

- how to choose between different possible solutions or
- how to establish a specific solution in an organization.

There is a clear gap between the requirements practices and the improvement practices, which hinders the efficient introduction of well chosen requirements methods into an organization. No existing method explicitly bridges this gap.

RE-KIT-FRAIME has been developed to support requirements process improvement. It ties together improvement and requirements practices in order to answer the questions listed above. In particular, it offers a framework for goal-oriented choice and design of improvement actions for requirements engineering processes.

In the following, we review improvement and requirements standards for their support of requirements process improvement in more detail. Then we present RE-KIT-FRAIME and industrial experiences on its application. We conclude with a summary and future work.

2 Improvement Methodologies for Requirements Processes

There are two main sources for guidance on improvement actions for requirements processes:

- Key practices as defined in the standard assessment models like CMM [1] or SPICE [2].
- Best practices as collected in international requirements standards like IEEE-830-1998 [13] or practice collections such as done in the REAIMS project [12].

In the following, we sketch these sources and evaluate the guidance they give for choosing and designing specific improvement actions for requirements processes.

2.1 Requirements Practices in Assessment Models

Typical recommended requirements practices can be found in the SEI Software Capability Maturity Model (CMM) [1], the SEI Software Acquisition Capability Maturity Model (SA-CMM) [14], and the standard for software process assessment ISO/IEC TR 15504 (SPICE) [2]. Table 1 depicts the recommended requirements-related practices of the latter, since they subsume the former two. CMM practices concentrate on management practices such as traceability, while SA-CMM practices focus on the customer-supplier relationship. Of course, requirements activities also include activities of support processes such as verification (SPICE-SUP.4), which we have omitted here for sake of brevity.

Table 1: Requirements Processes in SPICE

SPICE-CUS.1.1.BP1	Identify the need
SPICE-CUS.1.1.BP2	Define the requirements
SPICE-CUS.1.1.BP4	Define acceptance criteria
SPICE-CUS.1.2.BP1	Define acquisition requirements
SPICE-CUS.1.3.BP1	Provide supplier feedback
SPICE-CUS.1.3.BP2	Review development with supplier
SPICE-CUS.2.BP1	Prepare response
SPICE-CUS.3.BP1	Obtain customer requirements and requests
SPICE-CUS.3.BP2	Agree on requirements
SPICE-CUS.3.BP3	Establish customer requirements baseline
SPICE-CUS.3.BP4	Manage customer requirements changes
SPICE-CUS.3.BP5	Understand customer expectations
SPICE-CUS.3.BP6	Establish customer query mechanism
SPICE-ENG.1.1.BP1	Identify system requirements
SPICE-ENG.1.1.BP2	Analyse system requirements
SPICE-ENG.1.1.BP3	Describe system architecture
SPICE-ENG.1.1.BP4	Allocate requirements
SPICE-ENG.1.1.BP5	Develop release strategy
SPICE-ENG.1.1.BP6	Communicate system requirements
SPICE-ENG.1.1.BP7	Establish traceability
SPICE-ENG.1.2.BP1	Specify software requirements
SPICE-ENG.1.2.BP2	Determine operating environment impact
SPICE-ENG.1.2.BP3	Evaluate and validate requirements with customer
SPICE-ENG.1.2.BP4	Develop validation criteria for software
SPICE-ENG.1.2.BP3	Develop release strategy
SPICE-ENG.1.2.BP6	Update requirements
SPICE-ENG.1.2.BP7	Communicate software requirements
SPICE-ENG.1.2.BP8	Evaluate software requirements
SPICE-ENG.2.BP1	Determine maintenance requirements
SPICE-ENG.2.BP3	Analyse user problems and enhancements
SPICE-ENG.2.BP4	Determine modifications of next upgrade

Since for every practice only a two or three sentences-explanation is given, together with some characteristics of resulting documents, it is obvious that this does not provide enough guidance to answer the questions listed in the introduction.

2.2 Requirements Practices in Requirements Standards

The IEEE Standards *Recommended Practice for Software Requirements Specification* 830-1998 [13], *Guide for Developing System Requirements Specification* 1233-1998 [15], and *Guide for Information Technology System Definition - Concept of Operations (ConOps) Document* 1362-1998 [16], as well as the result of the REAIMS project [12] describe more specific requirements practices. All of the standards provide detailed outlines of requirements documents. An example is shown in Figure 1. Std-1362 focuses on the description of changes in the operating environment, Std.1233 and 830 on the specification of the system or software as input to software developers. Std. 1233 in addition gives guidance on how to collect the necessary information. REAIMS [12] do not provide document templates, but collects best practices for requirements documentation, elicitation, analysis, negotiation, validation, and management. The top-ten guidelines are shown in Table 2. These practices are structured into maturity levels to allow assessment of requirements processes.

Table 2: Top-ten guidelines from REAIMS

Guideline	Description
3.1	Define a standard document structure
3.8	Make the document easy to change
9.1	Uniquely identify each requirement
9.2	Define policies for requirements management
6.1	Define standard templates for requirements description
6.2	Use language simply, consistently and concisely
8.2	Organise formal requirements inspections
8.4	Define validation checklists
5.2	Use checklists for requirements analysis
5.4	Plan for conflicts and conflict resolution

While the guidance provided by these standards is much better than the one provided in the assessment models, it is still not sufficient to select and design specific improvement actions, since there is no way the standards can reflect the wide variety of requirements process contexts.

1 Scope
1.1 Identification
1.2 Document overview
1.3 System overview
2 Reverenced documents
3 Current system or situation
3.1 Background, objectives, and scope
3.2 Operational policies and constraints
3.3 Description of the current system or situation
3.4 Modes of operation for the current system or situation
3.5 User classes and other involved personnel
3.6 Support environment
4 Justification for and nature of changes
4.1 Justification of changes
4.2 Description of desired changes
4.3 Priorities among changes
4.4 Changes considered but not included
5 Concepts for the proposed system
5.1 Background, objectives, and scope
5.2 Operational policies and constraints
5.3 Description of the proposed system
5.4 Modes of operation
5.5 User classes and other involved personnel
5.6 Support environment
6 Operational scenarios
7 Summary of impacts
7.1 Operational impacts
7.2 Organizational impacts
7.3 Impacts during development
8 Analysis of the proposed system
8.1 Summary of improvements
8.2 Disadvantages and limitations
8.3 Alternatives and trade-offs considered
9 Notes

Figure 1: Outline IEEE Std. 1362-1998 (ConOps)

For example:

- The ConOps standard gives guidance on how to describe changes impacted by a new software system, but does not reflect the quite typical situation where part of the proposed system is an existing system, other parts are revisions of existing systems, and still other parts are newly developed by different suppliers. How detailed should the description of the current system be, how do you describe functionality that is shared between the current and the proposed system?

- The SRS standard gives guidance on how to structure functional requirements, but does not reflect the situation where there is a whole product line with overlapping functionality selected in different packages by different customers. How do you make commonalities for the developers explicit?
- All standards do not give guidance on where to focus an improvement in case of a big discrepancy of existing documents to the recommended ones.
- REAIMS provides preliminary guidance for improvement focus through the maturity level, but as well argued in the improvement literature [17] top-down approaches to improvement are not enough. Bottom-up approaches as QIP and GQM are needed to ensure that improvement is driven by and guided by the knowledge of the development organization.

2.3 Summary of Requirements-specific Improvement Guidance

Table 3 gives an overview of the level of granularity, the guidance for the selection, and the implementation of requirements-specific practices. It makes obvious that there is a need for more guidance on selection and design of improvement actions.

Recently, the PROFES project has introduced a third source for guidance on improvement actions: Product-Process-Dependency-Models (PPD) [5]. PPDs document the impact of a specific practice like inspections on a specific product quality goal like reliability when applied in a certain process. PPDs also contain a description of contextual situations in which - based on experiences - the stated impact can be expected. This documentation of experiences is a very promising idea. Because of the context description, the described practices can range from very specific ones to very general high-level ones. However, it will take considerable effort to collect such experiences. Also, a framework is needed to which the individual experiences can be related.

In the next section, we propose the method RE-KIT-FRAIME which provides such a framework on how to choose and design a specific requirements process improvement action.

Table 3: Summary of Requirements-specific Improvement Guidance

	Level of Detail	**Selection Support**	**Design Support**
SPICE	Very high-level	6 Maturity levels	Characteristics of documents
IEEE	Quite high-L+level	None	Document templates
REAIMS	Partially high-level, partially quite detailed	3 Maturity levels	One-page description

3 RE-KIT-FRAIME: Requirements-specific Improvement

The Fraunhofer Requirements Assessment and Improvement Method RE-KIT-FRAIME is part of RE-KIT. RE-KIT (*R*equirements *E*ngineering with emphasis on *K*nowledge management, *I*nterface specification and *T*raceability) is a collection of requirements engineering methods that is based on three common principles:

- Requirements are a particular kind of *knowledge* of the system (namely, knowledge concentrating on the outside view of the system) acquired and used over the whole system lifecycle. Thus, knowledge management principles and techniques have to be applied in requirements engineering.
- The core of the requirements describes the *interface* of the system, namely its functional and non-functional properties. This interface can only be adequately engineered based on knowledge of the two worlds that it connects: the user world and the machine world.
- Change of requirements is inherent in software engineering because of the new knowledge generated through usage of a new system. Thus, *traceability* of requirements from the user world to the machine world is indispensable.

Two particular examples of methods contained in RE-KIT-FRAIME are

- RE-KIT-MUC: a *m*ethod for the generation and usage of *u*se *c*ases, and
- RE-KIT-MAMBO: a *m*ethod for *amb*iguity detection and av*o*idance in natural language requirements specifications.

RE-KIT-FRAIME supports the efficient transfer of these methods and other requirements practices into requirements engineering processes. It has been developed based on our experiences with teaching and consulting requirements practices.

Figure 2 shows the main ingredients of RE-KIT-FRAIME: the Quality Improvement Paradigm (outer circle) and the process dimensions (inner circle). QIP provides the general process improvement framework. The process dimensions support in particular characterization and goal setting. Because the improvement focuses on the requirements process only, the improvement goals are typically on the level of specific improvement actions, like introducing a specific document structure or using a particular requirements engineering technique. Thus, in the following, we use the terms *improvement action* and *improvement goal* interchangeably.

The process dimensions give guidance on the facets of the requirements engineering process to be characterized as well as on the priority that should be given to specific improvement action. The combination of both allows a guided, situation-specific choice of requirements improvement actions based on general quality needs.

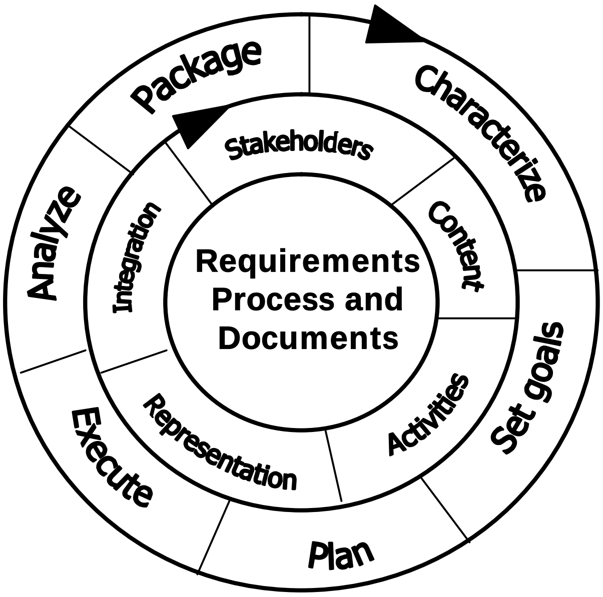

Figure 2 : RE-KIT-FRAIME

3.1 Process Dimensions

The process dimensions, depicted in the inner cycle of Figure 2, give guidance on how to model and assess the requirements process and on how to choose an improvement action. In particular, they allow to reflect the particular needs of the organization: e.g., while it may well be that the early derivation of test cases is adequate in one organization, it might be overkill in another organization where the requirements are particularly instable. Or, while the introduction of a requirements management tool improves efficiency in one organization, it might unreasonably slow down the requirements process of another organization. REAIMS has partitioned the recommended practices into three levels in order to reflect the maturity of the actual requirements process. The main contribution of RE-KIT-FRAIME is to reflect the organizations needs through the *up-front involvement of the stakeholders*. In the following, we sketch the five process dimensions.

3.1.1 Stakeholders

The requirements process is described by looking at the different documents produced by and the information needs of the stakeholders. The *stakeholders* are all the people producing and using the documents. So, the documents are not at first characterized by the content (as in the standards), but by the *role* they play during software development. Typical questions at this stage include: is this document the basis for test case generation? Is the customer expected to read the document and if so, what kind of decisions or feedback is he or she expected to make? Is this document only used in one project or in several projects related to a product line? This view on the requirements process can best be conveyed by a process diagram, e.g. an UML activity diagram, that shows for each document-related activity the incoming and outgoing information and the people involved.

The explicit information needs of the stakeholders are the basis for the *Set Goal*- step of the QIP-cycle. If, for example, the documents contain information that is not used by any stakeholder, but, on the other hand, miss information that is important for at least one stakeholder, a possible improvement goal could be to restructure the documents so that the necessary information for a certain group of stakeholders is concentrated in one document.

3.1.2 Content

Having determined the goals relating to the overall information needs of the stakeholders, goals for the detailed contents of the documents can be determined. At this level, it typically becomes apparent that even company-specific standards are usually too coarse-grained to uniquely determine the contents. Especially for non-functional requirements it is often not clear what they should cover. An example is the usual section "architectural requirements", which can mean anything from a coarse subsystem structure to a detailed run-time process structure. It is important to let the stakeholder fix a specific level of granularity sufficient and necessary for them. This results in improvement goals like a project-specific document standard or a company-wide standard including examples of different possible granularity levels.

3.1.3 Activities

Based on the knowledge of the detailed information needs of the stakeholders, goals for the activities that generate and use the documents can be determined. For *generation*, the important question is from where to get the needed content. Often a specific information in a document is left out, because it is difficult to get the necessary information, while other information is included only because it is easy to get. An improvement goal relating to generation could e.g. entail specific ways of customer-supplier interaction such as prototyping or work observation or the like. Other important improvement goals for generation relate to the validation steps for the documents, which could be formal inspections, informal reviews or, depending on the formality of the document, automated checks. Goals regarding *usage* mostly relate to requirements management, such as easy consistent change of the documents and easy traceability between requirements. Nowadays, requirements management is a fashionable focus for requirements process improvement. However, it is no use to establish an elaborate management program, if the document structure and the contents are not adequate to the stakeholders needs. This is why the stakeholder and the content dimension are considered first in RE-KIT-FRAIME.

3.1.4 Representation

The goals regarding representation are dependent on the decisions made with respect to structure, content, and activities. For example: while it might be desirable to represent requirements graphically, it might turn out that management is much more efficient for a structured text version. Or the use of the requirements document for test case generation mandates a specific order of presenting information in a use case description.

3.1.5 Integration

Finally, goals relating to the interaction between the requirements process and the other software development processes can be fixed. Such goals typically entail the identification of new requirements documents stakeholders. For example: requirements can support project planning by using the requirements for cost and effort estimation and project tracking. Or requirements specifications can be integrated with the generation of the user manual. With the identification of new stakeholders, all the other process dimensions and the identified goals for them have to be reconsidered.

In the above discussion of the process dimensions, we have recommended a specific order of using the process dimensions for characterization of the existing process and for goal-setting. This recommendation, however, might be overruled by project- or company-specific constraints. For example: a project might be interested in trying out a particular representation technique, even so the management issues for this technique are not solved. It is important, however, to be very conscious about such constraints, since they typically induce a further risk into the project.

3.2 Quality Improvement Paradigm

The last section has explained the RE-KIT-FRAIME technique for characterization and goal-setting for requirements processes. To tie characterization and goal-setting to an improvement method, RE-KIT-FRAIME has adapted the Quality Improvement Paradigm [3]. Our adaptation of QIP is similar to the product-focussed adaptation developed in the PROFES project [5]. For our purpose, the products are the requirements documents. Improvement actions are tied to quality goals for the requirements documents. This focus of documents is in line with the knowledge-principle of RE-KIT: the documents are important to make the knowledge on the software and its environment explicit. The QIP-steps of RE-KIT-FRAIME are explained in the following.

3.2.1 Characterize

The requirements document quality needs are gathered from the stakeholders of the requirements engineering process. If an overall improvement effort is carried out, these needs are aligned with the high-level requirements process goals (described as key practices in the assessment). For the detailed characterization, process modeling and a (possibly light-weight) application of the goal/question/metrics approach (GQM) [4] is used: The actual process is described and document quality goals are determined according to the process dimensions explained in the last section. In particular, the stakeholders participate in GQM-workshops to refine the document quality goals into correspondent metrics. Examples for document metrics include the adherence to standards, the degree of traceability supported in the documents, or the accuracy with which customer needs have been captured. These metrics can then be used for a detailed analysis of the existing process.

3.2.2 Set Goals

Based on the characterization results and again following the process dimensions, improvement goals are determined. As exemplified in the last section, these improvement goals are, in contrast to application of QIP in an overall improvement effort, on the level of specific changes to documents or requirements process activities.

3.2.3 Plan

Based on the identified goals, improvement is planned. There is no need to further select or describe the document or process changes, since the goals already determine the changes. However, still the organizational form of how to introduce the change and the measurements that allow the monitoring of the improvement have to be planned.

3.2.4 Execute, Analyze, and Package

The execution, analysis, and packaging of RE-KIT-FRAIME does not differ from the general QIP/GQM- methodology: during execution the changed documents and activities are applied and measurements are collected. Analysis aims at detecting deviations from the planned improvements and at documenting the lessons learned. Packaging makes these experiences available to the overall improvement effort.

4 Lessons Learned

We have applied RE-KIT-FRAIME principles in two different situations in two organizations. In the following, we sketch the application and our lessons learned.

4.1 Scoping a Requirements Documentation Structure

We came into the first company after it had conducted a CMM-assessment, which had identified the need to improve the requirements management practice. Internal discussion had revealed in particular the need to improve the documentation of the system requirements, since it served on one hand as the basis for the contracts with the customers and on the other hand as the basis for allocating the system requirements to hardware, software, and mechanical components. As typical for the embedded systems industry, the system is sold in several variants to different customers and the hardware, software, and mechanical components are also sold separately for integration with components from other suppliers.

In the QIP-characterization step, we first elicited the stakeholders of the requirements documentation and their particular information needs. We skipped the characterization of the activities, the representation, and the integration dimensions, since the company wanted to focus on the role and the content of the documents. Based on the characterization, we elicited detailed structure and content goals in discussions and by looking at existing documents. In the planning-step it was decided to redo the documentation of an existing system instead of applying it in an ongoing project. During execution and analysis, we restructured the information and validated the new structure with the stakeholders. For packaging, we extracted guidelines on how to document requirements according to the new structure. The stakeholders were quite satisfied with these guidelines. Regarding the use of RE-KIT-FRAIME, this project

- confirmed the fact that general assessments like CMM do not give enough guidance on how to improve the key requirements practices.
- supports the usefulness of the stakeholder-dimension: Requirements documents serve multiple purposes during a system development project, in particular in the context of complex customer-supplier relationships and product families. These purposes are best made explicit through stakeholder interviews.
- supports the priority of stakeholder needs over detailed content descriptions in order to determine an improved requirements documentation structure. The company was at first reluctant to support the effort for eliciting and interviewing the stakeholders and wanted to base the improvement solely on the existing documentation and the opinion of some key persons. However, the involvement of all the stakeholders made the different quality goals for requirements documents of individual systems or product line instantiations explicit.

4.2 Assessing the Quality of a Requirements Documentation

In the second company, the requirements documentation is of particular importance, because the development of major system parts is subject to outsourcing. Therefore, the company has established a review of the requirements documents, which is, however, subject to severe time constraints. We were involved to give advice on short-term improvements of a given requirements document, on long-term improvements of similar requirements documents, and on the review process itself. While the document adhered to a company-specific standard for requirements documents, previous project experience had revealed a need to improve the understandability of the requirements documentation to support the software supplier, who has only limited access to the future software system users. Also, there has been recognized a need to improve the accuracy of the requirements to support the verification and validation of their implementation by the procurer.

Since we did not have access to the stakeholders of the requirements, we concentrated for the characterization and the goal-setting on the supplier and the procurer and derived the document quality goals mainly from our general experience with procurement situations. The goals covered all process dimensions except the overall integration, since we had no knowledge on the overall software development process. Following the GQM approach, the goals were broken down into specific measurements, and suitable evaluation techniques were determined. These included checklists, traceability analysis and modeling. The choice reflected in particular the time constraints. As part of the characterization, we conducted a careful evaluation according to the chosen techniques and documented the results in a report. From these findings we derived the improvement goals. We were not involved in subsequent QIP-steps for realizing the improvement goals.

Again, the feedback of the involved stakeholders was very positive. In particular, this application of RE-KIT-FRAIME

- confirmed the fact that (company-specific) requirements document standards do not give enough guidance regarding the role and contents.
- confirmed the usefulness of the GQM approach for the determination of quality metrics. Although we did not have access to the stakeholders, this served to make our quality goals, as well as the context-dependent choice of evaluation techniques understandable to the company.
- supported the usefulness of the process dimensions for relating process issues to the document quality goals. However, we have not yet applied all dimensions for characterisation and goal-setting for a complete requirements process. In particular, prioritization and conflict resolution of different improvement goals is likely to require further support.

5 Conclusions

In this paper, we have made apparent the need for further support on requirements process improvement. We have presented RE-KIT-FRAIME, which provides further guidance, and we have sketched some experiences with its application. Obviously, there is further work needed. In particular, RE-KIT-FRAIME supports prioritization of improvement goals, but it does not give guidance on the relationship between product quality goals and requirements process improvements and on the relationship between requirements documents quality goals and improved requirements activities. As mentioned before, we think that Product-Process Dependencies (PPDs) provide such guidance. However, it will take time and effort to collect these experiences from industrial practice. Another way for collecting experiences with particular requirements techniques are student experiments such as [18], which has evaluated different requirements modeling techniques and their tool support. The context of student experiments is different from industrial settings. But student experiments can serve as input for industrial improvement actions from which evidence on requirements techniques in specific industrial contexts can be gained.

Acknowledgements

We thank Andreas Birk, Julio Leite, and the anonymous referees for helpful comments on earlier versions of the paper.

References

1. Software Engineering Institute. Key Practices of the Capability Maturity Model. Version 1.1, Technical Report, CME/SEI-93-TR-025, Carnegie Mellon University, 1993
2. ISO/IEC TR 15504 Information Technology - Software Process Assessment
3. Basili VR, Caldiera G. Improve Software Quality by Reusing Knowledge and Experience. Sloan Management Review, Fall 1995: 55-64
4. van Solingen R, Berghout E. The Goal/Question/Metric Paradigm, McGrawHill, 1999
5. Oivo M, Birk A, Komi-Sirviö S, Kuvaja P, van Solingen R: Establishing Product Process Dependencies in SPI. In: Proceedings of the 4th Annual European Software Engineering Process Group Conference (SEPG), Amsterdam 1999.
6. International Symposium of Requirements Engineering, IEEE Computer Society Press. Bi-annual series of conferences since 1993.
7. International Conference on Requirements Engineering, IEEE Computer Society Press. Bi-annual series of conferences since 1994.
8. Requirements Engineering Journal. Peri Loucopoulos, Colin Potts (ed.), Springer-Verlag. Since 1996
9. Macaulay LA. Requirements Engineering. Springer-Verlag, Applied Computing, 1995
10. Paech B. Aufgabenorientierte Softwareentwicklung. Springer-Verlag, 2000
11. Robertson S, Robertson J. Mastering the Requiremens Process. Addison-Wesley, 1999
12. Sommerville I, Sawyer P. Requirements Engineering – A good practice guide. John Wiley & Sons, 1997
13. Institute of Electrical and Electronics Engineers. Recommended Practice for Software Requirements Specification, IEEE Std. 830-1998
14. Software Engineering Institute. Software Acquisition Capability Maturity Model, Version 1.02, Technical Report, CMU-SEI-99-TR-002, 1999
15. Institute of Electrical and Electronics Engineers. Guide for Developing System Requirements Specification, IEEE Std.1233-1998
16. Institute of Electrical and Electronics Engineers. Guide for Information Technology System Definition – Concept of Operations (ConOps) Document, IEEE Std. 1362-1998
17. Thomas M, McGarry F. Top-Down vs. Bottom-Up Process Improvement. IEEE Software, July 1994: 12-13
18. Kamsties E, von Knethen A, Philipps J, Schätz B: Eine vergleichende Fallstudie mit CASE-Werkzeugen für formale und semi-formale Beschreibungstechniken. In: Proceedings of the 9. GI/ITG-Fachgespräch über Formale Beschreibungstechniken für verteilte Systeme, 1999, pp 103-112